Д. И. Соломон

ДРОБНОЕ ПРОГРАММИРОВАНИЕ И НЕДИФФЕРЕНЦИРУЕМАЯ ОПТИМИЗАЦИЯ

Кишинэу • Эврика • 2010

НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК УКРАИНЫ

ИНСТИТУТ КИБЕРНЕТИКИ ИМ. В.М. ГЛУШКОВА

МИНИСТЕРСТВО ПРОСВЯЩЕНИЯ
РЕСПУБЛИКИ МОЛДОВА

АКАДЕМИЯ ТРАНСПОРТА,
ИНФОРМАТИКИ И КОММУНИКАЦИЙ

Д. И. СОЛОМОН

ДРОБНОЕ ПРОГРАММИРОВАНИЕ И НЕДИФФЕРЕНЦИРУЕМАЯ ОПТИМИЗАЦИЯ

КИШИНЭУ • ЭВРИКА • 2010

Серия научных публикаций
«Недифференцируемая оптимизация и ее приложения»,
посвященная академику Н. З. ШОРУ

CZU 519.85
С 60

Descrierea CIP a Camerei Naţionale a Cărţii

Соломон, Д. И.

Дробное программирование и недифференцируемая оптимизация. / Д. И. Соломон; Нац. Акад. наук Украины, Акад. Транспорта, информатики и коммуникаций. – К.: Эврика, 2010 (Tipogr. AŞM). – 556 p.

150 ex.

ISBN 978-9975-941-53-2.

519.85
С 60

Монография посвящена задачам дробной оптимизации и методам их решения, основанным на алгоритмах минимизации недифференцируемых функций. Описываются различные субградиентные методы и основанные на них схемы декомпозиции, а также особенности их применения для решения задач дробной оптимизации.

Книга рассчитана на специалистов в области математического программирования, а также студентов и аспирантов соответствующих специальностей.

Рекомендовано к печати Ученым советом Института кибернетики им. В.М. Глушкова НАН Украины (протокол № 5 от 16.03.10) и Сенатом Академии транспорта, информатики и коммуникаций Республики Молдова (протокол № 6 от 18.02.10).

Рецензенты**: Н.Г. Журбенко**, канд. физ.-мат. наук, Институт кибернетики им. В.М. Глушкова, НАН Украины,
Д.К. Замбицкий, канд. физ.-мат. наук, Академия экономических знаний Молдовы

М–208–96
ISBN

НАУМ ЗУСЕЛЕВИЧ ШОР (1937 – 2006) – один из основоположников направления недифференцируемой оптимизации в математическом программировании. Автор 10 монографий и более 200 статей.

Родился 1 января 1937 г. в г. Киеве. После окончания в 1958 г. Киевского государственного университета им. Т. Г. Шевченко работал в Институте кибернетики им. В. М. Глушкова НАН Украины (с 1983 по 1990 г. – заведующим лабораторией, с 1990 по 2006 г. – заведующим отделом). Профессор (1989), член-корреспондент (1990), академик (1997) Национальной академии наук Украины. Лауреат Государственных премий УССР (1973), СССР (1981), Украины (1993, 1999), премий НАН Украины имени В. М. Глушкова (1987), имени В. С. Михалевича (1997).

Работы Н. З. Шора положили начало численным методам оптимизации недифференцируемых функций и существенно повлияли на развитие линейного, нелинейного, дискретного и стохастического программирования. Разработанные им методы получили мировую известность и способствовали становлению Института кибернетики им. В. М. Глушкова НАН Украины как одного из мировых лидеров в области теории и практики методов оптимизации.

Серия научных публикаций
«Недифференцируемая оптимизация и ее приложения», посвященная Науму Зуселевичу Шору

Лучшая память об ученом – это развитие его научных идей. В рамках научно-технического сотрудничества Института кибернетики им. В. М. Глушкова НАН Украины и Академии транспорта, информатики и коммуникаций Республики Молдова были изданы те результаты Н. З. Шора, которые рассредоточены по многочисленным статьям и стали ныне библиографической редкостью. Так появились сборники:

Шор Н. З. Методы недифференцируемой оптимизации и сложные экстремальные задачи: Сб. избр. тр. – Кишинэу: Эврика, 2008. – 270 с.

Шор Н. З. Методы минимизации негладких функций и матричные задачи оптимизации: Сб. избр. тр. – Кишинэу: Эврика, 2009. – 240 с.

В этих сборниках отражены основные результаты по численным методам минимизации недифференцируемых функций и их приложениям в сложных экстремальных задачах, которые получены в Институте кибернетики имени В. М. Глушкова НАН Украины в последние 50 лет.

Часть работ связана с методом обобщенного градиентного (субградиентного) спуска и разработанными на его основе декомпозиционными алгоритмами решения задач выпуклого программирования большой размерности. Значительное количество статей посвящено субградиентным методам с растяжением пространства, которые позволили эффективно решать задачи минимизации негладких функций с овражными особенностями. Мировое признание получили частные случаи этих методов – метод эллипсоидов и r-алгоритмы. Они значительно расширили область приложения методов недифференцируемой оптимизации.

В сборники включено также много работ по разработке алгоритмов оптимизации негладких матричных функций (с приложениями в теории динамических систем и теории устойчивости, для построения описанных и вписанных экстремальных эллипсоидов) и новых подходов к решению многоэкстремальных квадратичных задач (с приложениями в задачах полиномиальной оптимизации и комбинаторных булевых задачах). Последние позволили установить связь между задачами полиномиальной оптимизации и 17-й проблемой Гильберта, между двойственными оценками Шора и известными числами Ловаса $\theta(G, w)$ и $\theta'(G, w)$ для NP-трудной комбинаторной задачи о максимальном взвешенном независимом множестве вершин графа.

Чтобы более полно представить результаты, полученные под руководством Н. З. Шора, планируется продолжить издание сборников его избранных трудов, а также издать серию книг под общим названием «Недифференцируемая оптимизация и ее приложения». Каждая отдельная книга этой серии будет отражать современные теоретические и практические достижения тех или иных направлений, связанных с научным наследием академика Н. З. Шора.

Монография Д. И. Соломона открывает эту серию научных публикаций. Она является результатом многолетних исследований автора, в которых Институт кибернетики им. В. М. Глушкова НАН Украины и лично Н. З. Шор сыграли очень важную роль. Под его руководством Д. И. Соломон успешно защитил кандидатскую диссертацию (1985), результаты которой затем вошли в монографию: Шор Н. З., Соломон Д. И. Декомпозиционные методы в дробно-линейном программировании. – Кишинев: Штиинца, 1989. – 204 с. Эти исследования были продолжены в докторской диссертации Д. И. Соломона (1992).

Данная монография посвящена методам решения задач дробного программирования и тесно связана с главными направлениями научных исследований Н. З. Шора. Одним из центральных моментов в ней является использование субградиентных методов и основанных на них схем декомпозиции. Значительное внимание в монографии уделено задачам большой размерности и задачам транспортного типа. Именно для транспортной задачи в сетевой форме Н. З. Шор разработал первый метод негладкой оптимизации – метод обобщенного градиентного спуска (1962).

Следующими в серии «Недифференцируемая оптимизация и ее приложения» планируются такие публикации: Журбенко Н. Г. Субградиентные методы минимизации негладких функций; Стецюк П. И. Методы эллипсоидов и их обобщения.

Надеемся, что данная серия научных публикаций окажется полезной для специалистов в области математического программирования, которые найдут в ней ряд оригинальных идей как для развития численных методов негладкой оптимизации, так и для построения эффективных алгоритмов решения экстремальных задач (выпуклых, многоэкстремальных, комбинаторных и др.).

Редакционная коллегия

Введение

Оптимизационные задачи давно и прочно заняли свое место в процессах принятия решений, которые являются одним из важнейших аспектов деятельности людей. Управление сложными техническими, технологическими и экономическими процессами, задачи в различных областях науки, приводят к необходимости выбора и принятия оптимальных решений. Одной из областей математической науки, изучающей методы описания, анализа и обоснования оптимальных решений различных задач по принятию решений является «Исследование операций». Это направление, начиная с 50-х годах прошлого столетия, сформировалось как самостоятельная область математических исследований, связанная с разработкой моделей и методов, анализа сложных оптимизационных задач, получения оптимальных решений конкретных задач и практических применений.

Первые работы в данной области связаны с именами таких известных ученых, как Л.В. Канторович, Дж.Б. Данциг, Т. Купманс, Дж. фон Нейман, А. Гофман, А.У. Таккер, Г. Кун, Р. Дорфман, Л. Форд, Д. Фалкерсон, С. Гасс, Л. С. Понтрягин, Р. Беллман и многих других известных математиков, которые создали теоретическую основу исследования операций и предложили первые методы решения многих конкретных оптимизационных задач. Именно с именами этих ученых связано появление таких направлений как линейное и нелинейное программирование, транспортные задачи и оптимизация на сетях, теория игр, оптимальное управление и динамическое программирование, стохастическая оптимизация и статистический анализ.

В дальнейшем получили развитие конкретные методы и алгоритмы, предназначенные для решения отдельных классов оптимизационных задач, такие как симплексные методы в линейном программировании, методы возможных направлений и градиентные методы в выпуклом программировании, методы штрафных и барьерных функций для общих задач математического программирования, методы условной и безусловной оптимизации, методы и алгоритмы решения задач глобальной, дискретной, целочисленной, комбинаторной и недифференцируемой оптимизации.

К настоящему времени разработаны целые классы методов и алгоритмов, связанные с решением оптимизационных задач определенных типов, образовавшие отдельные направления оптимизации. К ним относится методы одномерной и многомерной оптимизации, методы условной и безусловной оптимизации, методы выпуклой и вогнутой оптимизации гладких и недифференцируемых функций, методы однокритериального, многокритериального и целевого программирования, методы стохастической, квадратичной и глобальной оптимизации, и многие другие методы, учитывающие специфику задач оптимизации. Особую роль в развитии численных методов оптимизации сыграли такие общие методы, как субградиентные, секущих плоскостей, центров, внутренних точек, ветвей и границ, которые приложили путь к новым методам локальной и глобальной оптимизации. На базе этих исследований опубликованы ряд монографических работ и книг по исследования операций, среди которых можно отметить работы [247, 544, 586, 587, 588, 592, 593, 650, 653, 699, 709, 739] и различным направлениям оптимизации по вопросам:

- линейного программирования [43, 93, 146, 162, 166, 167, 173, 217, 241, 349, 388, 397, 485, 625, 626, 641, 601, 602, 605, 740, 741];
- выпуклого программирования [18, 92, 141, 143, 393, 394, 396, 409, 417, 418, 521, 652, 660];
- математического программирования [19, 20, 21, 31, 76, 77, 88, 160, 175, 190, 193, 194, 196, 197, 199, 221, 230, 242], [255]–[257], [269, 311, 319, 326, 335, 352, 359, 365, 366, 389, 402, 408, 425, 459, 502, 504, 524, 532, 548, 571, 578, 581, 584, 598, 609, 612, 613, 618, 627, 632, 635, 644, 647, 655], [656]–[659], [661, 696, 697, 703, 707];
- дробно-выпуклого и дробно-линейного программирования [32, 36, 148, 359, 438, 493, 494, 633, 733];
- многокритериальной оптимизации [119, 180, 347, 364, 430, 503, 634, 650, 653, 654];
- стохастической оптимизации [81, 183, 184, 201, 246, 281, 282, 413, 615, 651, 743];
- недифференцируемой оптимизации [33, 302, 469, 470, 606, 607, 715].

В данной монографии аппарат недифференцируемой оптимизации используется для решения задач дробного программирования. Она написана на основе проведенных автором исследований в области решения задач дробно-линейного программирования большой размерности, обобщенных задач дробно-выпуклого программирования, сепарабельных блочных задач, специальных задач квадратичной, стохастической и полиномиальной дробной оптимизации, а также дробно-линейных транспортных задач. В монографии подытожены результаты, полученные автором и отраженные в его диссертационных работах [664, 667], научных статьях [482]–[484], [487, 496, 663, 665, 666], [668]–[694], монографиях [633, 733] и книгах [592, 593].

Изложенный материал в данной монографии можно разбить условно на четыре части, которые имеют свои особенности и области исследования, однако они преследуют единую цель, а именно – показать эффективность применения методов недифференцируемой оптимизации для решения различных задач дробного программирования.

К первой части относятся первые три главы данной монографии, которые носят общий характер. Здесь приведены теоретические и практические описания основных задач оптимизации, необходимые и достаточные условия оптимальности, функции Лагранжа, теория двойственности в математическом программировании. Описываются субградиентные методы недифференцируемой оптимизации и схемы декомпозиции по ограничениям, переменным и ресурсам, которые могут быть использованы для решения различных задач условной и безусловной оптимизации с общей или специальной структурой системы ограничений, с линейными, дробно-линейными, выпуклыми, дробно-выпуклыми или нелинейными функционалами.

Вторая часть данной монографии охватывает три последующие главы, которые посвящены вопросам решения основных трех классов задач дробного программирования, а именно:

1) задачи дробно-линейного программирования;

2) задачи дробно-выпуклого программирования;

3) обобщенные задачи дробного программирования.

Применяемые в данных главах методы и алгоритмы решения указанных дробных задач можно разбить на следующие три группы:

- прямые методы, в которых для решения дробных задач применяются общие методы математического программирования;

- параметрические методы, в которых дробная задача сводится к решению задач оптимизации параметрической функции, составленной из функций числителя и знаменателя;

- декомпозиционные методы, в которых учитывается специфика ограничений задачи, а также оптимизируемый функционал. Данные методы, как правило, основаны на функции Лагранжа и схемах декомпозиции по ограничениям, переменным и ресурсам, с применением субградиентных методов недифференцируемой оптимизации.

К третьей части относится седьмая глава данной монографии, в которой рассматриваются специальные задачи дробного программирования, а именно, задачи квадратичной, стохастической и полиномиальной дробной оптимизации. Также рассматриваются задачи дробной оптимизации с неизвестными коэффициентами.

К четвертой части данной монографии относится последняя, восьмая глава, в которой приводятся алгоритмы и методы решения дробных задач транспортного типа.

В каждой части данной книги рассматриваются особенности и эффективность применения методов недифференцируемой оптимизации для решения задач дробного программирования. Субградиентные методы применяются одновременно с другими методами и алгоритмами решения задач дробной оптимизации и показывается их эффективность в случае специальной структуры ограничений и дробных функционалов.

Остановимся кратко на содержание каждой главы.

Первая глава данной книги посвящена теоретическим основам дробной оптимизации. В ней приводятся постановки основных задач дробного программирования, рассматриваемых в данной работе, и проводится анализ разработанных методов и алгоритмов их решения. Приводятся основные свойства выпуклых и квазивыпуклых функций, определения и свойства субградиентов недифференцируемых функций, описываются общие задачи условной оптимизации, функции Лагранжа и задачи нахождения седловых точек. Приводятся основные теоретические результаты в нелинейном, выпуклом и дробном программировании. Рассматривается связь между методами недифференцируемой оптимизации и методами решения задач квадратичного программирования, методами штрафных и барьерных функций, а также приводятся негладкие штрафные функции для задач выпуклого и дробно-выпуклого программирования.

Во второй главе описываются основные три группы методов недифференцируемой оптимизации, а именно: методы обобщенного градиентного спуска, субградиентные методы с растяжением пространства и метод эллипсоидов. Данные методы в дальнейшем применяются для решения различных задач линейного и дробно-линейного, выпуклого и дробно-выпуклого программирования, обобщенных задач дробного программирования, а также транспортных задач с линейными и дробно-линейными функционалами.

В третьей главе приводятся основные направления применения методов недифференцируемой оптимизации, связанные с решением различных негладких задач математического программирования, построением точных негладких штрафных функций и полиномиальных алгоритмов решения задач оптимизации, разработке схем декомпозиции по ограничениям, переменным и ресурсам для решения задач блочного и сепарабельного типа. Применение методов релаксации по Лагранжу и рассмотрении двойственных задач, позволили свести задачи условной оптимизации к решению негладких задач безусловной оптимизации.

В четвертой главе рассматриваются задачи дробно-линейного программирования, для которых приводятся различные алгоритмы их решения: прямой и модифицированный симплекс-метод, метод сведения к задачам линейного программирования, параметрический метод, полиномиальные алгоритмы и декомпозиционные методы. Для блочных дробно-линейных задач предлагаются четыре группы декомпозиционных методов, а именно принцип декомпозиции Данцига-Вулфа и три схемы декомпозиции: первая - по переменным, вторая - по ограничениям и третья – по ресурсам. При их применении решаются подзадачи линейного программирования для каждого блока в отдельности.

В пятой главе данной книги рассматриваются задачи дробно-выпуклого программирования, для решения которых приводятся метод преобразования переменных, параметрический метод и методы нелинейной оптимизации квазивыпуклых функций. Одновременно с ними предлагается применить методы недифференцируемой оптимизации для решения общей задачи дробно-выпуклого программирования или для решения задач с блочно-диагональной структурой ограничений и сепарабельных дробно-выпуклых функционалов.

Субградиентные методы используются для нахождения седловой точки дробно-выпуклой функции, а метод эллипсоидов - для решения задач дробно-выпуклого программирования. Применение негладких штрафных функций позволяет свести дробные задачи к задачам безусловной оптимизации, к которым можно применить субградиентные методы. Для решения задач дробно-выпуклого программирования приводятся рассмотренные в предыдущей главе три группы схем декомпозиции, а именно: по переменным, ограничениям и ресурсам.

В шестой главе, которая является центральной для данной монографии, рассматриваются обобщенные задачи дробно-выпуклого программирования, которые получаются в результате применения операций максимума, суммирования или произведения дробных функций. Для их решения предлагаются три группы методов дробной оптимизации: параметрические методы, прямые методы и схемы декомпозиции с применением субградиентных методов. Данные методы также приводятся и для решения задач оптимизации обобщенных дробно-линейных функций на линейных многогранниках. Эти методы предлагается использовать при решении дробных задач многокритериальной оптимизации, когда в результате применения метода сверток приходим к обобщенным задачам дробного программирования.

В этой же главе рассматриваются и сепарабельные задачи обобщенного дробного программирования блочной структуры, для решения которых предлагаются схемы декомпозиции по ограничениям, переменным и ресурсам. В результате применения схем декомпозиции приходим к одноуровневым или двухуровневым алгоритмам недифференцируемой оптимизации.

Седьмая глава посвящена специальным задачам дробного программирования, для решения которых применяются методы недифференцируемой оптимизации. Первая рассматриваемая задача относится к задаче дробно-выпуклого квадратичного программирования, для решения которой кроме параметрического метода, предлагается применить метод преобразования переменных, схемы декомпозиции по переменным и субградиентные методы, в результате которых приходим к решению последовательности выпуклых задач квадратичной оптимизации или к задачам линейного программирования.

Вторая задача, рассматриваемая в данной главе - это задача дробного-полиномиального программирования, для которой путем замены переменных и применением схем декомпозиции по ограничениям, приходим к решению задач недифференцируемой оптимизации субградиентным методом, на каждом шаге которого решаем задачи дробно-линейного программирования

Для решения задачи двухэтапного стохастического дробного программирования предлагается применять схемы декомпозиции по переменным, которые позволяет свести их к решению задач безусловной оптимизации субградиентным методом, на каждом шаге которого решаются задачи линейного или дробно-линейного программирования.

Последняя задача из данной главы относится к задачам дробно-линейного программирования с неизвестными коэффициентами. Для решения задач с линейными коэффициентами, которые определяются для каждой переменной из отдельного многогранного множества, предлагается применить декомпозиционный метод Данцига-Вулфа и схемы декомпозиции по переменным и ограничениям. Для случая, когда неизвестные коэффициенты нелинейны, применяется схема декомпозиции по переменным и субградиентные методы.

В последней восьмой главе данной книги рассмотрены дробно-линейные транспортные задачи, обобщенные транспортные задачи с дробно-линейными функционалами и задачи дробного программирования производственно-транспортного типа. Для решения дробных задач транспортного типа предлагаются три группы прямых методов, а именно методы потенциалов, параметрические методы и принцип декомпозиции Данцига-Вулфа. Одновременно с ними, для решения транспортных задач большой размерности предлагается применить схемы декомпозиции по ограничениям и переменным, основанные на методах недифференцируемой оптимизации.

Именно для решения линейных транспортных задач получены первые эффективные субградиентные методы, которые определили в дальнейшем новые направления недифференцируемой оптимизации – методы обобщенного градиентного спуска, субградиентные методы с растяжением пространства, метод эллипсоидов, а также схемы декомпозиции по ограничениям и переменным.

Глава 1

Основы дробного программирования

В данной главе приводятся постановки основных задач дробного программирования, которые в дальнейшем будут исследованы и для решения которых предлагаются эффективные методы. Среди них можно выделить:

1. Задачи дробно-линейного программирования.
2. Задачи дробно-выпуклого программирования.
3. Задачи обобщенного дробно-линейного и дробно-выпуклого программирования.
4. Задачи сепарабельного дробно-выпуклого программирования.
5. Задачи дробной оптимизации специальной и блочной структуры.
6. Задачи дробной оптимизации транспортного типа.
7. Задачи дробно-квадратического, дробно-полиномиального и дробно-стохастического программирования.

Приводятся теоретические основы дробной оптимизации, необходимые для разработки методов и алгоритмов решения рассматриваемых задач дробного программирования. Даются понятия выпуклых и квазивыпуклых функций, свойства градиентов и субградиентов. Приводятся постановки общих задач оптимизации, функции Лагранжа, теория двойственности в нелинейном, выпуклом и дробном программировании, а также соответствующие им основные условия оптимальности. Рассматривается задача квадратичного программирования и ее связь с методами недифференцируемой оптимизации. Описываются методы штрафных и барьерных функций и рассматриваются негладкие штрафные функции для задач выпуклого и дробно-выпуклого программирования.

1.1. Задачи дробного программирования

В общем виде задача дробного программирования формулируется следующим образом: найти вектор переменных $x = (x_1, x_2, \ldots, x_n) \in R^n$, минимизирующий функцию

$$F_1(x) = \sum_{k=1}^{p} \frac{\varphi_k(x)}{\psi_k(x)} \tag{1.1}$$

при ограничениях

$$f_i(x) \le 0 \quad (i = \overline{1, m}), \tag{1.2}$$

$$x \in X, \tag{1.3}$$

где $X \in R^n$ – заданный компакт, $f_i(x)$ $(i = \overline{1, m})$ – выпуклые функции, а $\varphi_k(x)$, $(k = \overline{1, p})$ и $\psi_k(x)$, $(k = \overline{1, p})$ – непрерывные функции, определенные на компакте X.

Задача (1.1)–(1.3) может быть обобщена на случай, если в функции (1.1) вместо суммы использовать операцию взятия минимума или максимума. Тогда получим задачу минимизации функции

$$F_2(x) = \max_{1 \le k \le p} \frac{\varphi_k(x)}{\psi_k(x)} \tag{1.4}$$

или

$$F_3(x) = \frac{\max\limits_{1 \le k \le p} \varphi_k(x)}{\min\limits_{1 \le k \le p} \psi_k(x)} \tag{1.5}$$

при ограничениях (1.2)–(1.3).

Задачи дробного программирования можно расширять и на случаях, когда необходимо минимизировать мультипликативную функцию

$$F_4(x) = \prod_{k=1}^{p} \frac{\varphi_k(x)}{\psi_k(x)} \tag{1.6}$$

или векторную (многокритериальную) функцию

$$F_5(x) = \left(\frac{\varphi_1(x)}{\psi_1(x)}, \frac{\varphi_2(x)}{\psi_2(x)}, \ldots, \frac{\varphi_p(x)}{\psi_p(x)} \right) \tag{1.7}$$

при ограничениях (1.2)–(1.3).

Задачи дробного программирования широко изучены, и в данной области опубликовано достаточно много работ. В монографических

[32, 36, 148, 359, 438, 493, 494, 633, 733] и библиографических [433], [488]–[493] работах приведены и анализированы различные, общие и частные постановки задач дробного программирования, для решения которых приводятся многочисленные методы и алгоритмы, большинство из которых основаны на общей теории линейного и выпуклого программирования, нелинейного и динамического программирования, локальной, глобальной, многокритериальной или дискретной оптимизации.

В данной книге исследуются вопросы применения для решения задач дробного программирования методов недифференцируемой оптимизации, основная часть которых основана на функциях Лагранжа, двойственных задачах, штрафных функциях, седловой точке и субградиентных методах. В ней рассматриваются следующие классы задач дробного программирования, для решения которых, наряду с другими методами автором разработаны эффективные методы и алгоритмы, основанные на схемах декомпозиции по ограничениям, переменным и ресурсам с применением методов недифференцируемой оптимизации. Для начала приведем постановку основных задач дробного программирования.

1. Задача дробно-выпуклого программирования

$$\frac{\varphi(x)}{\psi(x)} \to \min,$$

$$\begin{aligned} f_i(x) &\le 0 \quad (i = \overline{1, m}), \\ x &\in X, \end{aligned}$$

в которой функции $\varphi(x)$, $-\psi(x)$, $f_i(x)$ $(i = \overline{1, m})$ являются выпуклыми.

2. Задача дробно-линейного программирования

$$\frac{\sum\limits_{j=1}^{n} c_j x_j + c_0}{\sum\limits_{j=1}^{n} d_j x_j + d_0} \to \min,$$

$$\begin{aligned} \sum_{j=1}^{n} a_{ij} x_j &\le b_i \qquad (i = \overline{1, m}), \\ x_j &\ge 0 \qquad (j = \overline{1, n}), \end{aligned}$$

в которой все функции являются линейными.

3. Дробно-линейная транспортная задача

$$\frac{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + c_0}{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}x_{ij} + d_0} \to \min,$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}),$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,n}),$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}).$$

4. Дробно-линейная задача назначения

$$\frac{\sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}x_{ij} + c_0}{\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}x_{ij} + d_0} \to \min,$$

$$\sum_{j=1}^{m} x_{ij} = 1 \quad (i = \overline{1,m}),$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad (j = \overline{1,m}),$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,m}).$$

5. Дробно-линейная дискретная задача размещения

$$\frac{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{j=1}^{n} \alpha_j y_j}{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}x_{ij} + \sum_{j=1}^{n} \beta_j y_j} \to \min,$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}),$$

$$\sum_{i=1}^{m} x_{ij} = y_j \quad (j = \overline{1,n}),$$

$$p_j \le y_j \le q_j \quad (j = \overline{1,n}),$$

$$x_{ij} = \begin{cases} a_i, \\ 0 \end{cases} \quad (i = \overline{1,m};\ j = \overline{1,n}).$$

6. Нелинейная производственно-транспортная задача дробного программирования

$$\frac{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} c_{ij}x_{ij} + \sum\limits_{j=1}^{n}\varphi_j(y_j) + \sum\limits_{j=1}^{n}\sum\limits_{l=1}^{k} p_{jl}z_{jl}}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} d_{ij}x_{ij} + \sum\limits_{j=1}^{n}\psi_j(y_j) + \sum\limits_{j=1}^{n}\sum\limits_{l=1}^{k} q_{jl}z_{jl}} \to \min,$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}),$$

$$\sum_{i=1}^{m} f_{ij}x_{ij} = y_j \quad (j = \overline{1,n}),$$

$$y_j = \sum_{l=1}^{k} z_{jl} \quad (j = \overline{1,n}),$$

$$\sum_{j=1}^{n} z_{jl} = b_l \quad (l = \overline{1,k}),$$

$$x_{ij} \ge 0 \quad (i = \overline{1,m};\ j = \overline{1,n}),$$

$$z_{jl} \ge 0 \quad (j = \overline{1,n};\ l = \overline{1,k}).$$

7. Задачи обобщенного дробно-выпуклого программирования, которые относятся к минимизации пяти специализированных функций при выпуклых функциях, а именно рассматриваются задачи:

$$\min F_r(x) \quad (r = \overline{1,5}),$$

$$f_i(x) \le 0 \qquad (i = \overline{1,m}),$$
$$x \in X,$$

в которых

$$F_1(x) = \sum_{k=1}^{p} \frac{\varphi_k(x)}{\psi_k(x)}, \quad F_2(x) = \max_{1\le k\le p} \frac{\varphi_k(x)}{\psi_k(x)}, \quad F_3(x) = \frac{\max\limits_{1\le i\le p} \varphi_k(x)}{\min\limits_{1\le i\le p} \psi_k(x)},$$

$$F_4(x) = \prod_{k=1}^{p} \frac{\varphi_k(x)}{\psi_k(x)}, \quad F_5(x) = \left(\frac{\varphi_1(x)}{\psi_1(x)}, \frac{\varphi_2(x)}{\psi_2(x)}, \ldots, \frac{\varphi_p(x)}{\psi_p(x)}\right),$$

а функции

$$\varphi_k(x) \ \ (k=\overline{1,p}), \quad -\psi_k(x) \ (k=\overline{1,p}), \quad f_i(x) \ \ (i=\overline{1,m})$$

являются выпуклыми или линейными.

8. Задача сепарабельного, блочного и дробно-выпуклого программирования в следующей постановке.

Пусть имеем p групп переменных $x_j = (x_j^1, x_j^2, \ldots, x_j^{n_j}) \in R^{n_j}$, образующих вектор $x = (x_1, x_2, \ldots, x_p) \in R^n$, где $n = \sum\limits_{j=1}^{p} n_j$. На множестве каждой группы переменных x_j определим следующие непрерывные функции $\varphi_j, \psi_j, g_j^i \ \ (i=\overline{1,m}),\ h_j^l \ \ (l=\overline{1,m_j}):\ R^{n_j} \to R$.

Тогда задача сепарабельного, блочного и дробно-выпуклого программирования имеет вид:

$$\frac{\sum\limits_{j=1}^{p} \varphi_j(x_j)}{\sum\limits_{j=1}^{p} \psi_j(x_j)} \to \min,$$

$$\sum_{j=1}^{p} g_j^i(x_j) \le a_i \quad (i=\overline{1,m}),$$

$$h_j^l(x_j) \le b_j^l \qquad (l=\overline{1,m_j};\ j=\overline{1,p}).$$

Функции $\varphi_j(x_j),\ -\psi_j(x_j),\ g_j^i(x_j) \ \ (i=\overline{1,m}),\ h_j^l(x_j) \ \ (l=\overline{1,m_j})$ выпуклы или линейны на множестве переменных $x_j \in R^{n_j}, \ \ j=\overline{1,p}$.

9. Задача дробно-линейного программирования блочно-диагональной структуры

$$\frac{\sum\limits_{j=1}^{p} c_j x_j}{\sum\limits_{j=1}^{p} d_j x_j} \to \min,$$

$$\sum_{j=1}^{p} A_j x_j \le b,$$

$$B_j x_j \le b_j \quad (j = \overline{1,p}),$$

$$x_j \ge 0 \quad (j = \overline{1,p}),$$

$$y \ge 0,$$

в которой c_j, d_j, x_j — n_j-мерные векторы; b — m-мерный вектор; b_j — m_j-мерные векторы; A_j – матрицы размерности $m \times n_j$; B_j – матрицы размерности $m_j \times n_j$.

10. Блочная задача дробно-выпуклого и сепарабельного программирования со связующими переменными и ограничениями в следующей постановке.

Пусть имеем p групп переменных $x_j = (x_j^1, x_j^2, \ldots, x_j^{n_j}) \in R^{n_j}$, образующих вектор $x = (x_1, x_2, \ldots, x_p) \in R^n$, где $n = \sum\limits_{j=1}^{p} n_j$. На множестве каждой группы переменных x_j определены следующие непрерывные функции $\varphi_j, \psi_j, g_j^i \ \ (i = \overline{1,m}),\ h_j^l \ \ (l = \overline{1,m_j}) : \ R^{n_j} \to R$.

Пусть также имеются переменные $y = (y_1, y_2, \ldots, y_k) \in R^k$, на которых определены функции $\alpha, \beta, F_0^i \ \ (i = \overline{1,m})$, $F_j^l \ \ (l = \overline{1,m_j}; j = \overline{1,p}) : \ R^k \to R$. Предположим, что функции $\varphi_j, -\psi_j, g_j^i \ \ (i = \overline{1,m})$, $h_j^l \ \ (l = \overline{1,m_j})$ выпуклы на множестве значений переменных x_j, а функции $\alpha, -\beta, F_0^i \ \ (i = \overline{1,m})$, $F_j^l \ \ (l = \overline{1,m_j}; j = \overline{1,p})$ выпуклы на множестве значений переменных y.

Тогда сепарабельная задача дробно-выпуклого программирования на множестве переменных (x, y), в которой система ограничений имеет блочно-диагональную структуру со связующими переменными и ограничениями имеет вид:

$$F(x,y) = \frac{C(x,y)}{D(x,y)} = \frac{\sum\limits_{j=1}^{p} \varphi_j(x_j) + \alpha(y)}{\sum\limits_{j=1}^{p} \psi_j(x_j) + \beta(y)} \to \min,$$

$$\sum_{j=1}^{p} g_j^i(x_j) + F_0^i(y) \le a_i \quad (i = \overline{1,m}),$$

$$h_j^l(x_j) + F_j^i(y) \le b_j^l \quad (l = \overline{1,m_j};\ j = \overline{1,p}),$$

в которой $a_i \quad (i = \overline{1,m})$ и $b_j^l \quad (l = \overline{1,m_j};\ j = \overline{1,p})$ – заданные числа.

11. Задача дробно-линейного программирования блочно-диагональной структуры со связующими переменными и ограничениями

$$\frac{\sum\limits_{j=1}^{p} c_j x_j + py}{\sum\limits_{j=1}^{p} d_j x_j + qy} \to \min,$$

$$\sum_{j=1}^{p} A_j x_j + F_0 y \le b,$$

$$B_j x_j + F_j y \le b_j \quad (j = \overline{1,p}),$$

$$x_j \ge 0 \quad (j = \overline{1,p}),$$

$$y \ge 0,$$

в которой c_j, d_j, x_j – n_j-мерные векторы, p, q, y – k-мерные векторы; b — m-мерный вектор; b_j — m_j-мерные векторы; A_j – матрицы размерности $m \times n_j$; B_j – матрицы размерности $m_j \times n_j$; F_0 – матрица размерности $m \times k$; F_j – матрицы размерности $m_j \times k$.

12. Обобщенные задачи сепарабельного дробно-линейного программирования блочно-диагональной структуры со связующими переменными и ограничениями в следующих постановках.

Пусть имеем p групп переменных $x_j = (x_j^1, x_j^2, \ldots, x_j^{n_j}) \in R^{n_j}$, образующих вектор $x = (x_1, x_2, \ldots, x_p) \in R^n$, где $n = \sum\limits_{j=1}^{p} n_j$. На множестве каждой группы переменных x_j определены следующие непрерывные функции: $\varphi_j,\ \psi_j,\ g_j^i \quad (i = \overline{1,m}),\ h_j^l \quad (l = \overline{1,m_j}) :\ R^{n_j} \to R$.

Пусть также имеются переменные $y = (y_1, y_2, \ldots, y_k) \in R^k$, на которых определены функции $\alpha, \beta, F_0^i \quad (i = \overline{1,m}), F_j^l \quad (l = \overline{1,m_j};\ j = \overline{1,p}) : \ R^k \to R$. Предположим, что функции $\varphi_j,\ -\psi_j,\quad g_j^i \quad (i = \overline{1,m_j}),\ h_j^l \quad (l = \overline{1,m_j})$ выпуклы на множестве значений переменных x_j, а функции $\alpha,\ -\beta, F_0^i \quad (i = \overline{1,m}), \quad F_j^l \quad (l = \overline{1,m_j}; \quad j = \overline{1,p})$ выпуклы на множестве значений переменных y.

Рассмотрим следующие два множества допустимых значений переменных x и y:

$$Q(x,y) = \{x, y\colon h_j^l(x_j) + F_j^l(y) \le b_j^l \quad (l = \overline{1,m_j};\ j = \overline{1,p})\},$$

$$S(x,y) = \left\{x, y\colon \sum_{j=1}^{p} g_j^i(x_j) + F_0^i(y) \le a_i \quad (i = \overline{1,m});\right.$$

$$\left. h_j^i(x_j) + F_j^l(y) \le b_j^l \quad (l = \overline{1,m_j};\ j = \overline{1,p})\right\}.$$

Для функций

$$F_1(x,y) = \frac{\sum\limits_{j=1}^{p} \varphi_j(x_j)}{\sum\limits_{j=1}^{p} \psi_j(x_j)} + \frac{\alpha(y)}{\beta(y)},$$

$$F_2(x,y) = \frac{\sum\limits_{j=1}^{p} \varphi_j(x_j)}{\sum\limits_{j=1}^{p} \psi_j(x_j)} + \alpha(y),$$

$$F_3(x,y) = \sum_{j=1}^{p} \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \frac{\alpha(y)}{\beta(y)},$$

$$F_4(x,y) = \sum_{j=1}^{p} \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \alpha(y),$$

$$F_5(x,y) = \max_j \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \frac{\alpha(y)}{\beta(y)},$$

$$F_6(x,y) = \max_j \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \alpha(y),$$

$$F_7(x,y) = \frac{\max\limits_j \varphi_j(x_j)}{\min\limits_j \psi_j(x_j)} + \frac{\alpha(y)}{\beta(y)},$$

$$F_8(x,y) = \frac{\max\limits_j \varphi_j(x_j)}{\min\limits_j \psi_j(x_j)} + \alpha(y),$$

$$F_9(x,y) = \sum_{j=1}^{p} \varphi_j(x_j) + \frac{\alpha(y)}{\beta(y)},$$

$$F_{10}(x,y) = \max_j \varphi_j(x_j) + \frac{\alpha(y)}{\beta(y)}$$

рассмотрим следующие задачи:

$$\min_{(x,y)\in Q(x,y)} F_r(x,y) \quad (r = \overline{1,10})$$

или

$$\min_{(x,y)\in S(x,y)} F_r(x,y) \quad (r = \overline{1,10}).$$

13. Двухэтапная дробно-линейная задача стохастического программирования.

Рассмотрим задачу минимизации одной из функций (первый этап):

$$F_1(x) = (c,x) + E_1(x),$$

$$F_2(x) = (c,x) + E_2(x),$$

$$F_3(x) = \frac{(c,x) + E_1(x)}{(d,x)},$$

$$F_4(x) = \frac{(c,x)}{(d,x)} + E_1(x),$$

$$F_5(x) = \frac{(c,x)}{(d,x)} + E_2(x)$$

при ограничениях

$$x \geq 0,$$

где c, d, x – n_1-мерные вектора, $E_1(x)$ и $E_2(x)$ – математические ожидания случайных функций $\varphi_1(x,\omega)$ и $\varphi_2(x,\omega)$, определенные одним из следующих способов (второй этап):

$$\varphi_1(x,\omega) = \min_z (h,z)$$

или
$$\varphi_2(x,\omega)=\min_z \frac{(h,z)}{(g,z)}$$
при ограничениях
$$z \geq 0,$$
$$Dz \geq b_\omega - Ax,$$
где h, g, z – n_2-мерные вектора, A, D – матрицы $(m \times n_1)$ и $(m \times n_2)$ соответственно, b_ω – случайный m-мерный вектор.

14. Общие и частные задачи дробно-квадратичного программирования:
$$\frac{C(x)}{D(x)} \to \min$$
при ограничениях
$$K_i(x) \leq 0 \quad (i=\overline{1,m}),$$
где $C(x), D(x), K_i(x)$ – квадратичные или линейные функции, $i=\overline{1,m}$.

15. Задача дробно-полиномиального программирования:
$$\frac{C(x)}{D(x)} \to \min,$$
$$P_i(x) = 0 \quad (i=\overline{1,m}),$$
где $x=(x_1,x_2,\ldots,x_n) \in E^n$, а $C(x), D(x)$ и $P(x)$ – полиномиальные функции от x, т.е. функции вида
$$C(x)=\sum_{r\in R_C} c_r \prod_{j=1}^n x_j^{\alpha_{rj}};$$
$$D(x)=\sum_{r\in R_D} d_r \prod_{j=1}^n x_j^{\beta_{rj}};$$
$$P_i(x)=\sum_{r\in R_{P_i}} a_{ir} \prod_{j=1}^n x_j^{\gamma_{rj}^i} \quad (i=\overline{1,m}),$$
в которых: R_C – множество мономов с ненулевыми коэффициентами c_r, которые формируют полином $C(x)$; R_D – множество мономов с ненулевыми коэффициентами d_r, которые формируют полином $D(x)$; R_{P_i} $(i=\overline{1,m})$ – множество мономов с ненулевыми коэффициентами a_{ir}, которые формируют полиномы $P_i(x)$; c_r $(r \in R_C), d_r$ $(r \in$

$\in R_D), a_{ir}$ $(r \in R_{P_i},\ i = \overline{1, m})$ – ненулевые коэффициенты мономов, которые соответственно входят в полиномы $C(x), D(x)$ и $P_i(x)$; α_{rj} $(r \in \in R_C,\ j = \overline{1, n})$; β_{rj} $(r \in R_D,\ j = \overline{1, n})$; γ^i_{rj} $(r \in R_{P_i},\ i = \overline{1, m},\ j = \overline{1, n})$ – положительные целые числа.

Как видно из приведенных выше постановок задач оптимизации с дробными функционалами, в задачах дробного программирования требуется минимизировать или максимизировать функционал, представляющий собой отношение двух функций. В зависимости от свойств функций, входящие в дробный функционал и тех, которые описывают допустимое множество оптимизации, задачи дробного программирования можно разделить на следующие классы:

- задачи дробно-линейного, дробно-выпуклого, дробно-квадратичного и дробно-стохастического программирования;
- задачи обобщенного дробного программирования;
- задачи целочисленной, дискретной, полиномиальной и многокритериальной дробной оптимизации;
- дробные задачи транспортного или распределительного типа;
- дробные задачи специальной структуры.

Как видно из перечисленных классов задач дробной оптимизации они покрывают почти весь спектр задач оптимизации с критериями, определенные для одной функции. Поэтому для решения задач дробной оптимизации используются методы и алгоритмы полученные для задач с одним функционалом. Ниже приведем некоторый анализ возможного применения таких методов для решения задач дробного программирования.

Рассмотрим общую задачу оптимизации в виде $\min\limits_{x \in S} \varphi(x)$, где $\varphi(x)$ – оптимизирующая функция, x – переменные, на которых проводится оптимизация, а S – множество допустимой области оптимизации. Для задач дробной оптимизации разработанные методы решения задачи $\min\limits_{x \in S} \varphi(x)$, можно разделить на два типа:

1) методы, которые непосредственно применены для дробной оптимизации (множество функций $\varphi(x)$ включают в себе дробные функции);

2) методы, разработанные для указанных задач нельзя прямо применить к задачам дробной оптимизации. Они требуют специальной

доработки (адаптации). Это означает, что функции $\varphi(x)$ не включают в себе дробные функции.

Если имеющиеся методы решения задачи $\min\limits_{x\in S}\varphi(x)$ позволяют решать и задачи оптимизации с дробным функционалом, то такие методы назовем прямыми методами решения задачи дробной оптимизации. В них необходимо провести прямые расчеты, но только с учетом, что функция $\varphi(x)$ является дробной и видоизменить некоторые формулы расчета. Процедурные аспекты таких методов остаются без изменения. В качестве примера можно привести симплекс-методы линейного программирования, когда для решения задачи дробно-линейного программирования проводятся лишь некоторые дополнительные вычисления и видоизменяется критерий оптимальности. Аналогичное место имеет и для полиномиальных алгоритмов решения задачи дробной оптимизации, для методов потенциалов решения дробно-линейной транспортной задачи, для метода эллипсоидов решения дробно-выпуклой задачи, и некоторые других задач дробного программирования.

Если же имеет место вторая ситуация, когда разработанные методы решения задачи $\min\limits_{x\in S}\varphi(x)$ не позволяют решить напрямую и задачу с дробными функционалами, то в таких случаях необходимо модифицировать данные методы или же разработать другие специальные методы решения задачи дробной оптимизации.

Рассмотрим задачу дробной оптимизации $\min\limits_{x\in S}\dfrac{\varphi(x)}{\psi(x)}$, в которой требуется минимизировать на множестве допустимых значений переменных $x\in S$ отношение двух функций $\varphi(x)$ и $\psi(x)$, притом предполагается, что $\psi(x)\neq 0$ для $\forall\, x\in S$. Для каждой отдельной постановки задачи дробной оптимизации имеется своя прототипная задача $\min\limits_{x\in S}\varphi(x)$ того же класса оптимизации. Тогда, если имеется метод решения дробной задачи определенного класса оптимизации, то в предположении, что $\psi(x)\equiv 1$ для $\forall\, x\in S$, такой метод, без его дополнительных модификаций, может быть использован и для решения задачи $\min\limits_{x\in S}\varphi(x)$, того же класса оптимизации. В таком случае, можем считать, что задача $\min\limits_{x\in S}\varphi(x)$ является частным случаем дробной задачи оптимизации $\min\limits_{x\in S}\dfrac{\varphi(x)}{\psi(x)}$, при $\psi(x)\equiv 1$ для $\forall\, x\in S$.

С другой стороны, задачу дробной оптимизации можем рассматривать как обобщение соответствующего класса задач оптимизации, когда предполагается, что функционал задачи является дробным, т.е. определяется отношением двух функций из того же класса задач оптимизации. В таком случае, имеющийся метод решения прототипной задачи $\min\limits_{x\in S}\varphi(x)$ обобщается и на случай решения дробной задачи оптимизации $\min\limits_{x\in S}\dfrac{\varphi(x)}{\psi(x)}$. По такому принципу было разработано большинство алгоритмов и методов решения задач дробной оптимизации, т.е. первоначально изучалась прототипная задача $\min\limits_{x\in S}\varphi(x)$, а затем полученные результаты и методы применялись с некоторой модификацией (адаптация метода) для решения дробной задачи оптимизации $\min\limits_{x\in S}\dfrac{\varphi(x)}{\psi(x)}$.

Другой прием, который был использован для решения задач дробной оптимизации $\min\limits_{x\in S}\dfrac{\varphi(x)}{\psi(x)}$, состоит в применение некоторых эквивалентных преобразований для их сведения к некоторой задаче оптимизации $\min\limits_{y\in Q}z(y)$, которая по отношению к первоначальной дробной задаче имеет другой набор переменных y из другого допустимого множество Q и с новым функционалом $z(y)$, который может быть из того же или другого класса задач оптимизации. В данном случае, вместо исходной задачи решается ее эквивалентная задача $\min\limits_{y\in Q}z(y)$, а потом восстанавливается решение исходной задачи $\min\limits_{x\in S}\dfrac{\varphi(x)}{\psi(x)}$.

Применение того или иного подхода к решению дробной задачи оптимизации $\min\limits_{x\in S}\dfrac{\varphi(x)}{\psi(x)}$ зависит от сложности задачи или от эффективности имеющиеся методов и алгоритмов решения прототипной задачи $\min\limits_{x\in S}\varphi(x)$ или эквивалентной задачи $\min\limits_{y\in Q}z(y)$ того же класса, или даже другого класса оптимизации. В качестве примера, может служить задача дробно-линейного программирования, для решения которой могут быть модифицированы симплекс-методы, задача может быть сведена к эквивалентной задачи линейного программирования, для решения которой можно использовать тот же симплекс-метод, либо задача может быть сведена к решению параметрической задачи линейного программирования с использованием некоторого итерационного метода нахождения корня соответствующего уравнения, на каждой итерации которого решается задача линейного программирования тем же симплекс-методом.

Следует отметить, что почти для всех классов задач дробной оптимизации $\min\limits_{x\in S}\dfrac{\varphi(x)}{\psi(x)}$ первоначально были исследованы и разработаны методы решения прототипных задач $\min_{x\in S}\varphi(x)$, и лишь потом данные методы были модифицированы для решения дробной задачи оптимизации, или же дробная задача была сведена путем эквивалентных преобразований к другой задачи оптимизации $\min_{y\in Q} z(y)$ из того или другого класса, для решения которой уже имелись эффективные методы. В качестве примеров, можем привести различные пары задач оптимизации, а именно: выпуклые и дробно-выпуклые задачи, квадратичные и дробно-квадратичные задачи, стохастические и дробно-стохастические задачи и тому подобные. Также исследованы пары таких задач, как транспортные задачи, распределительные задачи, задачи о назначениях, задачи дискретной или многокритериальной оптимизации, задачи теории графов и теории игр, в которых рассматриваются как не дробные, так и дробные функции оптимизации.

Одновременно с разработкой методов решения задачи дробной оптимизации в различных работах были рассмотрены вопросы определения необходимых и достаточных условии оптимальности, построения функций Лагранжа и двойственных задач. Также изучены вопросы выпуклости, квазивыпуклости и псевдовыпуклости дробных функций. В отдельных работах рассмотрены общие задачи квазивыпуклой оптимизации, в классе которых включены и задачи с дробными функциями, которые, как правило, обладают такими же свойствами выпуклости.

Аналогичные подходы были использованы при решении задач дробной оптимизации со специальной структурой целевой функции и системы ограничений, дробных задач теории графов и теории игр, дискретной и целочисленной оптимизации, многокритериальной и недифференцируемой оптимизации.

В данной монографии для решения задач дробного программирования использованы как модификации методов решения задач из аналогичного, не дробного, класса задач, так и приемы их сведения к задачам из другого класса задач оптимизации.

Задачи дробной оптимизации рассмотрены в многочисленных работах, которые проанализированы и классифицированы в библиографических обзорах [433], [488]–[493], [495]. Основные задачи дробного программирования и методы их решения приведены во многих монографических исследованиях [32, 36, 148, 359, 438, 493, 494, 633, 733], а также описаны в различных учебных пособиях [592, 655], которые содержат отдельные главы по дробному программированию. Вопросы дробной оптимизации рассмотрены на различных научных конференциях, среди которых можно выделить специализированную конференцию по обобщенной выпуклости [98, 158, 178, 240, 304, 314, 426, 448, 475], на которых рассмотрены различные доклады по вопросам дробной оптимизации [51, 55, 56, 62, 97, 100, 102, 236, 305, 404, 432, 437, 449, 472, 480, 497], квазивыпуклости [94, 202] и обобщенной выпуклости [106, 117, 443]. Также следует отметить тех работ, которые стояли у истоков дробной оптимизации. Одними из первых работ, в которых рассмотрены задачи дробно-линейного программирования относится к начало 60-х годах прошлого столетия. К ним относятся работы [133, 174, 222, 223, 264, 280, 702], [356]–[358], [360, 387, 508, 710], в которых предложены модификации симплекс-методов, параметрический метод и метод сведения дробно-линейной задачи к решению задач линейного программирования.

В дальнейшем вопросы дробной оптимизации изучались в различных работах, и как указано в приведенных выше библиографических обзорах, теоретические результаты по дробному программированию можно разделить на следующие области исследования:

- дробно-линейное программирование;
- нелинейное дробное программирование;
- теория двойственности в дробном программировании;
- анализ устойчивости задач дробного программирования;
- многокритериальное дробное программирование;
- параметрическое дробное программирование;
- сепарабельное дробное программирование;
- стохастическое дробное программирование;
- обобщенное дробное программирование;
- целочисленное дробное программирование;
- принципы декомпозиции в дробном программировании;
- задачи дробного программирования транспортного типа;
- приложения задач дробного программирования.

Как видно из перечисленных направлениях исследования, для каждого класса задач дробного программирования имеются свои прототипные (соответствующие) задачи оптимизации, но не с дробным функционалом. Большинство из полученных результатов исследования задач дробного программирования, основные методы их решения, а также их теоретические и практические аспекты тесно связаны с аналогичными задачами математического программирования.

Для подтверждения вышесказанного проведем сравнительный анализ по применению методов и алгоритмов решения пар из прототипной (не дробной) задачи $\min\limits_{x \in S} \varphi(x)$ и задачи дробной оптимизации $\min\limits_{x \in S} \dfrac{\varphi(x)}{\psi(x)}$, для тех задач дробной оптимизации, которые рассмотрены в данной монографии и для решения которых автором предложены эффективные методы и алгоритмы их решения, основанные на методах недифференцируемой оптимизации.

1. Задача дробно-линейного программирования. Методы решения задач дробно-линейного программирования основаны на идеях и методах решения ее прототипной задачи линейного программирования, которая впервые рассмотрено в работах [162, 163], [623]–[625]. Для решения такой задачи в данных работах предложен универсальный метод, который получил в дальнейшем название симплекс-методом. В различных монографиях и книгах по линейному программированию [43, 93, 146, 162, 166, 167, 173, 217, 241, 349, 388, 397, 485, 601, 602, 605, 625, 641, 740, 741] описаны и другие универсальные методы из класса симплексных, среди которых выделены модифицированный и двойственный метод, прямо-двойственные методы, а также методы с различными способами хранения и пересчета информации. В этих работах также приведены двойственные задачи и рассмотрена теория двойственности в линейном программировании. Предложены декомпозиционные методы [1, 162, 164, 165], методы внутренних точек [170, 377, 379], [610]–[612], и другие численные методы решения задачи линейного программирования. В дальнейшем были разработаны различные полиномиальные алгоритмы. Впервые класс таких алгоритмов основанные на методе эллипсоидов предложил Хачиан [293, 705, 706], а потом Кармаркар [285] предложил другой класс полиномиальных алгоритмов, основанные на операции проектирования пространства и соответствующие классу алгоритмов внутренних точек. На основе этих идей, и особенно на базе проективного алгоритма Кармаркара, разработаны различные модификации и варианты полиномиальных алгоритмов [22]-[26], [42, 35, 210], [218]–[220], [227, 254, 286, 287, 303, 378, 416, 420, 520, 643, 648]. Одной из таких модификаций предложено в работе [35]. Как отмечено в монографии [733], аналогичный алгоритм внутренних точек, но намного раньше, предложено в работе Дикина [610]. Полиномиальные алгоритмы типа Дикин-Кармаркар описаны и в общеизвестных работах [166, 167], в которых подведены итоги развития таких методов и определены дальнейшие направления исследования в области линейного программирования, связанные с повышением эффективности полиномиальных алгоритмов метода внутренних точек.

Для решения задач дробно-линейного программирования разработаны аналогичные методы и алгоритмы, как и для соответствующего класса прототипных линейных задач. При разработке таких методов и алгоритмов использованы четыре основные приема:

1) для решения дробной задачи модифицированы методы и алгоритмы линейных задач, которые в данной монографии названы прямыми методами;

2) задача дробно-линейного программирования преобразуется в одну или две задачи линейного программирования; такие методы названы методами линеаризации;

3) решение задачи дробно-линейного программирования сводится к анализу параметрической задачи линейного программирования и различными итеративными методами находится корень соответствующего уравнения от одного параметра; эти методы названы параметрическими методами;

4) решение задачи дробно-линейного программирования сводится к задачи недифференцируемой оптимизации с дальнейшем применением субградиентных методов, на каждом шаге которых решаются задачи линейного программирования. Эти методы названы методами Лагранжевой релаксации или схемами декомпозиции по ограничениям, переменным или ресурсам.

В результате применения таких приемов предложены следующие группы методов и алгоритмов решения задач дробно-линейного программирования:

- прямые симплекс-методы [32, 222, 223, 226, 305], [356]–[358], [493, 494, 508, 513, 531, 592, 635, 655, 710];

- методы линеаризации [32, 133, 493, 494, 577, 593, 601, 628, 673, 733];

- параметрические методы [4, 32, 85, 171, 280, 493, 494, 509, 525, 545, 546, 592, 733];

- прямые полиномиальные алгоритмы типа Хачиана-Кармаркара-Дикина [24, 593, 685, 733];

- прямые методы декомпозиции типа Данцига-Вулфа [16, 17, 112, 318, 320, 336, 624, 636, 663, 664, 667, 733];
- схемы декомпозиции по ограничениям, переменным и ресурсам с применением субградиентных методах [663]–[665], [667, 669, 670, 672, 674, 675, 733].

Для задач дробно-линейного программирования применена теория двойственности линейного программирования, в результате которой построены различные двойственные задачи [2, 46, 110, 152, 457, 460, 511]. В работах [11, 83, 132, 226, 531, 582, 590, 591], [666]–[668] рассмотрены другие вопросы и методы решения задач дробно-линейного программирования, а в работах [3, 32, 80, 99, 381, 493, 494, 733] проведен сравнительный анализ предложенных методов и алгоритмов в дробно-линейном программировании.

Из изложенного выше, можно сделать вывод, что задач дробно-линейного программирования имеется широкий диапазон методов и алгоритмов их решения, поэтому в практических целях выбирается тот, который соответствует требованию решаемой конкретной задачи, связанные с размерностью задачи, специфике и структуре ограничений, наличию программного обеспечения, требуемой точностью и времени решения.

2. Дробно-линейные транспортные задачи. Линейная транспортная задача рассмотрена почти во всех книгах по линейному программированию и исследованию операций [43, 93, 146, 162, 166, 167, 173, 217, 241, 247, 349, 388, 397, 485, 544], [586]–[588], [592, 593, 601, 602, 605, 625, 641, 650, 653, 699, 709], [739]–[741] в качестве примера задач оптимизации, для решения которых разработаны различные эффективные алгоритмы и методы. Среди таких методов можно выделить общеизвестные методы решения матричных и сетевых транспортных задач, такие как метод потенциалов, венгерский метод, метод Фогеля, метод ветвей и границ. Разные методы и алгоритмы рассмотрены в работах по транспортным задачам [543, 579, 583, 601, 637]. Также предложены и другие приближенные, эвристические и декомпозиционные методы, в которых учитывается специфика ограничений транспортных задач. Разработаны методы и алгоритмы решения различных линейных задач транспортного типа, таких как задача о назначениях, задача о коммивояжере, распределительная задача, динамическая и стохастическая транспортная задача. В работе [725] для решения сетевой транспортной задачи предлагается использовать субградиентный метод, а в работах [639, 640, 661] этот метод развит в виде схем декомпозиции по ограничениям и переменным для решения различных линейных задач транспортного типа большой размерности.

Линейные транспортные задачи являются прототипными для дробно-линейных транспортных задачах соответствующего класса. Так как транспортные задачи имеют специальную структуру ограничений, то предложенные методы и алгоритмы эффективны для решения задач только из своего класса. Поэтому в случае аналогичных транспортных задач с дробно-линейным функционалом, проводится модификация соответствующих методов и алгоритмов [36, 39, 237, 265, 506, 512, 664, 667, 672, 733].

Для решения дробно-линейных транспортных задач предложены различные методы и алгоритмы, среди которых метод потенциалов решения дробно-линейных транспортных задач, венгерский метод решения дробно-линейных задач о назначениях, метод потенциалов решения дробно-линейных распределительных задач, параметрические методы решения дробно-линейных задач транспортного типа, декомпозиционные методы решения дробно-линейных транспортных задач, а также схемы декомпозиции по ограничениям и переменным для решения дробно-линейных транспортных задач.

Для решения на ЭВМ дробно-линейных задач транспортного типа разработано соответствующее программное обеспечение [664], [667]–[679], [681, 686, 688, 733], которое использовано при решении различных практических задач оптимизации грузовых и пассажирских перевозок на автомобильном транспорте. Разработаны математические модели дробно-линейной оптимизации для решения задач оперативного и годового планирования, определения кольцевых маршрутов, контейнерных перевозок и решения других задач оптимизации на автомобильном транспорте [481, 664, 667, 680], [683]–[685], [687], [689]–[694], [733].

3. Задачи дробно-выпуклого программирования. Задачи выпуклого программирования, которые являются прототипом для задач с дробно-выпуклым функционалом, достаточно хорошо изучены, и в литературе по данным задачам [18, 92, 141, 143, 393, 394, 396, 409, 417, 418, 521, 600, 660] приводятся многочисленные теоретические и практические результаты в виде конкретных методов и алгоритмов. Изучение задач выпуклой оптимизации, в первую очередь, связано с выявлением и получением необходимых и достаточных условий оптимальности, с построением двойственных задач и соответствующей теории двойственности, с разработкой конкретных методов и алгоритмов, с проведением анализа эффективности и сходимости предложенных методов, с решением конкретных практических задач.

Относительно задач дробно-выпуклого программирования, полученные результаты для задач выпуклой оптимизации могут быть применены с учетом того факта, что дробно-выпуклые функционалы являются квазивыпуклыми или псевдовыпуклыми функциями [30, 94, 106, 181, 330, 406, 493, 494, 589, 629, 645, 695]. Поэтому для решения задач дробно-выпуклого программирования были предложены следующие группы алгоритмов и методов:

- прямые методы выпуклой оптимизации [37, 47, 48, 50, 151, 258, 266, 352, 439], которые адаптированы для решения дробных задач;

- параметрические методы [63, 172, 191, 260, 261, 362, 380, 436, 478], когда дробная задача сводится к решению последовательности выпуклых задач;

- методы внутренних точек [204, 206, 272, 390, 419], градиентные [205] и субградиентные методы [34, 300, 367, 375, 633], в частности метод эллипсоидов [633] для решения дробно-выпуклых задач, в которых решаются специальные задачи и выводятся формулы определения субградиентов, дающие направление движения;

- методы штрафных функций в дробном программировании [209, 633];

- декомпозиционные методы [107, 486] и схемы декомпозиции по переменным, ограничениям и ресурсам [496, 633, 667, 733] с применением методов недифференцируемой оптимизации.

Для задач дробно-выпуклого программирования рассмотрены двойственные задачи [8, 47, 63], [108]–[110], [195, 233, 270, 327, 384, 434, 435, 450, 453, 507, 526, 595, 596], рассмотрены условия оптимальности [341, 474, 564] и проведен анализ разработанных методов [51, 148, 150, 203, 263, 382, 383, 432, 437, 440, 441, 444, 446, 493, 494, 633].

4. Блочные задачи дробной оптимизации. Для решения задач линейного и выпуклого программирования большой размерности, в частности с блочно-диагональной структурой системы ограничений, разработаны различные декомпозиционные методы. В монографических работах [163, 166, 167, 335, 605, 635] приводятся основные принципы декомпозиции, которые относятся к классическим методам разложения Данцига-Вулфа [164, 165], процедурам расчленения Розена [421, 422], алгоритмам расчленения Бендерса [61], принципам распределения ресурсов Корнаи-Липтака [630], а также к другим методам и процедурам декомпозиции [594, 603, 621]. Другой класс декомпозиционных методов связаны с функциями Лагранжа, двойственными задачами и методами недифференцируемой оптимизации. К ним относятся схемы декомпозиции по переменным, ограничениям и ресурсам с применением субградиентных методов [470, 639, 640, 724, 726, 733]. Такие схемы использованы для решения задач линейного и выпуклого программирования, задач транспортного типа, задач дискретной, квадратичной, полиномиальной и стохастической оптимизации.

В данной книги декомпозиционные методы и схемы применены и для решения задач дробной оптимизации, в частности для задач дробно-линейного и дробно-выпуклого программирования, дробных задач транспортного типа, а также для решения обобщенных задач дробно-линейного и дробно-выпуклого программирования, дробных задач стохастического и квадратичного программирования.

Разработанные автором декомпозиционные методы решения блочных задач дробной оптимизации можно условно разделить на три группы:

- методы разложения Данцига-Вулфа [663, 664, 667, 733],
- методы распределения ресурсов Корнаи-Липтака [633, 733];
- схемы декомпозиции по переменным, ограничениям и ресурсам [663]–[665], [667, 669, 670, 672, 674, 675, 733].

Особенности, и тем самым сложности, применения таких декомпозиционных методов в дробном программировании связаны с тем, что дробный функционал не является аддитивно-сепарабельной функцией. Поэтому некоторые декомпозиционные методы разработаны в сочетание с параметрическим методом [171, 493, 494, 592, 733] анализа и решения задач дробной оптимизации.

5. Обобщенные задачи дробного программирования. Данные классы задач дробной оптимизации получены в результате применения операций взятия максимума или минимума от конечного числа дробных функций, суммирования или умножения дробных функций. Функции в таких операциях могут быть дробно-линейными, дробно-выпуклыми, или специальными функциями, например, дробные функции для задач транспортного типа или блочной структуры. Применение таких операций приводит к задачам дробного программирования, которые являются задачами глобальной и недифференцируемой оптимизации. Для решения таких задач можно применять общепринятые методы выпуклой, вогнутой или невыпуклой оптимизации, такие как методы внутренних точек [14, 18, 21, 204, 245, 272], [393]–[396], [548], методы ветвей и границ [377, 427], методы секущих плоскостей [29, 38, 472], а также и другие общеизвестные методы решения задач математического программирования [19, 20, 31, 44, 76, 78, 193, 213, 269], задач глобальной [88], [194]–[196], [255]–[257], [278, 521] или недифференцируемой оптимизации [12, 13, 33, 58, 141, 143, 274, 302, 351, 367, 376, 409, 469, 470, 569, 715].

Такие методы могут быть применены для решения общих задач обобщенного программирования с дробно-линейными или дробно-выпуклыми функциями. Однако, для решения конкретных дробных задач, разработаны специальные методы и алгоритмы, учитывающие специфику применяемой операции при получения обобщенных дробных функционалов, а также для решения обобщенных дробных задач транспортного типа или обобщенных задач блочной структуры с сепарабельными дробными функционалами.

Для обобщенных дробных минимаксных задач, которые, как правило, являются задачами квазивыпуклой оптимизации [30, 536], рассмотрены общие вопросы их анализа [27, 144, 189, 232, 331, 332, 407], приведены необходимые и достаточные условия оптимальности [12, 153, 328, 334, 339, 344, 345, 351, 372, 473, 476, 553, 563], [566]–[568], разработаны различные двойственные задачи [12, 13, 41, 52, 53, 58, 59, 113, 118, 147, 157, 236, 268, 297, 329, 330, 333, 342, 451, 454, 476, 541, 550, 552, 554, 562, 567, 568, 576] и для их решения предложены параметрические методы [52, 53, 90], [153]–[156], [633], методы частичной линеаризации [62, 153], субградиентные методы [484, 496, 633], а также другие методы их решения [40, 62, 75, 191, 231, 252, 455, 484, 487, 499, 505, 528, 567, 568, 633].

В данной монографии для решения минимаксных задач дробно-выпуклого и дробно-линейного программирования предлагаются прямые и двойственные параметрические методы, субградиентные методы, методы лагранжевой релаксации и схемы декомпозиции по ограничениям.

Обобщенные задачи оптимизации суммы дробных функций являются общими задачами математического программирования, так как при суммирование дробно-линейных функций или же дробно-выпуклых функций не получаем, как правило, выпуклый или квазивыпуклый функционал. В результате суммирования дробных функций получаем нелинейные, полиномиальные или степенные функционалы, которые приводят к задачам обобщенной выпуклости и глобальной оптимизации, для решения которых можно применить такие же общие методы, как и в случае минимаксных задач дробного программирования.

Для решения задач оптимизации суммы дробных функций предложены следующие группы методов:

- параметрические методы [15, 101, 176, 186, 445],
- методы внутренних точек [208],
- методы ветвей и границ [64], [67]–[70], [323, 325, 461, 533, 555],
- методы глобальной оптимизации [70, 71, 159, 462, 463],
- методы решения дробных задач с функционалами в виде суммы линейных и дробно-линейных функций [9, 10, 109, 248, 249, 355, 700], суммы двух [102, 631] или многих дробно-линейных функций [135, 136, 305, 306, 313, 323, 325, 533, 555, 585],
- другие вопросы и методы их анализа и решения [96, 109, 159, 250, 549].

В данной книге для решения задач оптимизации суммы дробно-выпуклых и дробно-линейных функций предлагаются нелинейные многопараметрические методы, методы лагранжевой релаксации нахождения седловой точке, алгоритмы частичной линеаризации, субградиентные методы решения прямой и двойственной задачи, а также схемы декомпозиции по ограничениям с применением методов недифференцируемой оптимизации.

Обобщенные задачи оптимизации произведения дробно-выпуклых функций (задачи мультипликативного дробного программирования) рассмотрены в работах [169, 251, 273, 308, 312, 447], в которых, по аналогии с выпуклыми мультипликативными задачами [65, 72, 216, 309, 310, 324], предложено использовать для их решения общие методы глобальной оптимизации.

В данной работе для решения задач оптимизации произведения дробно-выпуклых и дробно-линейных функций предлагаются нелинейные параметрические методы, алгоритмы метода частичной линеаризации, субградиентные методы решения прямой и двойственной задачи, методы лагранжевой релаксации и схемы декомпозиции по ограничениям.

Для обобщенных задач оптимизации отношения функций, полученных в результате применения операций взятия максимума и минимума от конечного числа выпуклых или линейных функций в данной работы приводятся эквивалентные задачи и предлагаются методы лагранжевой релаксации, субградиентные методы и схемы декомпозиции по ограничениям.

Другой класс обобщенных дробных задач, который рассмотрен в данной монографии относится к задачам многокритериальной (векторной) дробной оптимизации [111, 137, 168, 235, 291, 295, 385, 561, 605], для которых может быть применен общий подход для их анализа [79, 82, 87, 119, 180, 347, 348, 364, 400, 430, 442, 503, 634, 650, 653, 654] для нахождения эффективных решений. Однако, для дробных задач многокритериальной оптимизации рассмотрены необходимые и достаточные условия оптимальности [54, 138, 298, 299, 340, 343, 346, 398], приведены различные двойственные задачи [54, 56, 57, 91, 114, 116, 138, 179, 279, 292, 298, 299, 301, 321, 322, 340, 343, 346, 374, 386, 398, 412, 534], [536]–[539], рассмотрены недифференцируемые дробные многокритериальные задачи [54, 138, 296, 299, 301, 398, 346], а также предложены различные методов решения исходных многокритериальных задач или их сведения к однокритериальным задачам оптимизации [95, 102, 139, 207, 316, 350, 363, 399, 535].

В данной монографии предлагается применить метод сверток сведения исходной многокритериальной дробной задачи к одной из рассмотренных выше обобщенных задач дробного программирования. К дробным многокритериальным задачам применяются аддитивные, минимаксные или мультипликативные свертки. Также предложено применять смешанные и неоднотипные свертки для нахождения эффективных решений обобщенных многокритериальных дробных задач.

6. Специальные задачи дробного программирования. Данные задачи дробной оптимизации относятся к конкретным классам задач оптимизации, такие как дробные задачи сепарабельного и блочного программирования, целочисленной и дискретной оптимизации, динамического и квадратичного программирования, стохастической и полиномиальной оптимизации, и многие другие задачи дробной оптимизации, в которых необходимо учесть некоторые дополнительные требования к переменным, ограничениям или исходных данных задачи оптимизации.

В данной монографии рассмотрены специальные задачи дробной оптимизации, для решения которых эффективно можно применить схемы декомпозиции и методы недифференцируемой оптимизации. Такие схемы были использованы для решения задач дробно-квадратичного программирования, дробно-стохастической и дробно-полиномиальной оптимизации, а также для задач дробно-линейного программирования с неизвестными коэффициентами.

Предложенные декомпозиционные алгоритмы и методы решения задач дробного программирования основаны на теории недифференцируемой оптимизации и субградиентных методах, которые разработаны автором под руководством академика Н.З. Шора в сотрудничестве с учеными Института кибернетики имени В.М. Глушкова (Киев, Национальная Академия Украины) и Института математики и информатики (Кишинэу, Академия наук Молдовы). Основные результаты опубликованы автором в диссертационных работах [664, 667], монографиях [633, 733], учебных пособиях [592, 593] и научных статьях [482]–[663], [665, 666], [668]-[694], [487, 496]. Многие из них были использованы для решения различных практических задач оптимизации на автомобильном транспорте [481, 664, 667, 680], [683]–[685], [687], [689]–[694], [733].

1.2. Выпуклые и квазивыпуклые функции

Рассмотренные выше задачи дробной оптимизации содержат функции, которые являются линейными, выпуклыми или квазивыпуклыми. Разработанные методы решения этих задач основаны на свойствах таких функций. Ниже приведем основные понятия и свойства выпуклых и квазивыпуклых функций [493, 494, 607, 660, 733].

Пусть E^n – евклидово n-мерное пространство, а X – непустое выпуклое множество, и пусть задана действительная функция $f(x)$, определенная на выпуклом множестве $X \subset E^n$.

Графиком функции $f(x)$ называется подмножество $H[X, f]$ $(n+1)$-мерного пространства E^{n+1}, состоящего из точек $\{x, f(x)\}$, $x \in X$. Надграфиком функции $f(x)$ назовем множество $H^+[X, f] \in E^{n+1}$ точек вида $\{x, u\}$, где $u \geq f(x)$, $x \in X$.

Функция $f(x)$ называется выпуклой, если ее надграфик является выпуклым множеством. Эквивалентное определение дается в виде следующего утверждения.

Теорема 1.1. *Для того, чтобы функция $f(x)$, определенная на множестве $X \subset E^n$, была выпуклой, необходимо и достаточно, чтобы:*

1) *множество X было выпуклым;*

2) *для любых точек $x_1, x_2 \in X$ и числа $\alpha\,(0 \leq \alpha \leq 1)$ выполнялось соотношение*

$$\alpha\, f(x_1) + (1-\alpha)\, f(x_2) \geq f(\alpha\, x_1 + (1-\alpha)\, x_2). \tag{1.8}$$

Функция $f(x)$, определенная на выпуклом множестве $X \subset E^n$, называется квазивыпуклой, если для любых точек x_1, $x_2 \in X$ и числа α $(0 \le \alpha \le 1)$ выполняется соотношение

$$\max\,(f(x_1),\, f(x_2)) \ge f(\alpha x_1 + (1-\alpha)x_2). \tag{1.9}$$

Если функция $f(x)$ является квазивыпуклой на X, то множество $\{x :\, f(x) \le c\}$, $x \in X$ – выпукло для всех c и наоборот.

Функция $f(x)$ называется строго выпуклой или строго квазивыпуклой на множестве X, если условия (1.8) или (1.9) выполняются в виде строгого неравенства.

Функция $f(x)$ является явно квазивыпуклой на множестве X, если условие (1.9) выполняется в виде строгого неравенства и $f(x_1) \ne f(x_2)$.

Функция $f(x)$ является строго явно квазивыпуклой на множестве X, если условие (1.9) выполняется в виде строгого неравенства и $x_1 \ne x_2$.

Функция $f(x)$ является вогнутой (квазивогнутой) на выпуклом множестве X, если функция $-f(x)$ выпукла (квазивыпукла).

Функция $f(x)$ является монотонной на множестве X, если она является как квазивыпуклой, так и квазивогнутой на X, т.е. для любых точек x_1, $x_2 \in X$ и числа α $(0 \le \alpha \le 1)$ одновременно выполняются соотношения

$$\min[f(x_1),\, f(x_2)] \le f(\alpha\, x_1 + (1-\alpha)x_2) \le \max[f(x_1),\, f(x_2)].$$

Если $f(x)$ является дифференцируемой в точке $x_0 \in X$, тогда

$$\nabla f(x_0) = \left(\frac{\partial f(x_0)}{\partial x_1}, \frac{\partial f(x_0)}{\partial x_2}, \ldots, \frac{\partial f(x_0)}{\partial x_n}\right)$$

называется градиентом функции f в точке x_0.

Функция $f(x)$ является псевдовыпуклой на множестве X, если функция f дифференцируема на X, и тогда для любых точек $x_1, x_2 \in X$, для которых выполняется неравенство

$$(x_1 - x_2)\nabla f(x_2) \ge 0,$$

следует что

$$f(x_1) \ge f(x_2). \tag{1.10}$$

Функция $f(x)$ является строго псевдовыпуклой на множестве X, если для любых точек x_1, x_2, $\in X$ и $x_1 \ne x_2$ неравенство (1.10) выполняется строго.

Рассмотрим вопросы экстремальности выпуклых, псевдовыпуклых и квазивыпуклых дифференцируемых функций.

Пусть $f(x)$ непрерывная, дифференцируемая функция определенная на выпуклом множестве $X \subseteq R^n$. Тогда имеют место следующие утверждения.

Теорема 1.2. *Пусть X выпуклое множество. Тогда для того чтобы функция $f(x)$ была выпуклой на X, необходимо и достаточно выполнения неравенства*

$$f(x) - f(y) \geq (\nabla f(y), x - y), \quad \forall\, x, y \in X.$$

Теорема 1.3. *Пусть X выпуклое множество и пусть X^*– множество точек минимума функции $f(x)$ на X. Тогда в любой точке $x^* \in X^*$ должно выполняться неравенство*

$$(\nabla f(x^*), x - x^*) \geq 0, \quad \forall\, x \in X.$$

Если же $x^* \in \operatorname{int} X$, тогда в точке x^* имеет место равенство $\nabla f(x^*) = 0$.

Если, кроме того, $f(x)$ является выпуклой на X, то для того чтобы точка $x^* \in X^*$ необходимо и достаточно выполнения неравенства

$$(\nabla f(x^*), x - x^*) \geq 0, \quad \forall\, x \in X.$$

Имеют место следующие следствия.

Следствие 1.1. *Если функция $f(x)$ является псевдовыпуклой, то $x^* \in X^*$ тогда и только тогда, когда*

$$(\nabla f(x^*), x - x^*) \geq 0, \quad \forall\, x \in X.$$

Следствие 1.2. *Если функция $f(x)$ является квазивыпуклой на X, то X^* является выпуклым множеством.*

Следствие 1.3. *Если функция $f(x)$ является строго выпуклой на X, то множество X^* содержит не более одной точки.*

Теорема 1.4. *Пусть X выпуклое множество, а $f(x)$ определенная на X функция. Тогда:*

а) *если функция $f(x)$ является строго выпуклой на X, то глобальный минимум функции $f(x)$ на X является единственным;*

б) *если функция $f(x)$ является строго псевдовыпуклой на X, то глобальный минимум функции $f(x)$ на X является единственным;*

в) *если функция $f(x)$ является явно строго квазивыпуклой на X, то глобальный минимум функции $f(x)$ на X является единственным.*

Теорема 1.5. *Пусть X непустое, компактное и выпуклое множество. Если $f(x)$ является явно квазивыпуклой функцией на X, то $f(x)$ достигает своего глобального максимума в одном или в несколько экстремальных точках множества X.*

В дальнейшем приведем некоторые свойства выпуклых и квазивыпуклых функций [493, 494, 633, 733], которые использованы в данной работе.

Теорема 1.6. *Пусть $\{f_i(x)\}_{i=1}^k$ – семейство выпуклых функций, определенных на выпуклом множестве $X \subset E^n$; $c_1, \ldots, c_k$ – неотрицательные числа, тогда функции*

$$f(x) = \sum_{i=1}^{k} c_i f_i(x); \quad g(x) = \max_i c_i f_i(x); \quad g_i^+(x) = \max\{f_i(x); 0\} \ (i = \overline{1, k})$$

также являются выпуклыми в области определения X.

Теорема 1.7. *Пусть $\{f_i(x)\}_{i=1}^k$ – семейство квазивыпуклых функций, определенных на выпуклом множестве $X \subset E^n$; $c_1, \ldots, c_k$ – неотрицательные числа, тогда функция*

$$g(x) = \max_i c_i f_i(x)$$

также является квазивыпуклой в области определения X.

Теорема 1.8. *Пусть $\{f_i(x)\}_{i=1}^k$ – семейство квазивыпуклых функций, определенных на выпуклом множестве $X \subset E^n$, а функция*

$$f(x) = \sum_{i=1}^{k} c_i f_i(x)$$

определена на X. Тогда, если функции $f_i(x)$ таковы, что на любом отрезке $[x_1, x_2], x_1, x_2 \in X$ из неравенства $f(x_1) \leq f(x_2)$ следует выполнение неравенств $f_i(x_1) \leq f_i(x_2)$ для всех $i = \overline{1, k}$ одновременно, то функция $f(x)$ является квазивыпуклой на X.

Теорема 1.9. *Пусть функция $\varphi(t)$ одной переменной выпукла и не убывает на $[a,\, b]$, а функция $\psi(x)$ выпукла на выпуклом множестве X, причем $\psi(x) \in [a,\, b]$ при всех $x \in X$. Тогда функция*

$$f(x) = \varphi\big(\psi(x)\big)$$

выпукла на X.

Рассмотрим некоторые утверждения, показывающие связь между выпуклыми и квазивыпуклыми функциями.

Следствие 1.4. *Пусть $\varphi(x)$ – непрерывная функция, определенная на выпуклом множестве $X \subset E^n$, притом $\varphi(x) > 0$, $x \in X$. Тогда функция*

$$\psi(x) = \frac{1}{\varphi(x)}$$

в области определения X является выпуклой, если $\varphi(x)$ вогнута, и вогнутой, если $\varphi(x)$ выпукла.

Следствие 1.5. *Пусть функция $g(x)$ выпукла на выпуклом множестве X. Тогда функция*

$$g^+(x) = \max\{g(x),\ 0\}$$

выпукла на X.

Следствие 1.6. *Если функция $g(x)$ выпукла на X и $g(x) \geq 0$, $\forall\, x \in$ $\in X$, то функция*

$$f(x) = \bigl(g(x)\bigr)^p$$

выпукла на X при всех $p \geq 1$.

Следствие 1.7. *Если функция $g(x)$ выпукла на X, то функция*

$$f(x) = \bigl(\max\{g(x), 0\}\bigr)^p = \bigl(g^+(x)\bigr)^p$$

выпукла на X при всех $p \geq 1$.

Утверждение 1.1. *Пусть $\varphi(x)$ – непрерывная функция, определенная на выпуклом множестве $X \subset E^n$, а линейная функция $h(x) = (d, x) + d_0 > 0$ для любого $x \in X$. Тогда функция*

$$\psi(x) = \frac{\varphi(x)}{h(x)},$$

определенная на X является квазивыпуклой, если $\varphi(x)$ выпукла, и квазивогнутой, если $\varphi(x)$ вогнута.

Утверждение 1.2. *Пусть $\varphi(x)$, $\psi(x)$ – непрерывные функции, определенные на выпуклом множестве $X \subset E^n$, притом $\varphi(x) \geq 0$ и $\psi(x) > 0$ для любого $x \in X$. Тогда функция*

$$f(x) = \frac{\varphi(x)}{\psi(x)}$$

в области определения X является квазивыпуклой, если $\varphi(x)$ выпукла, а $\psi(x)$ вогнута, и квазивогнутой, если $\varphi(x)$ вогнута, а $\psi(x)$ выпукла.

Утверждение 1.3. *Пусть $\varphi(x)$, $\psi(x)$ – непрерывные функции, определенные на выпуклом множестве $X \subset E^n$, притом $\varphi(x) \geq 0$ и $\psi(x) \geq 0$ для любого $x \in X$. Тогда функция*

$$f(x) = \varphi(x) \cdot \psi(x),$$

определенная на X является квазивыпуклой, если функции $\varphi(x)$ и $\psi(x)$ вогнуты, и квазивогнутой, если функции $\varphi(x)$ и $\psi(x)$ выпуклы.

Утверждение 1.4. *Пусть на выпуклом множестве $X \subset E^n$ определена дробно-линейная функция*

$$f(x) = \frac{C(x)}{D(x)} = \frac{(c, x) + c_0}{(d, x) + d_0}$$

и предположим, что $D(x) > 0$ для любого $x \in X$, где c, d, x – n-мерные векторы, а c_0 и d_0 – константы. Тогда функция $f(x)$ является одновременно квазивыпуклой и квазивогнутой на выпуклом множестве X.

Пусть $\{\varphi_i(x)\}_{i=1}^k$ – семейство непрерывных неотрицательных выпуклых функций, определенных на $X \subset E^n$; $\{\psi_i(x)\}_{i=1}^k$ – семейство непрерывных положительных вогнутых функций, определенных на $X \subset E^n$. Определим функции $f(x) = \sum_{i=1}^{k} f_i(x)$ и $g(x) = \max_i f_i(x)$, в которых $f_i(x) = \dfrac{\varphi_i(x)}{\psi_i(x)}$. Тогда на основании теорем 1.7 и 1.8 функции $f(x)$ и $g(x)$ являются квазивыпуклыми на X.

Рассмотрим частные случаи вышеизложенных утверждений, когда функции $\varphi(x)$ и $\psi(x)$ определяются с помощью операции взятия максимума и минимума (поточечного) от выпуклых или вогнутых функций. Тогда имеют место утверждения:

Утверждение 1.5. *Если* $\{\psi_i(x)\}_{i=1}^k$ *– семейство вогнутых и положительных функций на* X*, то функция*

$$g(x) = 1/\min_i \psi_i(x)$$

выпукла на X*;*

Утверждение 1.6. *Если* $\{\psi_i(x)\}_{i=1}^k$ *– семейство выпуклых и положительных функций на* X*, то функция*

$$g(x) = 1/\max_i \psi_i(x)$$

вогнута на X*;*

Утверждение 1.7. *Если* $\varphi(x)$ *– выпуклая функция на* X, $\{\varphi_i(x)\}_{i=1}^k$ *– семейства выпуклых неотрицательных функций на* X, $h(x)$ *– линейная положительная функция на* X, $\{h_i(x)\}_{i=1}^k$ *– семейство линейных положительных функций, то функции*

$$g(x) = \frac{\varphi(x)}{\min\limits_i h_i(x)}, \quad f(x) = \max_i \frac{\varphi_i(x)}{h_i(x)},$$

$$p(x) = \frac{\max\limits_i \varphi_i(x)}{h(x)}, \quad q(x) = \frac{\max\limits_i \varphi_i(x)}{\min\limits_i h_i(x)}$$

квазивыпуклы на X*;*

Утверждение 1.8. *Если* $\{\varphi_i(x)\}_{i=1}^k$ *– семейство выпуклых неотрицательных функций на* X, $\{\psi_i(x)\}_{i=1}^k$ *– семейство вогнутых положительных функций на* X*, то функция*

$$g(x) = \frac{\max\limits_i \varphi_i(x)}{\min\limits_i \psi_i(x)}$$

квазивыпукла на X*.*

1.3. Свойства обобщенных градиентов

Рассмотрим определения и основные свойства обобщенных градиентов и субградиентных множеств для выпуклых и квазивыпуклых функциях [607, 633, 639, 715, 733].

Пусть функция $f(x)$ определена и выпукла на всем пространстве E^n, а X – некоторое выпуклое множество из E^n. Тогда через любую точку $\{x, f(x)\} \in H[X, f]$ можно провести опорную гиперплоскость к надграфику $H^+[X, f]$. Если точка y является внутренней для X, то уравнение этой опорной гиперплоскости имеет вид $(p, x - y) = z - f(x)$, при этом для всех точек надграфика $\{x, z\}$ должно выполняться соотношение $z - f(y) \geq (p, x - y)$. Если $z = f(x)$, то для всех $x \in X$

$$f(x) - f(y) \geq (p, x - y). \tag{1.11}$$

Произвольный вектор p, удовлетворяющий неравенству (1.11) для всех $x \in X$, называется субградиентом (обобщенным градиентом) функции $f(x)$ в точке y [639, 715, 733].

Назовем субградиентным множеством (множество обобщенных градиентов) выпуклой функции $f(x)$ в точке y множество векторов p, удовлетворяющих соотношению (1.11), и обозначим его $G_f(y)$, а его представителей – $g_f(y)$.

Теорема 1.10. *В любой внутренней точке области определения X выпуклой функции $f(x)$ существует не пустое, выпуклое, замкнутое и ограниченное множество субградиентов. Если $G_f(y)$ состоит из единственной точки $g_f(y)$, то $g_f(y)$ является градиентом функции $f(x)$ в точке y; в противном случае – в точке y функция $f(x)$ недифференцируема.*

Из (1.11) следует, что если внутренняя точка y является точкой минимума $f(x)$ на X, то $0 \in G_f(y)$. Справедливо и обратное: если $0 \in G_f(y)$, то y является точкой минимума выпуклой функции $f(x)$.

Пусть для некоторых $y \in X, t > 0, e \in E^n, e \neq 0$ выполняется соотношение $y + te \in X$. Если существует

$$\lim_{t \to +0} \frac{f(y + te) - f(y)}{t} = f'_e(y),$$

то $f'_e(y)$ называется производной по направлению e функции $f(x)$ в точке y.

Теорема 1.11. *В любой внутренней точке y области определения X выпуклой функции $f(x)$ существует производная $f'_e(y)$ по любому направлению $e \in E^n$; при этом $f'_e(y)$ связана с субградиентным множеством $G_f(y)$ соотношением*

$$f'_e(y) = \max_{g \in G_f(y)} (g, e).$$

Назовем направлением наискорейшего спуска в точке $y \in X$ направление η^*, на котором достигается минимум $f'_\eta(y)$ при $\|\eta\| = 1$.

Теорема 1.12. *Если точка y является внутренней и $G_f(y)$ не содержит 0 (т.е. точка y не является точкой минимума), то вектор $\eta^* = -g^*(y)/\|g^*(y)\|$, где $g^*(y) \in G_f(y)$ – ближайшая к 0 точка из субградиентного множества, является единственным направлением наискорейшего спуска.*

Пусть $f(x)$ выпуклая функция определенная на $E^n, y \in E^n$, $\varepsilon \geq 0$. ε-субградиентом $g_f^\varepsilon(y)$ функции $f(x)$ в точке y называется произвольный вектор g, удовлетворяющий неравенству

$$f(x) - f(y) \geq (g, x - y) - \varepsilon,$$

при любом $x \in E^n$.

Совокупность ε-субградиентов функции f в точке y образует ε-субградиентное множество $G_f^\varepsilon(y)$. При $\varepsilon = 0$ ε-субградиентное множество совпадает с субградиентном.

Для любого $y \in E^n$ и $\varepsilon \geq 0$ множество $G_f^\varepsilon(y)$ непусто, замкнуто и ограничено.

При $\varepsilon_1 > \varepsilon_2$ справедливо включение

$$G_f^{\varepsilon_1}(y) \supset G_f^{\varepsilon_2}(y).$$

ε-производная по направлению η в точке y называется величина

$$\frac{\partial_\varepsilon f(y)}{\partial \eta} = \max_{v \in G_f^\varepsilon(y)} (v, \eta).$$

Справедливо соотношение

$$\frac{\partial_\varepsilon f(y)}{\partial \eta} = \inf_{\alpha > 0} \frac{f(y + \alpha\eta) - f(y) + \varepsilon}{\alpha}.$$

Пусть $\varepsilon \geq 0$. Точка $\overline{x} \in E^n$ называется ε-стационарной, если $0 \in G_f^\varepsilon(\overline{x})$. Из определения понятия ε-субградиента вытекает, что $f(\overline{x}) - f^* \leq \varepsilon$, где $f^* = \inf\limits_{x \in E^n} f(x)$. Если точка y не является ε-стационарной, то можно определить направление ε-наискорейшего спуска η^* как направление, дающее минимальное значение ε-производной среди направлений $\eta, ||\eta|| = 1$. Это направление может быть получено тем же путем, что и направление наискорейшего спуска, только роль $G_f(y)$ играет $G_f^\varepsilon(y)$.

Непрерывная функция $f(x)$, определенная на E^n, называется *почти-дифференцируемой*, если она локально липшицева (из этого свойства следует существование градиента почти во всех точках E^n) и непрерывно дифференцируема на множестве, где градиент существует.

Пусть $\{x_i\}_{i=1}^{\infty} \to x_0$ и в точках x_i определен градиент $g_f(x_i)$ функции $f(x)$ $(i = 1, 2, \dots)$. Тогда *почти-градиентом* функции $f(x)$ в точке x_0 называется вектор, являющийся предельной точкой последовательности градиентов $g_f(x_1), g_f(x_2), \dots$, где $\{x_k\}_{k=1}^{\infty} \to x_0$. Если $f(x)$ – выпуклая функция в E^n, то она является почти-дифференцируемой, а ее любой почти-градиент в точке x_0 совпадает с некоторым ее субградиентом. Пусть $G_f(x_0)$ – множество почти-градиентов $f(x)$ в точке x_0. Вектор $g \in conv\, G_f(x_0)$ назовем обобщенным градиентом функции $f(x)$ в точке x_0.

Рассмотрим некоторое понятие обобщенного градиента для псевдовыпуклых и квазивыпуклых функций. Пусть задана квазивыпуклая функция $f(x)$, определенная на выпуклом множестве $X \subset E^n$. Тогда надграфик $H^+[X, f]$ квазивыпуклой функции $f(x)$ не является выпуклым множеством. Поэтому не для любой точки $\{x, f(x)\} \in H[X, f]$ можно провести опорную гиперплоскость к надграфику $H^+[X, f]$. Таким образом, не для всех точек надграфика $\{x, z\} \in H^+[X, f]$ выполняется соотношение $z - f(y) \geq (p, x - y)$, и тем самым, в таких точках не для всех $x \in X$ выполняется соотношение (1.11), определяющее обобщенный градиент. В связи с этим, рассмотрим некоторое понятие "обобщенного градиента", которое может быть взято в качестве направления спуска при минимизации квазивыпуклых функций.

Пусть R – некоторое выпуклое множество точек из вещественного пространства E^n. Множество

$$K(R, y) = \{p \in R : \ (p, x - y) \geq 0, \ x \in R\}$$

называется конусом опорных функционалов к множеству R в точке y. Если $\operatorname{int} R \neq \varnothing$ и $y \in \operatorname{int} R$, то в конусе $K(R, y)$ существуют ненулевые элементы. Пусть f – непрерывный квазивыпуклый функционал, определенный в E^n, D – выпуклое множество из E^n, $\operatorname{int} D \neq \varnothing$. Введем обозначения:

$$D(y) = \{x \in \operatorname{int} D : \ f(x) < f(y)\};$$

$$G(y) = \{x \in E^n : \ f(x) < f(y)\};$$

$$W(f, y) = -K(G(y), y).$$

Элементы множества $W(f,y)$ называются также обобщенно-опорными функционалами к функционалу f в точке y. Отметим, что в данном случае множество $G(y)$ выпукло и открыто, поэтому в конусе $W(f,y)$ для любого $y \in E^n$ существуют нулевые элементы. Обозначим:

$$S(y) = W(f,y) \cap S(0,1);$$

$$V(y) = \overline{conv}\, S(y),$$

где $S(0,1)$ единичный шар с центром в точке 0.

Установим некоторые свойства элементов конуса $W(f,y)$, подобные свойствам субградиентов выпуклых функционалов [619, 629].

Теорема 1.13. *Если функционал f является квазивыпуклым, слабо полунепрерывным сверху и липшицевым с константой L, то для любого $y \in E^n$ выполняется неравенство*

$$f(x) - f(y) \geq L \cdot \left(\frac{q}{\|q\|}, x - y \right) \tag{1.12}$$

для всех $x \in G(y)$ и всех ненулевых $q \in W(f,y)$.

Следствие 1.8. *При условиях теоремы* 1.13 *для любой точки $y \in E^n$ выполняется неравенство*

$$f(x) - f(y) \geq L \cdot (q, x - y) \tag{1.13}$$

для всех $x \in G(y)$ и всех $q \in V(y)$.

Следствие 1.9. *При условиях теоремы* 1.13 *соотношение* (1.13) *выполняется для всех $x \in D(y)$ и всех $q \in V(y) \setminus K(D,y)$.*

По аналогии с (1.11), векторы q, удовлетворяющие неравенствам (1.12) и (1.13), могут быть использованы для определения направления спуска в методах минимизации квазивыпуклых функционалов [629], т.е. векторы q играют роль "обобщенных градиентов" недифференцируемых квазивыпуклых функций.

1.4. Задачи оптимизации и метод множителей Лагранжа

Задача оптимизации функции нескольких переменных на множестве K, являющимся подмножеством пространства R^n, называется задачей условной оптимизации. В общем виде ее можно представить так:

найти min (минимальные) или max (максимальные) значения функции

$$z = f(x_1, x_2, \ldots, x_n) = f(x),$$

где $x = (x_1, x_2, \ldots, x_n) \in K$, $K \subset R^n$. Точки, в которых достигаются минимальные или максимальные значения, являются точками минимума или максимума функции $f(x)$, которые могут быть точками локального или глобального экстремума.

Множество K в практических задачах описывается системой равенств и неравенств. Если K совпадает с R^n, то получаем задачу безусловной оптимизации. Будем предполагать, что K – подмножество пространства R^n.

Очевидно, что для задачи условной оптимизации имеют место классические необходимые и достаточные условия оптимальности, если ее локальный минимум x^* является внутренней точкой множества K. Однако для многих задач условной оптимизации минимум целевой функции $z = f(x)$ достигается именно на границе множества K, в силу чего классические методы анализа не всегда применимы, и тем самым все вопросы оптимизации для таких задач становятся более сложными.

Рассмотрим классические методы решения задач на условный экстремум, которые имеют вид

$$\min z = f_0(x), \tag{1.14}$$

$$f_i(x) = 0 \quad (i = \overline{1, m}), \tag{1.15}$$

т.е. множество K определяется как множество решений системы уравнений $f_i(x) = 0 \quad (i = \overline{1, m})$.

При исследовании этой классической задачи на условный экстремум важную роль играет ее функция Лагранжа:

$$L(x, u_0, u) = u_0 f_0(x) + \sum_{i=1}^{m} u_i f_i(x), \tag{1.16}$$

где $x = (x_1, x_2, \ldots, x_m) \in R^n$, u_0 – число, а $u = (u_1, u_2, \ldots, u_m) \in R^m$.

Частные производные функции Лагранжа (1.16) по координатам x_j имеют вид

$$\frac{\partial L}{\partial x_j} = u_0 \frac{\partial f_0}{\partial x_j} + \sum_{i=1}^{m} u_i \frac{\partial f_i}{\partial x_j} \quad (j = \overline{1, n}).$$

Необходимые условия локальной оптимальности содержатся в следующей теореме.

Теорема 1.14. *Пусть функции $f_0(x), f_1(x), \ldots, f_m(x)$ непрерывно дифференцируемы в окрестности точки $x^* \in R^n$. Если x^* – точка локального минимума, то существуют числа $u_0^*, u_1^*, \ldots, u_m^*$, не равные одновременно нулю, и такие, что*

$$\frac{\partial L}{\partial x_j} = u_0^* \frac{\partial f_0}{\partial x_j} + \sum_{i=1}^{m} u_i^* \frac{\partial f_i}{\partial x_j} = 0 \quad (j = \overline{1, n})$$

в точке x^*. Если при этом градиенты $grad\, f_1(x^*), grad\, f_2(x^*), \ldots$ $\ldots, grad\, f_m(x^*)$ линейно независимы, то $u_0 \neq 0$.

Линейная независимость градиентов функций $f_1(x), f_2(x), \ldots, f_m(x)$ в точке x^* называется условием регулярности. Условия $\dfrac{\partial L}{\partial x_j} = 0$ $(j = \overline{1, n})$ означают что градиенты функций $f_0(x)$ и $f_1(x), f_2(x), \ldots, f_m(x)$ в точке x^* линейно зависимы.

Числа $u_0^*, u_1^*, \ldots, u_m^*$ называются множителями Лагранжа. Любая допустимая точка x^*, удовлетворяющая необходимым условиям оптимальности задачи условной оптимизации, называется стационарной точкой этой задачи. Таким образом, стационарные точки определяются как решения системы из $n + m$ уравнений с $n + m + 1$ неизвестными $x_1, x_2, \ldots, x_n$ и $u_0, u_1, u_2, \ldots, u_m$, т.е. системы

$$\frac{\partial L}{\partial x_j} = u_0 \frac{\partial f_0}{\partial x_j} + \sum_{i=1}^{m} u_i \frac{\partial f_i}{\partial x_j} = 0 \quad (j = \overline{1, n}), \tag{1.17}$$

$$f_i(x) = 0 \quad (i = \overline{1, m}). \tag{1.18}$$

В случае $u_0^* \neq 0$ можно всегда считать $u_0^* = 1$. Для этого следует все множители Лагранжа $u_1^*, u_2^*, \ldots, u_m^*$ поделить на u_0^*. Таким образом, если выполняется условие регулярности, то функция Лагранжа имеет вид

$$L(x, u) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x).$$

Во многих случаях решение системы уравнений (1.17), (1.18) в явном виде представляет собой весьма сложную задачу. Только в простейших случаях удается найти в явном виде решение данной системы.

Однако не всегда стационарные точки обязаны быть решениями задачи. Для выбора из них оптимальных применяются достаточные условия с привлечением вторых производных. Эти условия сформулируем в виде следующей теоремы.

Теорема 1.15. *Пусть функции $f_0(x), f_1(x), \ldots, f_m(x)$ дважды дифференцируемые в стационарной точке x^* и пусть*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 L}{\partial x_i \partial x_j} dx_i\, dx_j > 0$$

при любых ненулевых приращениях dx_i и dx_j, таких, что

$$\sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j} dx_j = 0 \quad (i = \overline{1, m}),$$

где $\dfrac{\partial^2 L}{\partial x_i \partial x_j}$ и $\dfrac{\partial f_i}{\partial x_j}$ вычислены при $x = x^*$ и $u_0^*, u_1^*, \ldots, u_m^*$. Тогда x^* – точка строгого локального минимума рассматриваемой задачи.

Таким образом, метод множителей Лагранжа включает в себя два этапа.

Этап 1. Составление и решение системы уравнений (1.17), (1.18) для нахождения стационарных точек (необходимые условия оптимальности).

Этап 2. Исследование достаточных условий (см. теорему 1.15) для каждой найденной стационарной точки.

Метод множителей Лагранжа имеет ограниченное практическое применение в силу часто непреодолимых трудностей. Поэтому на практике используют приближенные численные методы решения задач условной оптимизации с учетом особенностей тех или иных типов задач.

1.5. Теория двойственности в нелинейном программировании

Для любой задачи нелинейного программирования можно построить некоторую другую задачу нелинейной оптимизации, тесно связанную с исходной. Исходная задача называется прямой задачей, а вторая – двойственной к ней.

Рассмотрим задачу нелинейного программирования: найти минимум функции

$$z = f_0(x), \tag{1.19}$$

при ограничениях

$$f_i(x) \le 0 \quad (i = \overline{1, m}), \tag{1.20}$$

$$x \ge 0. \tag{1.21}$$

Известны различные постановки задачи, называемой двойственной к данной. Среди них особое место занимает двойственная по Лагранжу задача. Она приводит к различным алгоритмам решения как линейных, так и нелинейных задач.

Рассмотрим функцию

$$L(x,u) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x),$$

которая называется функцией Лагранжа для рассматриваемой задачи нелинейного программирования, где $u = (u_1, u_2, \ldots, u_m)$ – множители Лагранжа и $u_i \geq 0 \ \ (i = \overline{1,m})$.

Определим функции

$$F(x) = \max_{u \geq 0} L(x,u) \quad \text{и} \quad G(u) = \min_{x \geq 0} L(x,u).$$

Тогда задачу отыскания минимума функции $F(x)$ при $x \geq 0$ называют прямой задачей, а задачу максимизации функции $G(u)$ при $u \geq 0$ – двойственной. Таким образом, прямая задача – это задача

$$\min_{x \geq 0} F(x) = \min_{x \geq 0} \max_{u \geq 0} L(x,u),$$

а двойственная – это задача

$$\max_{u \geq 0} G(u) = \max_{u \geq 0} \min_{x \geq 0} L(x,u).$$

Говорят, что точка $(x^*,\, u^*)$ является седловой точкой функции Лагранжа $L(x,u)$ в области $x \geq 0, u \geq 0$, если

$$L(x^*,u) \leq L(x^*,u^*) \leq L(x,u^*)$$

для всех $x \geq 0$ и $u \geq 0$.

Теорема 1.16. *Пусть (x^*,u^*) – седловая точка функции Лагранжа. Тогда x^* является решением рассматриваемой задачи нелинейного программирования, причем справедливо следующее условие дополняющей нежесткости:*

$$\sum_{i=1}^{m} u_i^* f_i(x^*) = 0.$$

Отметим, что прямая задача нахождения минимума функции $F(x) = \max\limits_{u \geq 0} L(x, u)$ равносильна исходной задаче нелинейного программирования (1.19)–(1.21).

Действительно,

$$F(x) = \max_{u \geq 0} L(x, u) = \begin{cases} f_0(x), \text{ если } x \in K, \\ \infty, \text{ если } x \notin K. \end{cases}$$

Отсюда следует, что $\min F(x)$ находится в допустимой области K и совпадает с минимумом функции $f_0(x)$, т.е.

$$\min_{x \geq 0} \max_{u \geq 0} L(x, u) = \min_{x \in K} f_0(x).$$

Для пары двойственных задач имеют место следующие утверждения:

1) при любых $x \geq 0$, $u \geq 0$ имеет место неравенство

$$G(u) \leq F(x);$$

2) если функция Лагранжа имеет седловую точку и выполнено условие Слейтера, то обе задачи имеют оптимальные решения, и имеет место равенство $\max\limits_{u \geq 0} G(u) = \min\limits_{x \geq 0} F(x)$.

Отметим, что в процессе формирования двойственных задач вместо условия $x \geq 0$ можно взять любое множество S, которое задается частью ограничений данной задачи нелинейного программирования, что приводит к неоднозначности полученных двойственных задач, т.е. двойственную задачу можно сформировать различными способами.

Изложенные идеи теории двойственности позволяют построить ряд важных численных методов решения задач нелинейного программирования.

Задача минимизации функции

$$F(x) = \max_{u \geq 0} L(x, u) \quad \text{при} \quad x \in S,$$

эквивалентная задаче нелинейного программирования

$$\min z = f_0(x),$$

$$f_i(x) \leq 0 \quad (i = \overline{1, m}),$$

$$x \in S,$$

кажется более привлекательной по сравнению с данной задачей нелинейного программирования, так как снимаются ограничения $f_i(x) \leq 0$. Однако задача $\min F(x)$ может оказаться более сложной, так как операция взятия максимума для функции Лагранжа может привести к тому, что функция $F(x)$ будет недифференцируемой, даже если функции

$f_i(x)$ $(i = \overline{0,m})$ являются дифференцируемыми. Но, формируя множество S так, чтобы оно было простым, и на S легко было бы осуществить необходимые операции, можно развить важные численные методы недифференцируемой оптимизации решения задач нелинейного программирования.

Приведем еще одно общее условие оптимальности в задачах нелинейного программирования.

Рассмотрим общую задачу оптимизации

$$\min z = f(x),$$
$$x \in K.$$

Говорят, что вектор $g \in R^n$ задает направление убывания функции $f(x)$ в точке $x^* \in R^n$, если $f(x^* + hg) < f(x^*)$ при всех достаточно малых $h > 0$. Множество таких g обозначим $U(x^*, g)$. Вектор g задает возможное направление относительно множества K допустимых решений в точке $x^* \in K$, если $x^* + hg \in K$ при достаточно малых $h > 0$. Множество g возможных направлений обозначим $V(x^*, K)$.

Имеет место следующее необходимое условие локального минимума функции $f(x)$ на множестве K, не требующее никаких предположений относительно K и $f(x)$:

Теорема 1.17. *Если x^* – локальное решение задачи оптимизации, то* $U(x^*, g) \cap V(x^*, K) = \varnothing$.

Это условие геометрически означает, что из точки x^* локального минимума нельзя осуществить сколь угодно малый сдвиг вдоль какого бы ни было луча так, чтобы уменьшить значение целевой функции, оставаясь при этом в множестве K.

Эта теорема лежит в основе ряда других более содержательных условий оптимальности при дополнительных предположениях относительно K и $f(x)$, часть из которых будет изложена в дальнейшем применительно к задаче выпуклого программирования при разработке субградиентных методов недифференцируемой оптимизации.

1.6. Задачи выпуклого программирования

Среди задач нелинейного программирования, которые достаточно полно исследованы выделяют специальный тип задач условной оптимизации – задачи выпуклого программирования.

Задачу условной оптимизации: найти минимум функции

$$z = f(x), \quad \text{где } x \in K$$

называют задачей выпуклого программирования, если:

- область K допустимых решений является выпуклой;
- целевая функция $f(x)$ является выпуклой.

Пусть K – некоторое выпуклое множество в R^n и $f(x)$ – выпуклая функция на K. Тогда множество $S = \{x \in K\colon\ f(x) \le c\}$ является выпуклым при любом действительном c, где c – константа. С точки зрения локальных и глобальных экстремумов, для задачи выпуклого программирования имеет место

Теорема 1.18. *Пусть $z = f(x)$ – выпуклая функция на выпуклом множестве K. Тогда любой ее локальный минимум является глобальным.*

Следствие 1.10. *Если $f(x)$ – строго выпуклая функция на выпуклом множестве K, то ее глобальный минимум единственный.*

Следствие 1.11. *Если глобальный минимум выпуклой функции достигается в двух различных точках, то он достигается и в любой точке отрезка, соединяющего эти точки.*

Для выпуклой дифференцируемой функции $f(x)$ имеет место утверждение: *если в точке x^* имеем $grad\, f(x^*) = 0$, то x^* есть точка ее локального, а следовательно, и глобального минимума.*

При вогнутой целевой функции $z = f(x)$ имеем задачу максимизации функции $z = f(x)$ на выпуклом множестве K. Рассмотренные результаты можно сформулировать следующим образом: любой локальный максимум вогнутой функции $f(x)$ на выпуклом замкнутом множестве K является и глобальным; если $f(x)$ строго вогнутая функция, то глобальный максимум единственный. Градиент вогнутой функции $f(x)$ в точках максимума равен нулю, если $f(x)$ дифференцируемая функция.

Отметим, что часто в практических задачах выпуклого программирования выпуклое множество K задается системой ограничений $f_i(x) \le$ $\le b_i$ или $f_i(x) - b_i \le 0 \quad (i = \overline{1,m})$, где функции $f_i(x)$ – выпуклые функции.

Множество S точек x, удовлетворяющих условию $f(x) \le b$, является выпуклым, если это множество не пусто. Таким образом, каждое неравенство $f_i(x) \le b_i \quad (i = \overline{1,m})$ задает выпуклое множество S_i, если

$f_i(x)$ – выпуклая функция. Пересечение выпуклых множеств S_i выпукло. Таким образом, множество всех точек K (если K не пусто), удовлетворяющих условиям $f_i(x) \le b_i \quad (i = \overline{1, m})$, выпукло, если $f_i(x)$ – выпуклые функции. Если в задаче имеются и неравенства $x_j \ge 0$, то каждое такое неравенство задает полупространство, которое выпукло, и тогда множество всех допустимых решений K задачи выпуклого программирования также выпукло.

1.7. Функция Лагранжа, необходимые условия экстремума и двойственные задачи

Применение функции Лагранжа для анализа задачи математического программирования позволяет определить условия ее разрешимости, связанные с понятием седловой точки и двойственные задачи. Рассмотрим n-мерный вектор x, принадлежащий множеству R и m-мерный неотрицательный вектор $u \ge 0$. Пусть функция $L(x, u)$ выпукла по x на множестве R и вогнута по u на неотрицательном ортанте.

Пара (x^*, u^*) называется седловой точкой функции $L(x, u)$ на множестве всех $x \in R$ и $u \ge 0$, если $x^* \in R, u^* \ge 0$ и

$$L(x^*, u) \le L(x^*, u^*) \le L(x, u^*) \tag{1.22}$$

для всех $x \in R$ и $u \ge 0$.

Соотношение (1.22) можно записать также следующим образом:

$$L(x^*, u^*) = \max_{u \ge 0} \min_{x \in R} L(x, u) = \min_{x \in R} \max_{u \ge 0} L(x, u).$$

Рассмотрим случай, когда $R = \{x\colon x \ge 0\}$. Пусть функция $L(x, u)$ выпукла по x для всех $x \ge 0$, вогнута по u для всех $u \ge 0$ и непрерывно дифференцируема по x и по u.

Теорема 1.19. *Для того чтобы пара (x^*, u^*) $(x^* \ge 0,\ u^* \ge 0)$ была седловой точкой функции $L(x, u)$ в области $x \ge 0$, $u \ge 0$, необходимо и достаточно выполнение условий:*

$$\frac{\partial L^*}{\partial x} \ge 0, \quad \left(x^*, \frac{\partial L^*}{\partial x}\right) = 0, \quad x^* \ge 0,$$

$$\frac{\partial L^*}{\partial u} \le 0, \quad \left(u^*, \frac{\partial L^*}{\partial u}\right) = 0, \quad u^* \ge 0,$$

где

$$\frac{\partial L^*}{\partial x} = \frac{\partial L(x,u)}{\partial x}\Bigg|_{\substack{x = x^* \\ u = u^*}},$$

$$\frac{\partial L^*}{\partial u} = \frac{\partial L(x,u)}{\partial u}\Bigg|_{\substack{x = x^* \\ u = u^*}}.$$

Эти условия будут в дальнейшем использованы для определения седловой точки функции Лагранжа.

Теорема 1.20. *Пусть имеется задача выпуклого программирования: найти*

$$\min\, f_0(x) \tag{1.23}$$

при ограничениях

$$f_i(x) \le 0 \quad (i = \overline{1,m}), \tag{1.24}$$

$$x \in X \subset E^n, \tag{1.25}$$

где X – замкнутое выпуклое множество, $f_\tau(x)$ $(\tau = \overline{0,m})$ – выпуклые функции, определенные на E^n. Пусть $u = \{u_1, u_2, \ldots, u_m\}$ – множители Лагранжа для ограничений (1.24),

$$L(x,u) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x)$$

– функция Лагранжа и пусть выполняется условие Слейтера: существует такая точка $\overline{x} \in X$, что $f_i(\overline{x}) < 0$ для $i = \overline{1,m}$. Тогда, для того чтобы x^* было оптимальным решением задачи (1.23)–(1.25), *необходимо и достаточно существование такого $u \ge 0$, что (x^*, u^*) является седловой точкой функции Лагранжа на множестве $X \times \{u : \; u \ge 0\}$, т.е.*

$$L(x^*, u) \le L(x^*, u^*) \le L(x, u^*)$$

при $x \in X$, $u \ge 0$. При этом $u_i^* f_i(x^*) = 0 \;\; (i = \overline{1,m})$.

Если $f_0(x)$ и $f_i(x)$ – дифференцируемые функции, то можно доказать, что условия теоремы 1.20 эквивалентны следующим условиям оптимальности Куна-Таккера, представленные в виде системы уравнений:

1) $\dfrac{\partial L}{\partial x_j} - v_j = 0 \;\; (j = \overline{1,n})$ при условиях $x_j \cdot v_j = 0$; $x_j \ge 0$, $v_j \ge 0 \;\; (j = \overline{1,n})$;

2) $\dfrac{\partial L}{\partial u_i} + y_i = 0 \quad (i = \overline{1,m})$ при условиях $u_i \cdot y_i = 0$; $u_i \geq 0$, $y_i \geq 0 \quad (i = \overline{1,m})$.

Решение (x^*, u^*) данной системы уравнений является седловой точкой функции Лагранжа $L(x, u)$.

На основе функции Лагранжа могут быть рассмотрены вопросы двойственного анализа задачи выпуклого программирования. Пусть имеем задачу (1.23)–(1.25) и ее функцию Лагранжа $L(x, u)$. Тогда задачу нахождения седловой функции Лагранжа может быть разделена на два этапа.

На первом этапе рассматривается задача минимизации функции Лагранжа $L(x, u)$ по переменным x, а именно:

$$\varphi(u) = \min_{x \in R} L(x, u),$$

которая является задачей выпуклого программирования. В общем случае, функция $\varphi(u)$ является кусочно-линейной, вогнутой и недифференцируемой.

Тогда на втором этапе получаем задачу недифференцируемой оптимизации

$$\max_{u \geq 0} \varphi(u),$$

которая называется двойственной задачей выпуклого программирования.

Для данных задач имеют место основные утверждения теории двойственного анализа задачи выпуклого программирования.

Если x – любое решение задачи выпуклого программирования (1.23)–(1.25), то найдутся такие $u \geq 0$, для которых имеет место неравенство

$$\varphi(u) \leq f_0(x),$$

которое означает, что $\varphi(u)$ является оценкой снизу для функционала $f_0(x)$ при любом допустимом решением задачи (1.23)–(1.25).

Если x^* – оптимальное решение задачи выпуклого программирования (1.23)–(1.25), то найдутся такие значения $u^* \geq 0$, которые являются оптимальным решением задачи $\max\limits_{u \geq 0} \varphi(x)$ и для которых имеет место равенство

$$f_0(x^*) = \varphi(u^*),$$

т.е. $\varphi(u)$ является точной оценкой функционала $f_0(x)$. Пара (x^*, u^*) является седловой точкой функции Лагранжа, т.е. для нее выполняются равенства

$$L(x^*, u^*) = \max_{u \geq 0} \min_{x \in R} L(x, u) = \min_{x \in R} \max_{u \geq 0} L(x, u).$$

Таким образом, путем решения пары двойственных задач выпуклого программирования находим седловую точку функции Лагранжа $L(x, u)$.

Рассмотрим теперь условия оптимальности для задач дробно-выпуклого программирования [478, 493, 494]. Пусть заданы непрерывно дифференцируемые выпуклые функции $\varphi(x)$, $-\psi(x)$ и $f_i(x)$ $(i = \overline{1, m})$, определенные для любого $x \in E^n$. Тогда в задаче дробно-выпуклого программирования требуется: найти

$$\min \left\{ f_0(x) = \frac{\varphi(x)}{\psi(x)} \right\} \tag{1.26}$$

при ограничениях

$$f_i(x) \leq 0,\ i = \overline{1, m}; \tag{1.27}$$

$$x \in X \subset E^n. \tag{1.28}$$

Предположим, что $\varphi(x) \geq 0$ и $\psi(x) > 0$ для всех $x \in X$. Тогда функция $f_0(x)$ является квазивыпуклой, и в целом, задача (1.26)–(1.28) является задачей квазивыпуклой оптимизации, для решения которой можно использовать общие методы [30, 94, 106, 181, 330, 406, 493, 494, 589, 629, 645, 695]. В случае дробно-выпуклого функционала можно применять теорию и методы выпуклого программирования [18, 92, 141, 143, 393, 394, 396, 409, 417, 418, 521, 600, 660].

В общем случае функция Лагранжа для задачи (1.26)–(1.28) определяется по формуле

$$L_1(x, u) = \frac{\varphi(x)}{\psi(x)} + \sum_{i=1}^{m} u_i f_i(x), \tag{1.29}$$

где $u = \{u_i\}_{i=1}^m$ – множители Лагранжа для ограничений (1.27). Следует отметить, что функционал $L_1(x, u)$ не является однозначно выпуклым или квазивыпуклым по переменным x. Для анализа ее связи с задачей (1.26)–(1.28) используем общую теорию нелинейного программирования. Тогда по теореме 1.14 для этой функции существуют множители $u^* = \{u_1^*, u_2^*, \ldots, u_m^*\}$ такие, что в точке x^* имеют место равенства

$$\begin{aligned} \frac{\partial L_1(x^*, u^*)}{\partial x_j} &= \psi(x^*) \frac{\partial \varphi(x^*)}{\partial x_j} - \varphi(x^*) \frac{\partial \psi(x^*)}{\partial x_j} + \\ &+ \psi^2(x^*) \sum_{i=1}^{m} u_i^* \frac{\partial f_i(x^*)}{\partial x_j} = 0,\ j = \overline{1, n}. \end{aligned} \tag{1.30}$$

Для задач дробно-выпуклого программирования (1.26)–(1.28) имеет место теорема Куна-Таккера. Пусть $\varphi(x)$, $-\psi(x)$ и $f_i(x)$ – выпуклые функции, определенные на E^n. Предположим, что $\varphi(x) \geq 0$ и $\psi(x) > 0$ для всех $x \in X$.

Теорема 1.21. *Пусть для задачи* (1.26)–(1.28) *выполняются условия Слейтера. Тогда для того чтобы x^* было оптимальным решением задачи* (1.26)–(1.28), *необходимо и достаточно существование такого $u^* \geq 0$, что (x^*, u^*) является седловой точкой функции Лагранжа на множестве $X \times \{u: \; u \geq 0\}$, т.е.*

$$L_1(x^*, u) \leq L_1(x^*, u^*) \leq L_1(x, u^*)$$

при $x \in X$, $u \geq 0$. При этом $u_i^* f_i(x^*) = 0 \quad (i = \overline{1, m})$.

Для нахождения седловой точке функции Лагранжа составляем и решаем выше приведенную систему уравнений 1) и 2) относительно функции $L_1(x, u)$, которые в данном случае имеют вид:

1) $$\frac{\partial L_1}{\partial x_j} = \psi(x)\frac{\partial \varphi(x)}{\partial x_j} - \varphi(x)\frac{\partial \psi(x)}{\partial x_j} + \psi^2(x)\sum_{i=1}^{m} u_i \frac{\partial f_i(x)}{\partial x_j} - v_j = 0;$$
$x_j v_j = 0; \quad x_j \geq 0, \, v_j \geq 0 \quad (j = \overline{1, n})$;

2) $\dfrac{\partial L_1}{\partial u_i} = f_i(x) + y_i = 0; \quad u_i y_i = 0; \quad u_i \geq 0, \; y_i \geq 0 \quad (i = \overline{1, m})$.

Функцию Лагранжа задачи (1.26)–(1.28) также можно построить по следующей формуле [595, 596, 598].

$$L_2(x, u) = \frac{\varphi(x) + \sum\limits_{i=1}^{m} u_i f_i(x)}{\psi(x)},$$

которая в данном случае является квазивыпуклой функцией.

Для функции $L_2(x, u)$ необходимые условия локального минимума типа (1.30) в точке x^* имеют вид

$$\begin{aligned} \frac{\partial L_2(x^*, u^*)}{\partial x_j} \quad = \quad & \frac{\partial \varphi(x^*)}{\partial x_j} - \frac{\partial \psi(x^*)}{\partial x_j} \frac{\varphi(x^*) + \sum\limits_{i=1}^{m} u_i^* f_i(x^*)}{\psi(x)} + \\ & + \sum_{i=1}^{m} u_i^* \frac{\partial f_i(x^*)}{\partial x_j} = 0. \end{aligned} \tag{1.31}$$

Тогда в соответствии с теоремой 1.20 для нахождения седловой точке Лагранжа для функции $L_2(x,u)$ системы уравнений типа 1) и 2) имеют вид:

$$1)\qquad \frac{\partial L_2}{\partial x_j} = \psi(x)\frac{\partial \varphi(x)}{\partial x_j} - \varphi(x)\frac{\partial \psi(x)}{\partial x_j} + \sum_{i=1}^{m} u_i \frac{\partial f_i(x)}{\partial x_j} - v_j = 0;$$

$x_j v_j = 0;\ \ x_j \geq 0,\ v_j \geq 0\ \ (j=\overline{1,n});$

$$2)\quad \frac{\partial L_2}{\partial u_i} = f_i(x) + y_i = 0;\quad u_i y_i = 0;\quad u_i \geq 0,\ y_i \geq 0\quad (i=\overline{1,m}).$$

Если $\varphi(x) \geq 0$ и $\psi(x) > 0$ для любых $x \in X$, то функции Лагранжа $L_1(x,u)$ и $L_2(x,u)$ задачи (1.26)–(1.28) эквивалентны. Это вытекает из следующих теорем.

Теорема 1.22. *Пусть $\varphi(x) \geq 0$ и $\psi(x) > 0$ для любых $x \in X$. Если x^* точка локального минимума задачи* (1.26)–(1.28), *а $u_1^*, u_2^*, \ldots, u_m^*$ – множители Лагранжа функции $L_1(x,u)$, для которых справедливы* (1.30), *то найдутся такие множители Лагранжа $\bar{u}_i^* = u_i^*\,\psi(x^*)$ $(i = \overline{1,m})$, для которых имеет место* (1.31) *для функции $L_2(x,u)$, и наоборот, если $\{\bar{u}_i^*\}_{i=1}^m$ – множители Лагранжа функции $L_2(x,u)$, для которых в точке локального минимума x^* задачи* (1.26)–(1.28) *выполняются* (1.31), *то найдутся такие множители Лагранжа $u_i^* = \bar{u}_i^*/\psi(x^*)$ $(i=\overline{1,m})$, для которых имеет место* (1.30) *для функции $L_1(x,u)$.*

Теорема 1.23. *Пусть $\varphi(x) \geq 0$ и $\psi(x) > 0$ для любых $x \in X$. Если пара (x^*, u^*) является седловой точкой функции Лагранжа $L_1(x,u)$ задачи* (1.26)–(1.28) *на множестве $X \times \{u : u \geq 0\}$, то пара $(x^*, \bar{u}^*)$, где $\bar{u}_i^* = u_i^*\psi(x^*)$ $(i=\overline{1,m})$, является седловой точкой функции $L_2(x,u)$ на множестве $X \times \{\overline{u} : \overline{u} \geq 0\}$, и наоборот.*

Для задачи дробно-выпуклого программирования на основе функции Лагранжа можно рассматривать аналогичные двойственные аспекты. Пусть имеем функции Лагранжа $L_1(x,u)$ и $L_2(x,u)$ задачи дробно-выпуклого программирования (1.26)–(1.28).

Тогда для нахождения седловой точке функций Лагранжа $L_1(x,u)$ или $L_2(x,u)$ также можно использовать двухэтапную схему. На первом этапе рассматривается одна из задач

$$L_1^*(u) = \min_{x\in X} L_1(x,u) \quad \text{или} \quad L_2^*(u) = \min_{x\in X} L_2(x,u).$$

В отличие от функций $L_1(x,u)$ и $L_2(x,u)$, которые имеют различные аспекты выпуклости, функции $L_1^*(u)$ и $L_2^*(u)$ являются кусочно-линейными, вогнутыми и недифференцируемыми. Поэтому, на втором этапе получаем соответствующие задачи недифференцируемой оптимизации

$$\max_{u\geq 0} L_1^*(u) \quad \text{или} \quad \max_{u\geq 0} L_2^*(u),$$

которые называются двойственные задачи дробно-выпуклого программирования.

Следует отметить, что для задачи дробно-выпуклого программирования можно построить и другие двойственные задачи [8, 47, 63, 108, 110, 117, 195, 233, 270, 327, 384, 434, 435, 450, 453, 507, 526, 595, 596].

Для двойственных задач дробно-выпуклого программирования имеют место основные утверждения теории двойственного анализа, а именно:

Если x – любое решение задачи дробно-выпуклого программирования (1.26)–(1.28), то найдутся такие множители Лагранжа $u \geq 0$ или $\overline{u} \geq 0$, для которых имеют место соответствующие неравенства

$$L_1^*(u) \leq \frac{\varphi(x)}{\psi(x)} \quad \text{или} \quad L_2^*(\overline{u}) \leq \frac{\varphi(x)}{\psi(x)}.$$

Эти неравенства представляют собой оценки снизу дробно-выпуклого функционала $f_0(x)$ при любых $x \in X$.

Если же x^* является оптимальное решение задачи дробно-выпуклого программирования (1.26)–(1.28), то найдется такое значение $u^* \geq 0$, которое является оптимальное решение задачи $\max\limits_{u\geq 0} L_1^*(u)$ и значение $\overline{u}^* \geq 0$, которое является оптимальное решение задачи $\max\limits_{u\geq 0} L_2^*(\overline{u})$, для которых имеют место равенства

$$L_1^*(u^*) = L_2^*(\overline{u}^*) = f_0(x^*) = \frac{\varphi(x^*)}{\psi(x^*)},$$

т.е. $L_1^*(u^*) = L_2^*(\overline{u}^*)$ является точной оценкой дробного функционала $f_0(x)$. Таким образом пара (x^*, u^*) является седловой точкой функции Лагранжа $L_1(x,u)$, а пара $(x^*, \overline{u}^*)$ – седловая точка функции Лагранжа $L_2(x,u)$, т.е. для них выполняются равенства

$$L_1(x^*, u^*) = \max_{u\geq 0} \min_{x\in X} L_1(x,u) = \min_{x\in X} \max_{u\geq 0} L_1(x,u),$$

или

$$L_2(x^*, \overline{u}^*) = \max_{u\geq 0} \min_{x\in X} L_2(x,u) = \min_{x\in X} \max_{u\geq 0} L_2(x,u).$$

Таким образом, решая одну задачу из пар двойственных задач дробно-выпуклого программирования находим седловую точку (x^*, u^*) функции $L_1(x, u)$, или $(x^*, \overline{u}^*)$ – функции $L_2(x, u)$.

Замечание. *При решении задачи выпуклого программирования прямыми методами, а не методом нахождения седловой точки, в некоторых случаях нам необходимо кроме оптимальных значений переменных x найти и оптимальные значения множителей Лагранжа u. Для их нахождения можно решить следующую систему однородных уравнений:*

$$h_j(u) = \left(x_j^*, \frac{\partial L(x^*, u)}{\partial x_j}\right) = 0$$

для всех $j = \overline{1, n}$*, для которых* $x_j^* \neq 0$*. Найденные решения* u_i^* *согласно теореме Куна-Таккера и будут оптимальными значениями множителей Лагранжа.*

Данное замечание относится и к задаче дробно-выпуклого программирования.

1.8. Задачи квадратичного программирования и недифференцируемая оптимизация

Частным случаем задачи математического программирования (1.19)–1.21) является задача квадратичного программирования, в которой ограничения $f_r(x_1, x_2, \ldots, x_n) \leq 0$ $(r = \overline{1, m})$ являются линейными $\sum\limits_{j=1}^{n} a_{rj}x_j - b_r \leq 0$ $\left(\sum\limits_{j=1}^{n} a_{rj}x_j \leq b_r\right)$, а целевая функция z представляет собой сумму линейной и квадратичной функции

$$z = \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_i x_j \quad (d_{ij} = d_{ji}).$$

Квадратичная функция n переменных называется квадратичной формой и может быть записана в матрично-векторном виде:

$$D(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_i x_j = (Dx,\, x),$$

где $x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$, а $D = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \dots & \dots & \dots & \dots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{pmatrix}$ – квадратная симметричная матрица ($d_{ij} = d_{ji}$).

Если предположить, что матрица D положительно определенная, т.е. $(p, Dp) \geq 0$ при всех векторов $p \in R^n$, то квадратичная форма $D(x)$ представляет собой выпуклую функцию и целевая функция z является выпуклой функцией, так как она является суммой линейной функции (которая и выпукла и вогнута) и положительно определенной квадратичной формы. В этом случае получаем частный случай задачи квадратичного программирования в виде задачи выпуклого программирования. Благодаря специфики задачи квадратичного программирования для ее решения разработаны различные методы [198, 200, 403, 405, 411, 547, 559, 574, 734].

Рассмотрим особенности применения функции Лагранжа и необходимых условий оптимальности на примере задачи выпуклого квадратичного программирования, которая сводится к решению системы линейных уравнений, т.е. к задаче линейного программирования.

Для заданной задачи выпуклого квадратичного программирования функция Лагранжа имеет вид

$$L(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m) =$$

$$= \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_i x_j + \sum_{r=1}^{m} u_r \Big(\sum_{j=1}^{n} a_{rj} x_j - b_r \Big).$$

Условия Куна-Таккера имеют вид:

$$\frac{\partial L}{\partial x_j} = c_j + \sum_{i=1}^{n} d_{ij} x_i + \sum_{r=1}^{m} a_{rj} u_r \geq 0 \qquad (j = \overline{1, n}),$$

$$x_j \cdot \frac{\partial L}{\partial x_j} = x_j \Big(c_j + \sum_{i=1}^{n} d_{ij} x_i + \sum_{r=1}^{m} a_{rj} u_r \Big) = 0 \qquad (j = \overline{1, n}),$$

$$\frac{\partial L}{\partial u_r} = \sum_{j=1}^{n} a_{rj} x_j - b_r \le 0 \qquad (r = \overline{1,m}),$$

$$u_r \cdot \frac{\partial L}{\partial u_r} = u_r \Big(\sum_{j=1}^{n} a_{rj} x_j - b_r \Big) = 0 \qquad (r = \overline{1,m}).$$

Введем дополнительные переменные $v_j \ge 0$ $(j = \overline{1,n})$ так, чтобы

$$c_j + \sum_{i=1}^{n} d_{ij} x_i + \sum_{r=1}^{m} a_{rj} u_r - v_j = 0 \quad (j = \overline{1,n}),$$

и $y_r \ge 0$ $(r = \overline{1,m})$ так, чтобы

$$\sum_{j=1}^{n} a_{rj} x_j - b_r + y_r = 0 \quad (r = \overline{1,m}).$$

Тогда условия Куна-Таккера можно записать в форме

$$\frac{\partial L}{\partial x_j} = c_j + \sum_{i=1}^{n} d_{ij} x_i + \sum_{r=1}^{m} a_{rj} u_r - v_j = 0 \qquad (j = \overline{1,n}),$$

$$x_j \cdot \frac{\partial L}{\partial x_j} = x_j \cdot v_j = 0 \qquad (j = \overline{1,n}),$$

$$x_j \ge 0, \quad v_j \ge 0 \qquad (j = \overline{1,n}),$$

$$\frac{\partial L}{\partial u_r} = \sum_{j=1}^{n} a_{rj} x_j - b_r + y_r = 0 \qquad (r = \overline{1,m}),$$

$$u_r \cdot \frac{\partial L}{\partial u_r} = u_r \cdot y_r = 0 \qquad (r = \overline{1,m}),$$

$$u_r \ge 0, \quad y_r \ge 0 \qquad (r = \overline{1,m}).$$

Таким образом, чтобы найти решение задачи выпуклого квадратичного программирования, нужно определить неотрицательное решение системы линейных уравнений

$$\begin{cases} \sum_{i=1}^{n} d_{ij}x_i + \sum_{r=1}^{m} a_{rj}u_r - v_j = -c_j \qquad (j = \overline{1,n}), \\ \qquad \sum_{j=1}^{n} a_{rj}x_j + y_r = b_r \qquad (r = \overline{1,m}), \end{cases}$$

которое удовлетворяет условиям $x_j \cdot v_j = 0 \quad (j = \overline{1,n})$ и $u_r \cdot y_r = 0 \ \ (r = \overline{1,m})$. Решение данной системы равносильно нахождения базисного решения следующей задаче линейного программирования методом искусственных переменных. Для этого вводим переменные $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ и $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ и формируем псевдофункцию

$$Z(\alpha, \beta) = \sum_{j=1}^{n} M\alpha_j + \sum_{r=1}^{m} M\beta_r,$$

для которой находим минимальное значение при следующих ограничениях

$$\sum_{i=1}^{n} d_{ij}x_i + \sum_{r=1}^{m} a_{rj}u_r - v_j + \alpha_j = -c_j \quad (j = \overline{1,n}),$$

$$\sum_{j=1}^{n} a_{rj}x_j + y_r + \beta_r = b_r \quad (r = \overline{1,m}),$$

$$x_j \geq 0, \quad v_j \geq 0, \quad \alpha_j \geq 0 \quad (j = \overline{1,n}),$$

$$u_r \geq 0, \quad y_r \geq 0, \quad \beta_r \geq 0 \quad (r = \overline{1,m}).$$

Тогда процесс нахождения решения задачи выпуклого квадратичного программирования с использованием условий Куна-Таккера включает следующие этапы:

- составление функции Лагранжа;

- запись необходимых и достаточных условий существования седловой точки для функции Лагранжа;

- решение задачи линейного программирования и нахождение седловой точки функции Лагранжа, если она существует, или установление отсутствия такой точки;

- нахождение оптимального решения и оптимального значения целевой функции.

Рассмотрим двойственный анализ задачи квадратичного программирования [468, 470, 712, 714, 734]. Пусть $L(x, u)$ является функцией Лагранжа задачи квадратичного программирования. Тогда в соответствие с теории двойственности по Лагранжу рассмотрим задачу

$$\varphi(u) = \min_{x \in R^n} L(x, u) = \min_{x \in R^n} \left[(c, x) + (Dx, x) + +(u,\ Ax - b)\right].$$

Допустим, что матрица D невырождена. Тогда приравнивая производные от $L(x, u)$ по x нулю, получаем

$$\frac{\partial L(x, u)}{\partial x_j} = c_j + \sum_{i=1}^{n} d_{ij} x_j + \sum_{r=1}^{m} a_{rj} u_r = 0 \quad (j = \overline{1, n}),$$

или же в матричном виде

$$Dx + c + A^T u = 0,$$

где $A = \{a_{rj}\}_{m \times n}$ – матрица коэффициентов линейных ограничений, $c = (c_1, c_2, \ldots, c_n)$, $u = (u_1, u_2, \ldots, u_m)$, соответствующие вектора. Тогда получаем решение

$$x = -D^{-1}[c + A^T u].$$

Подставляя это выражение в функцию $L(x, u)$ имеем

$$\varphi(u) = -\big(D^{-1}[c + A^T u],\, c + A^T u\big) - (b, u).$$

Тогда двойственная задача квадратичного программирования имеет вид

$$\max_{u \geq 0} \varphi(u),$$

или же задача максимизации функции

$$\varphi(u) = -\big(D^{-1}[c + A^T u],\, c + A^T u\big) - (b, u)$$

при ограничениях

$$u_r \geq 0 \quad (r = \overline{1, m}).$$

Имеет место утверждение [657]:

Пусть D – положительно определенная невырожденная матрица. Тогда задача максимизации функции $\varphi(u)$ при ограничениях $u \geq 0$ является двойственной задачей квадратичного программирования. Если u^* – ее решение, то $x^* = -D^{-1}[c + A^T u^*]$ является решением исходной задачи квадратичного программирования.

Для общей задачи квадратичного программирования вида: определить

$$K^* = \inf K_0(x), \quad x \in M \subseteq E^n$$

при ограничениях

$$K_r(x) \le 0 \qquad (r = \overline{1, m_1}),$$

$$K_r(x) = 0 \qquad (r = \overline{m_1 + 1, m_1 + m_2}),$$

где $K_\tau(x) = (A_\tau x,\, x) + (l_\tau,\, x) + c_\tau \;\; (\tau = \overline{0, m})$, где $m = m_1 + m_2$ – квадратичные функции, определенные на n-мерном евклидовом пространстве E^n, A_τ – симметричные матрицы размерности $n \times n$, l_τ – n-мерные векторы, c_τ – константы $(\tau = \overline{0, m})$, M – либо совпадает с E^n, либо является выпуклым замкнутым подмножеством в E^n.

Квадратичные задачи являются обобщением задачи выпуклого квадратичного программирования, для решения которых применяются различные методы [198, 411, 501, 559, 632, 712, 714, 734]. Среди задач квадратичного программирования можно выделить следующие частные случаи:

а) выпуклые квадратичные задачи, в которых при $m = m_1$ матрицы $A_\tau \;\; (\tau = \overline{0, m_1})$ являются положительно определенными;

б) квадратичные задачи с линейными ограничениями, в которых $A_r = 0 \;\; (r = \overline{1, m})$. Если A_0 – неотрицательно-определенная матрица, то получаем выпуклую задачу квадратичного программирования с линейными ограничениями;

в) задачи квадратичной оптимизации, когда A_0 – отрицательно-определенная матрица, $A_r = 0 \;\; (r = \overline{1, m})$ или A_r – положительно-определенные матрицы является многоэкстремальной и относится к классу NP-трудных задач (к этому классу относится задача вогнутого квадратичного программирования);

г) квадратичные оптимизационные задачи с булевыми переменными, когда на компоненты вектора x накладываются условия $x_j \in \{0,\, 1\} \;\; (j = \overline{1, n})$, которое эквивалентна квадратичным равенствам $x_j^2 - x_j = 0 \;\; (j = \overline{1, n})$.

Многоэкстремальные квадратичные задачи в общей постановке рассмотрены в различных работах [198, 411, 501, 559, 574, 632]. Так как в общем случае, для решения таких задач не избежать перебора, то предлагается применить схему метода "ветвей и границ" при реализации которой необходимо иметь эффективный и достаточно общий метод получения оценок снизу для целевого функционала [467, 470, 712, 714, 734].

Таким общим методом является двойственный подход для получения нижних оценок по функционалу, основанный на использование функции Лагранжа и методов недифференцируемой оптимизации [467, 471, 727].

Рассмотрим общую схему такого метода. Для этого построим функцию Лагранжа

$$L(x,u) = K_0(x) + \sum_{r=1}^{m} u_r K_r(x)$$

и рассмотрим задачу

$$\psi(u) = \inf_{x \in M} L(x,u).$$

Тогда, если X – множество допустимых решений исходной квадратичной задачи, т.е.

$$X = \{x: \ K_r(x) \le 0 \ \ (r = \overline{1,m_1}); \ K_r(x) = 0 \ \ (r = \overline{m_1+1,m}); \ x \in M\}$$

и имеем точку $(\overline{x},\overline{u})$, для которой $\overline{x} \in X$, а $\overline{u}_r \ge 0 \ \ (r = \overline{1,m_1})$, то $\overline{u}_r \cdot K_r(\overline{x}) \le 0 \ \ (r = \overline{1,m_1})$ и $\overline{u}_r \cdot K_r(\overline{x}) = 0 \ \ (r = \overline{m_1+1,m})$. Тогда $L(\overline{x},\overline{u}) \le K_0(\overline{x})$ и

$$\psi(\overline{u}) = \inf_{x \in M} L(x,\overline{u}) \le \inf_{x \in X} L(x,\overline{u}) \le \inf_{x \in X} K_0(x) = K^*.$$

Таким образом $\psi(u)$ является оценкой снизу для K^* при $u_i \ge 0 \ \ (i = \overline{1,m})$. Для получения наиболее точной оценкой такого типа нужно решить задачу нахождения

$$\psi^* = \sup_{u \ge 0} \psi(u),$$

которая является двойственной для задачи квадратичного программирования.

Так как функция $\psi(u)$ является кусочно-линейной, вогнутой и недифференцируемой функцией, то для решения задачи нахождения ψ^* применяются методы недифференцируемой оптимизации.

В таких методах при фиксированных значениях параметров $u = \overline{u}$ необходимо решать задачу квадратичной оптимизации $\psi(u) = \inf_{x \in M} L(x,u)$, в которой функция Лагранжа может быть записана в следующем общем виде

$$L(x,u) = (A(u)x, \, x) + (l(u), \, x) + c(u),$$

в которой
$$A(u) = A_0 + \sum_{r=1}^{m} u_r A_r, \quad l(u) = l_0 + \sum_{r=1}^{m} u_r l_r, \quad c(u) = c_0 + \sum_{r=1}^{m} u_r c_r.$$

Таким образом, элементы матрицы $A(u) = \{a_{ij}(u)\}_{i,j=1}^{n}$ линейно зависят от u и тогда, в случае когда матрица $A(u)$ является неотрицательно-определенной и невырожденной, то задача нахождения $\psi(u)$ является задачей выпуклого квадратичного программирования и она может быть решена одним из эффективным конечношаговым алгоритмом.

Тогда для нахождения $\psi(u)$ нужно решить систему линейных уравнений
$$2A(u)x(u) + l(u) = 0,$$
откуда
$$x(u) = -\frac{1}{2}A^{-1}(u)l(u).$$
Тогда получим
$$\psi(u) = -\frac{1}{4}\left(A^{-1}(u)l(u),\ l(u)\right) + c(u).$$

Так как функция $\psi(u)$ в данном случае является выпуклой, непрерывно дифференцируемой, то ее градиент определяется по формуле
$$g_{\psi}(u) = \{K_r(x(u))\}_{r=1}^{m},$$
а задача
$$\max_{u \geq 0} \psi(u)$$
является задачей выпуклого программирования, для решения которой можно использовать общеизвестные методы, среди которых, как будет отмечено далее, относится метод эллипсоидов.

1.9. Методы штрафных и барьерных функций и недифференцируемая оптимизация

Эффективными приемами используемые при решении задач оптимизации является применение штрафных или барьерных функций, на основе которых исходная задача оптимизации при заданных ограничениях преобразуется в одну или в последовательности задач безусловной оптимизации [584, 632].

С помощью специальных функций, построенных на заданных ограничениях задачи, строятся так называемые штрафные или барьерные функции, которые добавляются к целевой функции исходной задачи так, что нарушение какого-либо ограничения становится невыгодным при решении полученной задачи безусловной оптимизации. В зависимости от типа используемых штрафных или барьерных функций можем получить задачу безусловной оптимизации дифференцируемой или недифференцируемой функции. Таким образом, для сведения задач условной оптимизации к безусловной можно использовать гладкие или негладкие штрафные функции. Также следует отметить, что к задачам недифференцируемой оптимизации можем прийти, если даже использовать только гладкие функции.

В дальнейшем рассмотрим методы штрафных и барьерных функций и их возможная связь с задачами недифференцируемой оптимизации.

1. Метод штрафных функций. В методе штрафных функций решения задач нелинейного программирования основной трудностью является выбор величины параметра штрафа, в результате которой вместо одной безусловной задаче, необходимо решать целую серию вспомогательных задач, каждый раз увеличивая размер штрафа. При этом целевые функции вспомогательных задач выбираются таким образом, чтобы решения x_k таких задач сходились при $k \to \infty$ к решению x^* исходной задачи. Для решения вспомогательных задач применяются методы решения задач без ограничений. В таких методах эффективность их применения достигается за счет искусства выбора значений последовательности коэффициентов штрафа в штрафных функциях и в меньшей степени зависит от выбора самой штрафной функции.

Следует отметить, что в некоторых случаях выгодно часть ограничений, определяющие допустимое множество исходной задачи, сохранить в процессе образования вспомогательных задач, а часть из них свертывать в штрафной функции. Это могут быть граничные значения для части переменных x или некоторые ограничения специальной структуры (линейные, транспортные, блочные и т.п.).

Для общей задаче нелинейного программирования

$$\min z = f(x) \tag{1.32}$$

при ограничениях

$$g_i(x) \le 0 \quad (i = \overline{1,m}), \tag{1.33}$$

$$h_k(x) = 0 \quad (k = \overline{1,l}), \tag{1.34}$$

типичная штрафная функция может быть рассмотрена в общем виде

$$\alpha(x) = \sum_{i=1}^{m} \varphi(g_i(x)) + \sum_{k=1}^{l} \psi(h_k(x)),$$

где $\varphi(t)$ и $\psi(t)$ – непрерывные функции одной переменной, удовлетворяющие условием

$$\varphi(t) = 0, \quad \text{если } t \leq 0 \quad \text{и} \quad \varphi(t) > 0, \quad \text{если } t > 0;$$

$$\psi(t) = 0, \quad \text{если } t = 0 \quad \text{и} \quad \psi(t) > 0, \quad \text{если } t \neq 0.$$

Также могут применяться более сложные по своей конструкции штрафные функции, когда в зависимости от структуры функций $g_i(x)$ и $h_k(x)$, для каждого из них определяются свои функции штрафа $\varphi_i(g_i(x))$ и $\psi_k(h_k(x))$. Тогда имеем общую штрафную функцию

$$\alpha(x) = \sum_{i=1}^{m} \varphi_i(g_i(x)) + \sum_{k=1}^{l} \psi_k(h_k(x)),$$

в которой $\varphi_i(t)$ и $\psi_k(t)$ – непрерывные неотрицательные функции от переменной t, которые удовлетворяют тем же условиям, как и функции $\varphi(t)$ и $\psi(t)$. Типичными такими функциями φ и ψ являются функции вида $\varphi(t) = (\max\{0;\, t\})^p$ и $\psi(t) = |t|^p$, где p – целое положительное число.

В этом случае штрафная функция имеет вид

$$\alpha(x) = \sum_{i=1}^{m} (\max\{0;\, g_i(x)\})^p + \sum_{k=1}^{l} |h_k(x)|^p.$$

Таким образом, для общей задаче нелинейного программирования, в результате применения метода штрафной функции, задача безусловной оптимизации имеет вид

$$\min \Phi(x, \mu) = f(x) + \mu\, \alpha(x)$$

$$\text{при} \quad x \in E^n,$$

где $\alpha(x)$ – штрафная функция.

Задача минимизации функции $\Phi(x, \mu)$ является задачей безусловной оптимизации, для решения которой можно использовать различные традиционные алгоритмы. Сложности применения таких алгоритмов проявляются при приближением к оптимальному значению целевой функции исходной задачи, когда вычисляются значения функции $\Phi(x, \mu)$ при достаточно больших значениях штрафного параметра μ.

Приведем общую схему алгоритма метода штрафных функций для решения задачи (1.32)–(1.34). Используется штрафная функция вида

$$\alpha(x) = \sum_{i=1}^{m} \Big(\max\{0, g_i(x)\} \Big)^2 + \sum_{k=1}^{l} h_k^2(x).$$

Этот метод не накладывает каких-либо ограничений на функции $f(x)$, $g_i(x)$ и $h_k(x)$, помимо их непрерывности.

Начальный этап. Выбрать $\varepsilon > 0$ (параметр критерия остановки процесса вычислений). Выбрать начальную точку x_1, величину штрафного параметра μ_1 и число $\beta > 1$. Положить $r = 1$ и перейти к основному этапу.

Основной этап. Шаг 1. При начальной точке x_r решить задачу безусловной оптимизации

$$\min \Phi(x, \mu_r) = f(x) + \mu_r \alpha(x).$$

Пусть x_{r+1} – оптимальное решение этой задачи. Перейдем к шагу 2.

Шаг 2. Если $\mu_r \alpha(x_{r+1}) < \varepsilon$, то остановиться; в противном случае положить $\mu_{r+1} = \beta \mu_r$. Заменить r на $r + 1$ и перейти к шагу 1.

В результате применения итерационного алгоритма метода штрафных функций вырабатывается последовательность штрафов $\mu_1, \ldots$ $\ldots, \mu_r, \ldots$, таких что $\mu_{r+1} > \mu_r > 0$, $r = 1, 2, \ldots$ и $\lim\limits_{r\to\infty} \mu_r = \infty$, при которых последовательность решений $x_1, \ldots, x_r, \ldots$, удовлетворяет соотношениям:

1) $\Phi(x_r, \mu_r) \le \Phi(x_{r+1}, \mu_{r+1})$, $\quad r = 1, 2, \ldots;$

2) $f(x_r) \le f(x_{r+1})$, $\quad r = 1, 2, \ldots;$

3) $f(x_r) \le \Phi(x_r, \mu_r) \le f(x^*)$, $\quad r = 1, 2, \ldots;$

4) $\lim\limits_{r\to\infty} x_r = x^*$, $\lim\limits_{r\to\infty} \Phi(x_r, \mu_r) = \lim\limits_{r\to\infty} f(x_r) = f(x^*)$.

Таким образом, оптимальное значение $f(x^*)$ исходной задачи (1.32)–(1.34) является верхней оценкой для неубывающих последовательностей $\{f(x_r)\}$ и $\{\Phi(x_r, \mu_r)\}$.

Если в задаче нелинейного программирования (1.32)–(1.34) имеются только ограничения типа неравенств (1.33), то в качестве штрафных функций можно взять, например, следующие две типичные функции:

1) $\alpha(x) = \max\limits_{i=\overline{1,m}} \{0, g_i(x)\}$,

2) $\alpha(x) = \sum\limits_{i=1}^{m} \alpha\left(g_i(x)\right)$,

где $\alpha(t) = 0$, если $t \le 0$ и $\alpha(t) > 0$ если $t > 0$.

Таким образом, метод решения задач условной оптимизации с применением штрафных функций обеспечивает приближение безусловных локальных минимумов последовательности задач $\{\min \Phi(x_r, \mu_r)\}$ к допустимой области путем движения со стороны внешней части допустимой области. Поэтому метод штрафных функций называется методом внешней точки, который обеспечивает приближение к оптимальному решению со стороны недопустимых точек, т.е. внешних точек множества $K = \{x\colon g_i(x) \leq 0 \ \ (i = \overline{1, m})\}$. Метод штрафных функций дает простую и универсальную схему решения задач условной оптимизации. Так как на практике имеется достаточно широкий выбор штрафных функций, то при составлении функции $\alpha(x)$ следует обеспечить нужную гладкость этой функции, ее выпуклость, а также простоту вычисления ее значений и требуемых ее производных и т.п. Однако не всегда функция $\alpha(x)$ будет гладкой, а также она может стать более овражной, чем исходная, функцией, что не обеспечивает достаточную эффективность метода штрафных функций.

Через призму методов штрафных функций могут быть рассмотрены и двойственные аспекты задачи нелинейного программирования. Рассмотрим прямую задачу нелинейного программирования:

$$\min\{f(x) : \ x \in K)\},$$

$$K = \{x \in R^n : \ g_i(x) \leq 0 \ \ (i = \overline{1, m})\}.$$

Тогда по методу штрафных функций вместо рассматриваемой прямой задачи нелинейного программирования решаются последовательность задач безусловной оптимизации

$$\min\{f(x) + \mu_k \alpha(x) : \ x \in R^n\},$$

где $\alpha(x) = \sum\limits_{i=1}^{m} g_i^{+^2}(x)$; $g_i^+(x) = \max\{0, g_i(x)\}$.

Метод штрафов может рассматриваться как "двойственный метод" для решения прямой задачи нелинейного программирования. Если прямую задачу записать в эквивалентном виде

$$\min\{f(x) : \ \alpha(x) \leq 0, \, x \in R^n\},$$

то двойственной к данной задачей является задачей безусловной оптимизации

$$\max\{\varphi(\mu) : \ \mu \geq 0\},$$

где $\varphi(\mu) = \inf\limits_{x \in R^n} \{f(x) + \mu\alpha(x)\}$.

Таким образом, используя "двойственный метод" штрафов, исходная задача условной оптимизации сводится к решению двух (внешней и внутренней) задач безусловной оптимизации.

Внешняя задача – одномерная задача максимизации недифференцируемой вогнутой функции $\varphi(\mu)$, т.е. имеем задачу

$$\max_{\mu \geq 0} \varphi(\mu).$$

Внутренняя задача – безусловная задача минимизации по x недифференцируемой функции $f(x)+\mu\,\alpha(x)$ при фиксированном μ, т.е. имеем задачу

$$\inf_{x \in R^n} \{f(x) + \mu\,\alpha(x)\}.$$

В случае когда функции $f(x)$ и $g_i(x)$ $(i = \overline{1,m})$ – выпуклы, то имеем задачу выпуклой безусловной оптимизации. Если для решения задачи $\max\limits_{\mu \geq 0} \varphi(\mu)$ применять один из методов недифференцируемой оптимизации, то значения параметра штрафа μ выбираются в соответствие с теоретическими аспектами используемого субградиентного метода.

Если функция $\alpha(x)$ является негладкой, то внутренняя задача также является недифференцируемой, и поэтому, для ее решения могут быть использованы субградиентные методы безусловной оптимизации выпуклой функции.

2. Метод барьерных функций. Кроме метода внешних штрафных функций, для решения задач нелинейного программирования с ограничениями-неравенствами используют метод барьеров, суть которого заключается в преобразовании задачи с ограничениями в задачу безусловной оптимизации с помощью введения специальных барьерных функций, которые как бы препятствуют выходу рассматриваемой точки из допустимой области. Рассмотрим, как исходная задача преобразуется в задачу безусловной оптимизации с помощью введения барьерной функции.

Пусть дана задача

$$\min z = f(x),$$

при

$$g_i(x) \leq 0 \quad (i = \overline{1,m}),$$

где функции f и $g_1, g_2, \ldots, g_m$ – непрерывные функции n переменных. Введем барьерную функцию $B(x)$, которая должна обладать свойствами неотрицательности и непрерывности внутри допустимой области и стремящаяся к бесконечности при приближении изнутри к границе области K. Поэтому метод барьерных функций называют методом внутренней точки, так как приближение к оптимальной точке осуществляется путем движения по внутренним точкам к границе допустимой области.

Типичным примером барьерной функции является функция

$$B(x) = \sum_{i=1}^{m} \frac{-1}{g_i(x)}.$$

Тогда исходная задача сводится к задаче безусловной оптимизации функции

$$F(x, \lambda) = f(x) + \lambda B(x),$$

где $\lambda > 0$ и значение которого убывает с каждым шагом алгоритма барьерных функций.

Следует отметить, что метод барьерных функций характерен тем свойством, что при приближении к границе допустимой области K штраф резко растет, т.е. как только некоторое $g_i(x) \to -0$, штраф $B(x)$ становится очень большим. Это имеет место из-за дробного вида барьерной функции.

Таким образом, в методе барьерных функций исходная задача сводится к решению последовательности вспомогательных задач $F(x, \lambda_k)$ таких, что

$$F(x, \lambda_k) = f(x) + \lambda_k B(x),$$

где функции $B(x)$ непрерывны на множестве внутренних "точек Слейтера" и $\lim\limits_{k\to\infty} B(x_k) = \infty$, а от последовательности $\{\lambda_k\}$, требуется, чтобы $\lambda_k > \lambda_{k+1} > 0, \quad k = 1, 2, \ldots$ и $\lim\limits_{k\to\infty} \lambda_k = 0$.

Алгоритм метода барьерных функций соответствует по своей структуре алгоритму метода штрафов. Пусть в результате алгоритма метода барьерных функций порождается последовательность внутренних точек $\{x_k\}$, тогда для нее имеют место соотношения:

1) $F(x_k, \lambda_k) \geq F(x_{k+1}, \lambda_{k+1}), \quad k = 1, 2, \ldots;$

2) $f(x_k) \geq f(x_{k+1}), \quad k = 1, 2, \ldots;$

3) $F(x_k, \lambda_k) \geq f(x_k) \geq f(x^*);$

4) $\lim\limits_{k\to\infty} x_k = x^*, \ \lim\limits_{k\to\infty} F(x_k, \lambda_k) = \lim\limits_{k\to\infty} f(x_k) = f(x^*).$

Следует отметить, что функция $F(x, \lambda)$ преобразует исходную задачу нелинейного программирования при наличии ограничений в задачу без ограничений с гораздо более сложной (по сравнению с исходной задачей) структурой, но зато характеризуется наличием весьма сильных барьеров вдоль границы допустимой области.

Рассмотрим следующие примеры барьерных функций, которые могут быть использованы для реализации алгоритмов метода барьерных функций:

1) $B(x) = \dfrac{1}{\max\limits_{i=\overline{1,m}} g_i(x)}$;

2) $B(x) = \sum\limits_{i=1}^{m} b\left(-g_i(x)\right)$, где функция $b(t)$ неотрицательна и непрерывна при $t > 0$ и $\lim\limits_{t\to+0} b(t) = +\infty$ (в качестве примера функции $b(t)$ можно привести функции $b_1(t) = t^{-p}$ для $p > 0$ и $b_2(t) = -\ln t$);

3) $B(x) = \sum\limits_{i=1}^{m} \varphi_i\big(- g_i(x)\big)$, где функции $\varphi_i(t)$ неотрицательны и непрерывны при $t > 0$ и $\lim\limits_{t\to+\infty} \varphi_i(t) = +\infty$ для всех $i = \overline{1,m}$.

4) $B(x) = \sum\limits_{i=1}^{m} \dfrac{-\omega_i}{g_i(x)}$;

$$B(x) = \sum_{i=1}^{m} \frac{\omega_i}{g_i^2(x)};$$

$$B(x) = \sum_{i=1}^{m} \omega_i \ln[-g_i(x)];$$

$$B(x) = \sum_{i=1}^{m} \omega_i \big(\max\{-\ln(-g_i(x); 0\}\big)^p.$$

Для данных функций для ограничений неравенств $g_i(x) \leq 0$ коэффициенты ω_i следует полагать равным нулю до тех пор, пока не окажется нарушенным хотя бы одно из ограничений. Для активных ограничений ω_i полагаются равными единице.

При необходимости, функции $\varphi_i(t)$ выбираются такими, чтобы обеспечить выпуклость функции $B(x)$, ее непрерывность и гладкость, простоту вычисления ее значений и необходимых производных и т.п. Однако не всегда можно обеспечить данные свойства функции $B(x)$, и в таких случаях используются более универсальные методы оптимизации, таких как субградиентные методы негладкой и недифференцируемой оптимизации.

Аналогично, как и в случае метода штрафных функций рассмотрим двойственный аспект задачи нелинейного программирования основанный на методе барьерных функций.

Рассмотрим прямую задачу нелинейного программирования

$$\min\{f(x) : x \in K\},$$

которая по методу барьерных функций сводится к решению последовательности задач безусловной оптимизации

$$\min\{f(x) + \lambda B(x),\ x \in R^n\},$$

где $B(x) = \dfrac{1}{\max_{1\leq i\leq m} g_i(x)}$.

Если рассмотреть метод барьерных функций как "двойственный метод", то прямая задача имеет вид

$$\min\{f(x) : B(x) \leq 0,\ x \in R^n\}$$

или в эквивалентной форме

$$\min\{f(x) : \max_{1\leq i\leq m} g_i(x) \leq 0,\ x \in R^n\}.$$

Двойственная к данной задаче является задача безусловной оптимизации

$$\max\{\psi(\lambda) : \lambda \geq 0\},$$

где $\psi(\lambda) = \inf\limits_{x\in R^n}\{f(x) + \lambda B(x)\}$.

В данном случае исходная задача нелинейного программирования также сводится к решению двух задач безусловной оптимизации.

Первая (внешняя) задача – одномерная задача максимизации недифференцируемой функции $\psi(\lambda)$, т.е. имеем задачу

$$\max_{\lambda\geq 0}\psi(\lambda).$$

Вторая (внутренняя) задача – безусловная задача минимизации по x недифференцируемой функции $F(x, \lambda)$ при фиксированном λ, т.е. имеем задачу

$$\inf_{x\in R^n}\{f(x) + \lambda B(x)\}.$$

Таким образом на каждом этапе имеем по одной задаче недифференцируемой оптимизации, для решения которых применяются субградиентные методы.

Замечание 1. *В некоторых случаях, при решении общей задачи нелинейного программирования эффективно использовать вспомогательные задачи безусловной оптимизации, построенные на смешанном применением методов штрафных и барьерных функций. В таком случае, вместо исходной задаче нелинейного программирования*

$$f(x) \to \min,$$

$$g_i(x) \le 0 \quad (i = \overline{1,m}),$$

$$h_k(x) \le 0 \quad (k = \overline{1,l})$$

построить расширенную функцию для безусловной оптимизации вида

$$F(x,\mu) = f(x) + \mu\alpha(x) + \frac{1}{\mu}B(x),$$

в которой $\alpha(x)$ *является штрафной функцией для ограничений* $g_i(x) \le \le 0 \quad (i = \overline{1,m})$*, а* $B(x)$ *– барьерной функцией для ограничений* $h_k(x) \le \le 0 \quad (k = \overline{1,l})$*.*

С точки зрения двойственного метода, для решения данной задачи применяем двухэтапный подход с использованием ранее рассмотренных штрафных и барьерных функций. Тогда имеем внешнюю задачу максимизации недифференцируемой вогнутой функции $\varphi(\mu)$, т.е. задачу $\max\limits_{\mu \ge 0} \varphi(\mu)$, где в данном случае

$$\varphi(\mu) = \inf_{x \in R^n} \left\{ F(x,\mu) = f(x) + \mu \sum_{i=1}^{m} (\max\{0,\, g_i(x)\})^2 + \frac{1}{\mu \max_{1 \le k \le l} h_k(x)} \right\}.$$

Внутренняя задача имеет вид

$$\inf_{x \in R^n} F(x,\mu),$$

которая является также задачей недифференцируемой оптимизации, для решения которой в случае выпуклых или вогнутых функций $g_i(x) \quad (i = \overline{1,m})$ и $h_k(x) \quad (k = \overline{1,l})$ могут быть использованы субградиентные методы.

Замечание 2. *Для задач выпуклого или квазивыпуклого программирования, т.е. когда функции* $f(x)$ *и* $g_i(x) \quad (i = \overline{1,m})$ *являются выпуклыми или квазивыпуклыми в методах штрафов и барьерах, функции* $\alpha(x)$ *и* $B(x)$ *следует выбрать такими, чтобы построенная расширенная функция не выводила нас из класса задач выпуклой или квазивыпуклой оптимизации.*

1.10. Негладкие штрафные функции для задач выпуклого программирования

Рассмотрим особенности применения негладких штрафных функции для решения задач выпуклого программирования [470, 639]. Пусть задана задача выпуклого программирования (1.23)–(1.25). Для ее сведения к задаче безусловной оптимизации можно использовать точные негладкие штрафные функции. Пусть множество оптимальных решений X^* задачи (1.23)–(1.25) непусто, $f_0(x^*) = f_0^*$ при $x^* \in X^*$ и пара векторов $\{x^*, u^*\}$ образуют седловую точку функции Лагранжа $L(x,u)$, построенной для задачи выпуклого программирования (1.23)–(1.25).

Приведем некоторые негладкие штрафные функции, которые могут быть использованы для сведения задачи (1.23)–(1.25) к задаче безусловной оптимизации [470, 639].

1. Пусть $s = \{s_i\}_{i=1}^m \geq 0$ – вектор штрафных мультипликаторов. Построим штрафную функцию

$$S_1(x,s) = f_0(x) + \sum_{i=1}^m s_i \max\{0, f_i(x)\}$$

с помощью которой, задача (1.23)–1.25) сводится к решению задачи

$$\rho^* = \min_{x \in X} S_1(x,s). \tag{1.35}$$

Теорема 1.24. *Если* $s_i \geq u_i^*$ *для всех* $i = \overline{1,m}$, *тогда* $f_0^* = \rho^*$ *и* $X^* \subseteq X_1^*$, *где* X_1^* *– множество точек минимума функции* $S_1(x,s)$. *Если* $s_i > u_i^*$, *тогда* $X^* = X_1^*$, *т.е. задача выпуклого программирования* (1.23)–(1.25) *является эквивалентной задачи* (1.35).

2. Пусть $p > 0$ штрафной множитель. Построим штрафную функцию

$$S_2(x,p) = f_0(x) + p \max[0, \max_{1 \leq i \leq m} f_i(x)]$$

с помощью которой, задача (1.23)–(1.25) сводится к решению задачи

$$\min_{x\in X} S_2(x,p). \tag{1.36}$$

Теорема 1.25. *Если* $\overline{p} \geq \sum_{i=1}^{m} u_i^*$, *тогда* $\min_{x\in X} S_2(x,\overline{p}) = f_0^*$, *где* $u^* = \{u_i^*\}$ *оптимальный вектор множителей Лагранжа задачи* (1.23)–(1.25). *Если* $\overline{p} > \sum_{i=1}^{m} u_i^*$, *то область минимумов задачи* (1.36) *совпадает с* X^*, *т.е. задача выпуклого программирования* (1.23)–(1.25) *является эквивалентной задачи* (1.36).

3. Пусть заданы одномерные выпуклые функции $\alpha_i(t)$ $(i=\overline{1,m})$, принимающие значение 0 при $t \leq 0$ и $\alpha_i(t) > 0$ для $t > 0$. Построим штрафную функцию

$$S_3(x,\alpha) = f_0(x) + \sum_{i=1}^{m} \alpha_i(f_i(x)),$$

с помощью которой, задача (1.23)–(1.25) сводится к решению задачи безусловной оптимизации

$$\min_{x\in X} S_3(x,\alpha), \tag{1.37}$$

где $\alpha = \{\alpha_i\}$ – вектор функция. Предположим, что существует точка x^*, которая минимизирует функцию $S_3(x,\alpha)$.

Пусть $\alpha_i'(+0) = \lim\limits_{t\to+0} \dfrac{\alpha_i(t)}{t}$ $(i=\overline{1,m})$, где $t \to +0$ означает, что t стремится к нулю справа, а $\alpha_i'(+0)$ – предельные значения.

Теорема 1.26. *Для того, чтобы* x^* *было бы решение задачи выпуклого программирования* (1.23)–(1.25), *необходимо, чтобы* $\alpha_i'(+0) \geq u_i^*$ *для всех* $i=\overline{1,m}$, *где* $u^* = \{u_i^*\}$ *оптимальный вектор множителей Лагранжа задачи* (1.23)–(1.25). *Если* $\alpha_i'(+0) > u_i^*$ $(i=\overline{1,m})$, *то область минимумов задачи* (1.37) *совпадает с* X^*, *т.е. задача выпуклого программирования* (1.23)–(1.25) *является эквивалентной задачи* (1.37).

4. Пусть задана одномерная выпуклая функция $\beta(t)$, принимающая значение 0 при $t \leq 0$ и $\beta(t) > 0$ при $t > 0$. Построим штрафную функцию

$$S_4(x,\beta) = f_0(x) + \beta(\max_{1\leq i\leq m} f_i(x))$$

с помощью которой, задача (1.23)–(1.25) сводится к решению задачи безусловной оптимизации.

$$\min_{x \in X} S_4(x, \beta). \tag{1.38}$$

Пусть $\beta'(+0) = \lim\limits_{t \to +0} \dfrac{\beta(t)}{t}$ – предельное значение.

Теорема 1.27. *Если* $\beta'(+0) > \sum\limits_{i=1}^{m} u_i^*$, *то область минимумов задачи* (1.38) *совпадает с* X^*, *т.е. задача выпуклого программирования* (1.23)–(1.25) *является эквивалентной задачи* (1.38).

Для решения задач минимизации функций $S_1(x, s)$, $S_2(x, p)$, $S_3(x, \alpha)$ и $S_4(x, \beta)$ можно использовать субградиентные методы негладкой оптимизации.

Практические значения теорем 1.24–1.27 весьма велики. Дело в том, что обычные гладкие функции штрафа, например с квадратичным штрафом вида

$$S(x) = f_0(x) + \sum_{i=1}^{m} s_i \cdot (\max\{0, f_i(x)\})^2, \tag{1.39}$$

при минимизации при любых конечных положительных параметров $\{s_i\}_{i=1}^m$ дают лишь приближение к оптимальному решению задачи (1.39). Для получения решения с большой точностью при использовании штрафных функций вида (1.39) нужно работать с большими коэффициентами штрафа, при этом минимизируемая функция становится плохо обусловленной. Для избежания этих трудностей предложен подход, основанный на модифицированных функциях Лагранжа, но при этом задачу безусловной минимизации нужно решать многократно.

При использовании точных гладких функций штрафа, как правило, удается работать с умеренными значениями штрафных параметров, используя априорные оценки множителей Лагранжа. Также, субградиентные методы негладкой оптимизации с растяжением пространства типа r-алгоритмов оказались устойчивыми по отношению к плохо обусловленным задачам. Это позволяет практически эффективно использовать метод негладких штрафов при решении сложных задач выпуклого программирования.

1.11. Негладкие штрафные функции для задач дробно-выпуклого программирования

Рассмотрим теперь аналогичные штрафные функции для решения задачи дробно-выпуклого программирования (1.26)–(1.28). Так как для задачи дробно-выпуклого программирования можно построить функцию Лагранжа двух типов $L_1(x,u)$ и $L_2(x,u)$, то возможны два варианта построения негладких штрафных функций.

Пусть множество оптимальных решений X^* задачи (1.26)–(1.28) непусто, $f_0(x^*) = f_0^*$ при $x^* \in X^*$ и пара векторов $\{x^*, u^*\}$ образуют седловую точку функции Лагранжа $L_1(x,u)$ (пара $\{x^*, \overline{u}^*\}$ образуют седловую точку функции $L_2(x,u)$).

1. Пусть $s = \{s_i\}_{i=1}^m \geq 0$ – вектор штрафных мультипликаторов. Построим штрафные функции

$$Q_1(x,s) = \frac{\varphi(x)}{\psi(x)} + \sum_{i=1}^m s_i \max\{0, f_i(x)\},$$

$$P_1(x,s) = \frac{\varphi(x) + \sum\limits_{i=1}^m s_i \max\{0, f_i(x)\}}{\psi(x)}$$

с помощью которых, задача (1.26)–(1.28) сводится к решению одной из задач

$$\min_{x\in X} Q_1(x,s), \tag{1.40}$$

$$\min_{x\in X} P_1(x,s). \tag{1.41}$$

Теорема 1.28. *Если $s_i \geq u_i^*$ для всех $i = \overline{1,m}$, где u_i^* оптимальный вектор множителей Лагранжа функции $L_1(x,u)$ задачи* (1.26)–(1.28), *тогда $f_0^* = \min\limits_{x\in X} Q_1(x,s)$ и $X^* \subseteq X_1^*$, где X_1^* множество точек оптимума функции $Q_1(x,s)$. Если $s_i > u_i^*$, тогда $X^* = X_1^*$, т.е. задача дробно-выпуклого программирования* (1.26)–(1.28) *эквивалентна задачи* (1.40).

Теорема 1.29. *Если* $s_i \geq \overline{u}_i^*$ *для всех* $i = \overline{1,m}$, *где* $\overline{u}^* = \{\overline{u}_i^*\}$ *оптимальный вектор множителей Лагранжа функции* $L_2(x,u)$ *задачи дробно-выпуклого программирования* (1.26)–(1.28), *тогда* $f_0^* = \min\limits_{x\in X} P_1(x,s)$ *и* $X^* \subseteq X_1^*$, *где* X_1^* *множество точек оптимума функции* $P_1(x,s)$. *Если* $s_i > \overline{u}_i^*$, *тогда* $X^* = X_1^*$, *т.е. задача дробно-выпуклого программирования* (1.26)–(1.28) *эквивалентна задачи* (1.41).

2. Пусть $p > 0$ штрафной множитель. Построим штрафные функции

$$Q_2(x,p) = \frac{\varphi(x)}{\psi(x)} + p\max[0, \max_{1\leq i\leq m} f_i(x)],$$

$$P_2(x,p) = \frac{\varphi(x) + p\max[0, \max\limits_{1\leq i\leq m} f_i(x)]}{\psi(x)}$$

с помощью которых, задача (1.26), (1.27) сводится к решению одной из задач

$$\min_{x\in X} Q_2(x,p), \tag{1.42}$$

$$\min_{x\in X} P_2(x,p). \tag{1.43}$$

Теорема 1.30. *Если* $\overline{p} \geq \sum\limits_{i=1}^{m} u_i^*$, *где* $u = \{u_i^*\}$ *оптимальный вектор множителей Лагранжа функции* $L_1(x,u)$ *задачи дробно-выпуклого программирования* (1.26)–(1.28), *тогда* $f_0^* = \min\limits_{x\in X} Q_2(x,s)$ *и* $X^* \subseteq X_1^*$, *где* X_1^* *множество точек оптимума функции* $Q_2(x,s)$. *Если* $\overline{p} > \sum\limits_{i=1}^{m} u_i^*$, *то область минимумов задачи* (1.42) *совпадает с* X^*, *т.е. задача дробно-выпуклого программирования* (1.26)–(1.28) *совпадает с задачей* (1.42).

Теорема 1.31. *Если* $\overline{p} \geq \sum\limits_{i=1}^{m} \overline{u}_i^*$, *где* $\overline{u}^* = \{\overline{u}_i^*\}$ *оптимальный вектор множителей Лагранжа функции* $L_2(x,u)$ *задачи дробно-выпуклого программирования* (1.26)–(1.28), *тогда* $f_0^* = \min\limits_{x\in X} P_2(x,p)$ *и* $X^* \subseteq X_1^*$, *где* X_1^* *множество точек оптимума функции* $P_2(x,p)$. *Если* $\overline{p} > \sum\limits_{i=1}^{m} \overline{u}_i^*$, *то область минимумов задачи* (1.43) *совпадает с* X^*, *т.е. задача дробно-выпуклого программирования* (1.26)–(1.28) *эквивалентна задачи* (1.43).

3. Пусть заданы одномерные выпуклые функции $\alpha_i(t)$ $(i = \overline{1,m})$, принимающие значение 0 при $t \le 0$ и $\alpha_i(t) > 0$ для $t > 0$. Построим штрафные функции

$$Q_3(x,\alpha) = \frac{\varphi(x)}{\psi(x)} + \sum_{i=1}^{m} \alpha_i(f_i(x)),$$

$$P_3(x,\alpha) = \frac{\varphi(x) + \sum\limits_{i=1}^{m} \alpha_i(f_i(x))}{\psi(x)}$$

с помощью которых, задача дробно-выпуклого программирования (1.26)–(1.28) сводится к решению одной из задач безусловной оптимизации

$$\min_{x\in X} Q_3(x,\alpha), \tag{1.44}$$

$$\min_{x\in X} P_3(x,\alpha). \tag{1.45}$$

Предположим, что существует точка x^*, которая минимизирует функцию $Q_3(x,\alpha)$ или $P_3(x,\alpha)$.

Теорема 1.32. *Для того, чтобы x^*, было бы решение задачи дробно-выпуклого программирования* (1.26)–(1.28)*, необходимо, чтобы $\alpha_i'(+0) \ge u_i^*$ $(i = \overline{1,m})$, где $u^* = \{u_i^*\}$ оптимальный вектор множителей Лагранжа функции $L_1(x,u)$ задачи дробно-выпуклого программирования* (1.26)–(1.28)*. Если $\alpha_i'(+0) > u_i^*$ $(i = \overline{1,m})$, то область минимумов задачи* (1.44) *совпадает с X^*, т.е. задача дробно-выпуклого программирования* (1.26)–(1.28) *является эквивалентной задачи* (1.44).

Теорема 1.33. *Для того, чтобы x^*, было бы решение задачи дробно-выпуклого программирования* (1.26)–(1.28)*, необходимо, чтобы $\alpha_i'(+0) \ge \overline{u}_i^*$ $(i = \overline{1,m})$, где $\overline{u}^* = \{\overline{u}_i^*\}$ оптимальный вектор множителей Лагранжа функции $L_2(x,u)$ задачи дробно-выпуклого программирования* (1.26)–(1.28)*. Если $\alpha_i'(+0) > \overline{u}_i^*$ $(i = \overline{1,m})$, то область минимумов задачи* (1.45) *совпадает с X^*, т.е. задача дробно-выпуклого программирования* (1.26)–(1.28) *является эквивалентной задачи* (1.45).

4. Пусть задана одномерная выпуклая функция $\beta(t)$, принимающая значение 0 при $t \le 0$ и $\beta(t) > 0$ для $t > 0$. Построим штрафные функции

$$Q_4(x,\beta) = \frac{\varphi(x)}{\psi(x)} + \beta\big(\max_{1\le i\le m} f_i(x)\big),$$

$$P_4(x,\beta) = \frac{\varphi(x) + \beta\big(\max\limits_{1\le i\le m} f_i(x)\big)}{\psi(x)}$$

с помощью которых, задача дробно-выпуклого программирования (1.26)–(1.28) сводится к решению одной из задач безусловной оптимизации

$$\min_{x\in X} Q_4(x,\beta), \tag{1.46}$$

$$\min_{x\in X} P_4(x,\beta). \tag{1.47}$$

Теорема 1.34. *Если* $\beta'(+0) > \sum\limits_{i=1}^{m} u_i^*$, *где* $u^* = \{u_i^*\}$ *оптимальный вектор множителей Лагранжа функции* $L_1(x,u)$ *задачи дробно-выпуклого программирования* (1.26)–(1.28), *то область минимумов задачи* (1.46) *совпадает с* X^*, *т.е. задача дробно-выпуклого программирования* (1.26)–(1.28) *эквивалентна задачи* (1.46).

Теорема 1.35. *Если* $\beta'(+0) > \sum\limits_{i=1}^{m} \overline{u}_i^*$, *где* $\overline{u}^* = \{\overline{u}_i^*\}$ *оптимальный вектор множителей Лагранжа функции* $L_2(x,u)$ *задачи дробно-выпуклого программирования* (1.26)–(1.28), *то область минимумов задачи* (1.47) *совпадает с* X^*, *т.е. задача дробно-выпуклого программирования* (1.26)–(1.28) *эквивалентна задачи* (1.47).

В общем случае, для задачи дробно-выпуклого программирования (1.26)–(1.28) могут быть построены следующие два типа негладких штрафных функций:

$$Q(x) = \frac{\varphi(x)}{\psi(x)} + S(x);$$

$$P(x) = \frac{\varphi(x) + S(x)}{\psi(x)},$$

где $S(x)$ определяется по одной из формул:

$$S(x) = \sum_{i=1}^{m} s_i \max\{0, f_i(x)\},$$

$$S(x) = p\,\max\{0, \max_{1\le i\le m} f_i(x)\},$$

$$S(x) = \sum_{i=1}^{m} \alpha_i(f_i(x)),$$

$$S(x) = \beta\big(\max_{1\le i\le m} f_i(x)\big).$$

Тогда решение задачи дробно-выпуклого программирования (1.26)–(1.28) сводится к решению одной из задач безусловной оптимизации

$$\min_{x\in X} Q(x)$$

или

$$\min_{x\in X} P(x),$$

которые являются задачами недифференцируемой оптимизации, для решения которых применяются субградиентные методы.

Как и в случае функций Лагранжа $L_1(x,u)$ и $L_2(x,u)$ для задачи дробно-выпуклого программирования, штрафные функции $Q(x)$ и $P(x)$ являются квазивыпуклыми. Действительно, штрафные функции $S_1(x), S_2(x), S_3(x)$ и $S_4(x)$ являются выпуклыми, притом $S_t(x) \ge 0$ $(t = \overline{1,4})$ для любого $x \in E^n$. Тогда выпуклая функция $\varphi(x) + S(x) \ge 0$ для любого $x \in E^n$. Тогда функция

$$P(x) = \frac{\varphi(x) + S(x)}{\psi(x)}$$

по определению является квазивыпуклой.

Как уже отмечено, при $S(x) \ge 0$ сумма выпуклых и квазивыпуклых функций не всегда будет квазивыпуклой функцией. Поэтому функция $Q(x) = \dfrac{\varphi(x)}{\psi(x)} + S(x)$ не является обязательно квазивыпуклой, тогда будем иметь локальные минимумы, которые не будут глобальными.

Глава 2

Методы недифференцируемой оптимизации

Методы недифференцируемой оптимизации являются по существу обобщением различных модификаций градиентных процессов минимизации гладких функций. Такие методы основаны на понятие обобщенного градиента, введенного для недифференцируемых функций.

В данной главе приводятся три основные группы методов недифференцируемой оптимизации, которые в дальнейшем применяются для решения различных задач условной оптимизации с линейными и дробно-линейными функциями, задач выпуклого и дробно-выпуклого программирования, транспортных задач.

Эти три группы относятся к:

- методам обобщенного градиентного спуска, для которых приводятся различные способы регулировки шага движения в направление субградиента или нормированного субградиента;

- субградиентные методы с растяжением пространства в направлении субградиента или в направлении разности двух последовательных субградиентов;

- методам эллипсоидов, на основе которых разработаны эффективные (полиномиальные) алгоритмы решения задач линейного и дробно-линейного программирования, задач выпуклого программирования и оптимизации выпуклых негладких функций.

2.1. Недифференцируемая оптимизация

На практике при построении и анализе математических моделей сложных технических и экономических систем используется широкий класс недифференцируемых (негладких) функций, что приводит к необходимости решения задач недифференцируемой оптимизации. Широкое применение методов недифференцируемой оптимизации позволяет расширить диапазон используемых критериев оценки качества полученных решений, более адекватно описывать происходящие процессы.

На основании методов недифференцируемой оптимизации можно разработать более точные алгоритмы и схемы решения задач без необходимости изменения их постановок или сглаживания негладких функций в ущерб адекватности математических моделей описания реальных процессов.

Кроме сложных математических моделей технической и экономической оптимизации с нелинейными негладкими функциями, другие классы задач недифференцируемой оптимизации порождаются в результате применения различных приемов и процедур при разработке эффективных методов и алгоритмов решения задач математического программирования большой размерности, дискретной, матричной, полиномиальной или стохастической оптимизации.

По мнению Н.З. Шора [470, 716, 732], основные источники, порождающие задачи негладкой оптимизации, являются следующие теоретические предпосылки и практические приемы.

Во-первых, это задачи математического программирования большой размерности с блочной структурой и сравнительно небольшим числом связей между блоками. Использование схем декомпозиции по ограничениям, переменным или по ресурсам для решения таких задач приводит к задачам минимизации (максимизации), как правило, негладких функций от связывающих переменных или от множителей Лагранжа (двойственных оценок), соответствующих связывающим ограничениям.

Во-вторых, это задачи минимизации функции максимума. Пусть задано параметрическое семейство выпуклых функций, определенных на E^n, $\{f_i(x)\}_{i\in I}$, где $I = \{1, 2, \ldots, m\}$. Основным источником получения негладких функций в выпуклом программировании является операция взятия поточечного максимума по параметру i, т.е. построение функции максимума:

$$F(x) = \sup_{i\in I} f_i(x).$$

Область X определения функции $F(x)$ совпадает с такими значениями $x \in E^n$, при которых $\{f_i(x)\}$ ограничена сверху по i некоторым участком функции $f_i(x)$. Для каждого $\overline{x} \in X$ определим подмножество индексов

$$I(\overline{x}) = \{i \in I :\ f_i(\overline{x}) = F(\overline{x})\}.$$

Субградиентное множество $G_F(\overline{x})$ функции F в точке $\overline{x}$ определяется формулой

$$G_F(\overline{x}) = \overline{conv}\{\cup_{i\in I(\overline{x})} G_{f_i}(\overline{x})\},$$

где $\overline{conv}\{M\}$ обозначает операцию нахождения минимального выпуклого замкнутого множества, содержащего M, $G_{f_i}(\overline{x})$ – субградиентные подмножества функций f_i в точке $\overline{x}$, $i \in I(\overline{x})$. Если все функции f_i $(i \in I(\overline{x}))$ являются дифференцируемыми в точке $\overline{x}$, то $G_{f_i}(\overline{x})$ состоит из единственной точки, совпадающей с $g_{f_i}(\overline{x})$, и формула принимает следующий вид:

$$G_F(\overline{x}) = \overline{conv}\{\cup_{i\in I(\overline{x})} g_{f_i}(\overline{x})\}.$$

В случае, когда $I(\overline{x})$ – конечное множество, то все крайние точки множества $G_F(\overline{x})$ являются градиентами некоторых функций f_i, $i \in I(\overline{x})$, в точке x, и $G_F(\overline{x})$ представляет собой выпуклый многогранник соответствующей размерности.

Третий источник негладких оптимизационных задач – лагранжевы оценки в задачах математического программирования. Рассмотрим достаточно общую задачу математического программирования, ограничения которой разбиты на две части, часть из которых имеет вид условия принадлежности $x \in X \subseteq E^n$, а другая часть определяется системой равенств: найти

$$f_0^* = \inf_{x\in X} f_0(x), \quad X \subseteq E^n,$$

при ограничениях

$$f_i(x) = 0 \quad (i = \overline{1,m}).$$

Предположим, что X – замкнутое подмножество n-мерного евклидового пространства, $f_i(x)$ – непрерывные функции, определенные на X.

Для оценки f_0^* введем функцию Лагранжа

$$L(x,u) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x),$$

где $u = \{u_1, \ldots, u_m\}$ – вектор множителей Лагранжа. Рассмотрим оценку

$$\psi(u) = \inf_{x\in X} L(x,u).$$

При любом допустимом x и произвольном u имеем $\psi(u) \leq f_0(x)$, откуда следует что $\psi(u) \leq f_0^*$.

Задача нахождения наилучшей оценки для оптимального значения рассматриваемой задачи математического программирования в данном классе лагранжевых оценок сводится к решению координирующей задачи: найти

$$\psi^* = \sup_u \psi(u).$$

Функция $\psi(u)$ является вогнутой как результат операции минимизации по $x \in X$ параметрического (u – параметр) семейства линейных по u функций. Предположим, что $\psi(u)$ – собственная вогнутая функция с непустой областью $\operatorname{dom}\psi$, имеющей внутренние точки. Пусть $\overline{u}$ – некоторая внутренняя точка $\operatorname{dom}\psi$, т.е. $\overline{u} \in \operatorname{int}\operatorname{dom}\psi$.

Тогда по правилам вычисления субградиента от функции максимума $\psi(u)$ субградиентное множество $G_\psi(x)$ определяется следующим образом:

$$G_\psi(\overline{u}) = \overline{conv}\{\bigcup_{x \in X(\overline{u})} F\left(x(\overline{u})\right)\},$$

где $X(u)$ – множество всевозможных решений локальной задачи $\inf_{x \in X} L(x,u)$; $F\left(x(u)\right)$ – вектор "невязок", соответствующий решению $x(\overline{u})$, $F\left(x(\overline{u})\right) = \{f_1\left(x(\overline{u})\right), \ldots, f_m\left(x(\overline{u})\right)\}$. Таким образом, если в точке $\overline{u}$ локальная задача имеет единственное решение, то ψ в соответствующей точке дифференцируема и ее градиент совпадает с вектором невязок $\{f_i\left(x(\overline{u})\right)\}_{i=1}^m$. В противном случае, как правило, градиент функции $\psi(\overline{u})$ в соответствующей точке терпит разрыв.

Четвертый источник – задачи минимизации функции максимума, характерные для моделей игрового характера, "многокритериальных" моделей оптимального планирования и исследования операций. К такого рода задачам сводятся задачи решения систем уравнений и неравенств, определения коэффициентов нелинейной регрессии, когда в качестве критерия используется чебышевский критерий минимизации максимума невязки (модуля невязки).

Пятый источник составляют задачи нелинейного программирования, для решения которых используется метод негладких штрафных функций. Негладкие штрафные функции определенного вида обладают несомненным преимуществом по сравнению с обычно применяемыми гладкими функциями штрафа: при использовании негладких штрафных функций, как правило, нет необходимости устремлять штрафные коэффициенты к $+\infty$.

Шестой источник – задачи оптимального управления с непрерывным и дискретным временем. Использование принципа максимума или дискретного принципа максимума во многих случаях приводит к задачам минимизации функций с разрывным градиентом. Эти задачи можно рассматривать как специальные задачи нелинейного программирования, для решения которых применимы схемы декомпозиции или метод негладких штрафных функций.

Седьмой источник составляют задачи дискретного программирования или задачи смешанного дискретно-непрерывного типа. Многие задачи такого рода достаточно успешно могут решаться с использованием метода ветвей и границ с получением оценок путем решения двойственной задачи. Двойственная задача обычно оказывается задачей минимизации выпуклой кусочно-линейной функции с огромным числом "кусков" при простых ограничениях, т.е. задачей негладкой оптимизации.

И наконец, функции с разрывным градиентом могут непосредственно входить в модель задачи оптимального планирования, проектирования или исследования операций как результат кусочно-гладкой аппроксимации технико-экономических характеристик реальных объектов.

Следует также отметить, что с прикладной точки зрения нет резкой границы между негладкими и гладкими функциями. С позиций прикладной математики и вычислительной практики функция с очень быстро меняющимся градиентом близка по своим свойствам к негладкой функции. Поэтому вычислительные методы, разработанные для решения задач негладкой оптимизации, оказываются эффективными и для оптимизации "плохих" гладких функций (например, функций овражного типа). Многочисленные приложения алгоритмов негладкой оптимизации для указанных классов задач можно найти в монографиях Н.З. Шора [469, 470, 639, 715, 733].

Использование методов недифференцируемой оптимизации в процессах принятия решений в сложных технических и экономических систем привело к появлению новых математических приемов и алгоритмов оптимизации, среди которых отметим следующие.

Полиномиальные алгоритмы [705, 706]. Методы недифференцируемой оптимизации легли в основу разработки полиномиальных алгоритмов решения задач линейного и дробно-линейного программирования.

Двойственные оценки [467, 470, 471, 727]. Применение методов негладкой оптимизации и функции Лагранжа позволяет оценить сложность различных трудных задач с помощью получения двойственных оценок анализа комбинаторных задач, задач теории графов и глобальной минимизации полиномиальных функций.

Схемы декомпозиции [470, 715, 724, 726, 733, 734]. Схемы декомпозиции (по переменным, ограничениям или ресурсам), основанные на функциях Лагранжа и методах недифференцируемой оптимизации, разработаны для задач выпуклого и квадратичного программирования, задач дробно-выпуклого и дробно-линейного программирования, линейных задач и задач транспортно-производственного типа.

Негладкие штрафные функции [470, 471, 639]. Негладкие штрафные функции применяются в задачах выпуклого программирования, которые позволяют свести их к задачам безусловной минимизации, полностью эквивалентных первоначальным. Применение негладких штрафных функций при использовании априорных оценок множителей Лагранжа позволяет работать с умеренными значениями штрафных параметров. Использование штрафных функций особенно полезно при решении задач с большим числом ограничений.

Развитие методов недифференцируемой оптимизации связано прежде всего с именем академика Н.З. Шора, который предложил первую версию метода обобщенного градиентного спуска для решения сетевой транспортной задачи [725]. В дальнейших работах [468, 711], [713]–[722], [729]–[732] он совместно с учениками обобщили этот метод для общего случая минимизации выпуклых негладких функций и разработали новые методы недифференцируемой (негладкой) оптимизации.

Среди методов негладкой оптимизации, нашедших практическое применение, можно выделить следующие.

1. *Обобщенный градиентный спуск* (*ОГС*). Это класс немонотонных субградиентных процессов минимизации выпуклых функций, основанных на движении в направлении, которое дает уменьшение расстояния до точки минимума, если шаговый множитель достаточно мал. К преимуществам этого метода следует отнести простоту реализации, а к недостаткам – медленную, как правило, сходимость.

2. *Обобщенные градиентные методы с растяжением пространства* (*ОГСРП*). В них, для ускорения сходимости, используются преобразование метрики пространства. Этот класс методов показал свою высокую эффективность в схемах декомпозиции по ограничениям и по переменным, в частности при решении задач производственно-транспортного типа.

3. *Методы эллипсоидов.* Использование в методах оптимизации секущих гиперплоскостей позволяет осуществить процесс аппроксимации графика функции или последовательного уменьшения объема области локализации экстремума. К числу методов этого типа относится – метод эллипсоидов и его модификации. С другой стороны, метод эллипсоидов

можно отнести к методам обобщенного градиентного спуска с растяжением пространства.

4. *ε-субградиентные методы.* Они представляют собой монотонные процессы, основанные на выборе устойчивого направления спуска за счет аппроксимации ε-субградиентного множества в текущей точке путем вычисления субградиентов в окрестности этой точки.

5. *Стохастические субградиентные методы.* Среди них можно выделить стохастические субградиентные процессы, которые по сравнению с детерминированными обладают более медленной сходимостью. Области их эффективного применения связана с задачами стохастического программирования, в которых вычисление стохастического субградиента требует гораздо меньше времени по сравнению с вычислением субградиента.

В дальнейшем приводим краткое описание разработанных в Институте Кибернетики Академии наук Украины академиком Н.З. Шором и его учениками методов недифференцируемой оптимизации.

1. Методы обобщенного градиентного спуска, положившие начало новому направлению математического программирования – численным методам негладкой оптимизации, которому в настоящее время посвящены многочисленные научные статьи и монографии.

2. Субградиентные методы с растяжением пространства в направлении субградиента, которые имеют ускоренную сходимость в сравнении с методами обобщенного градиентного спуска. Эти методы дали теории оптимизации уникальный алгоритм – метод эллипсоидов, скорость сходимости которого зависит лишь от размерности пространства. Использование метода эллипсоидов позволило решить ряд важных вопросов в теории сложности задач математического программирования.

3. Субградиентные методы с растяжением пространства в направлении разности двух последовательных субградиентом – r-алгоритмы. В рамках этого семейства методов получены достаточно эффективные реализации r-алгоритмов. Число итераций для нахождения оптимального значения f^* с ε-точностью для функций от n переменных эмпирически оценивается как $N = O\left(n \log \frac{1}{\varepsilon}\right)$. Разработанные модификации r-алгоритма являются эффективным средством минимизации выпуклых негладких функций. При минимизации гладких функций они оказались конкурентоспособными с наиболее удачными реализациями методов сопряженных направлений и методов квазиньютоновского типа.

r-алгоритмы используются в задачах оптимизации большой размерности и в квазиблочных задачах с различными схемами декомпозиции, для вычисления двойственных лагранжевых оценок в много экстремальных и комбинаторных задачах оптимизации. На практике он применялся для решения задач оптимального планирования, оптимального проектирования, синтеза сетей, восстановления изображений, эллипсоидальной аппроксимации и локализации и др.

Рассмотрим в дальнейшем вычислительные аспекты и теоретическое обоснование разработанных методов недифференцируемой оптимизации, без приведения доказательств основных теорем. Более детальные результаты в этом направление приведены в монографиях Н.З. Шора и его учениками [469, 470], [714]–[716], [639, 640, 733].

2.2. Методы обобщенного градиента

Одними из первых методов недифференцируемой оптимизации стали методы обобщенного градиентного спуска (ОГС).

Пусть $f(x)$ – выпуклая функция, определенная на евклидовом пространстве E^n; M^* – множество минимумов (оно может быть и пустым); $x^* \in M^*$ – точка минимума; $\inf f(x) = f^*$; $g_f(x)$ – субградиент (произвольный) функции $f(x)$ в точке x.

Пусть $\overline{x} \in E^n$. Из основного неравенства для субградиентов

$$f(x) - f(x_0) \geq (g_f(x_0),\, x - x_0),\ \ \textit{для}\ \forall\, x \in E^n$$

получаем $f(x) - f(\overline{x}) \geq (g_f(\overline{x}), x - \overline{x})$, $x \in E^n$. Если $f(x) < f(\overline{x})$, то

$$(-g_f(\overline{x}),\, x - \overline{x}) > 0. \tag{2.1}$$

Геометрически это означает, что антисубградиент в точке $\overline{x}$ образует острый угол с произвольным направлением, проведенным из $\overline{x}$ в сторону точки x с меньшим значением $f(x)$. Отсюда, если M^* непусто и $\overline{x} \notin M^*$, при движении из $\overline{x}$ в направлении $-g_f(\overline{x})$ с достаточно малым шагом расстояние до M^* убывает. Этот факт лежит в основу субградиентного метода, или метода обобщенного градиентного спуска (ОГС).

Метод ОГС состоит в построении последовательности точек $\{x_k\}_{k=0}^{\infty}$ (x_0 – заданная начальная точка) по формуле

$$x_{k+1} = x_k - h_k(x_k)\, g_f(x_k),\ \ k = 0,\, 1,\, 2, \ldots, \tag{2.2}$$

где $\{h_k(x_k)\}_{k=0}^{\infty}$ – последовательность положительных чисел, $h_k(x_k)$ называется шаговым множителем, который, вообще говоря, зависит от x_k; $g_f(x_k)$ – произвольный субградиент функции $f(x)$ в точке x_k. Если $g_f(x_{k^*}) = 0$, то x_{k^*} – точка минимума, и процесс останавливается.

На основании процедуры (2.2) построения последовательности $\{x_k\}_{k=0}^{\infty}$ получены различные варианты метода ОГС в зависимости от способа задания величины шага $h_k(x_k)$.

1. Метод ОГС с фиксированным шагом. Рассмотрим вариант метода ОГС, когда величина шага фиксирована. Пусть задано постоянное число $h > 0$, а величина шага $h_k(x_k)$ определяется с учетом нормы обобщенного градиента функции $f(x)$ в точке x_k следующим образом:

$$h_k(x_k) = h/||g_f(x_k)||, \ h > 0.$$

Тогда получим формулу вычисления последовательности $\{x_k\}$:

$$x_{k+1} = x_k - h\frac{g_f(x_k)}{||g_f(x_k)||}, \ k = 0, 1, 2, \ldots \tag{2.3}$$

Метод ОГС, определенный формулой (2.3), назовем методом с постоянным шагом и нормированным обобщенным градиентом.

Имеет место

Теорема 2.1. *Пусть $f(x)$ – выпуклая функция с непустым множеством минимумов M^*. Тогда для любых $\varepsilon > 0$ и $x^* \in M^*$ найдутся такие $k = k^*$ и $\overline{x}$, что при использовании ОГС с постоянным шагом h для построения последовательности $\{x_k\}$ по формуле* (2.3) *будет выполняться свойство $f(\overline{x}) = f(x_{k^*})$, причем $||\overline{x} - x^*|| < h(1+\varepsilon)/2$.*

Следствие 2.1. *В условиях теоремы* 2.1 *для произвольного $\delta > 0$ существует такое $h_\delta > 0$, что при применении метода ОГС с постоянным шагом $h = h_\delta$ при любом $x_0 \in E^n$ найдется либо такое k^*, что $x_{k^*} \in M^*$, либо такая подпоследовательность $k_1 < k_2 < k_3 < \cdots < k_l$, что $f(x_{k_l}) - f^* < \delta$, где $f^* = \min\limits_{x \in E^n} f(x)$.*

Следствие 2.2. *Если для функции $f(x)$ множество M^* содержит сферу радиуса $r > h/2 > 0$, то при использовании метода ОГС с постоянным шагом h найдется такое k^*, что $x_{k^*} \in M^*$.*

2. Метод ОГС со сходящимся рядом шагового множителя и нормированием градиента. Рассмотрим модификацию метода ОГС, когда величина шага не является постоянной, а меняется на каждом его шаге. Из теоремы 2.1 видно, что с уменьшением шага h уменьшается и расстояние до точек области минимумов на данном шаге гарантируется только в том случае, если мы находимся за пределами некоторой окрестности области минимумов, зависящей от шага h. Поэтому, чтобы получить обычные теоремы сходимости, нужно, чтобы h_k стремилось к нулю не слишком быстро. В частности, если ряд $\sum\limits_{k=1}^{\infty} h_k$ будет сходящимся, то последовательность $\{x_k\}_{k=0}^{\infty}$ будет сходиться, но не обязательно к точке, принадлежащей M^*. Таким образом, мы приходим к ставшими уже классическими условиям:

$$h_k > 0; \quad h_k \xrightarrow[k \to \infty]{} 0; \quad \sum_{k=1}^{\infty} h_k = +\infty. \tag{2.4}$$

По методу ОГС определим последовательность $\{x_k\}$ по формуле

$$x_{k+1} = x_k - h_{k+1} \frac{g_f(x_k)}{||g_f(x_k)||}, \; k = 0, 1, 2, \ldots \tag{2.5}$$

Имеет место

Теорема 2.2. *Пусть $f(x)$ – выпуклая функция, определенная на E^n, с ограниченной областью минимумов M^*, $\{h_k\}$ $(k = 1, 2, \ldots)$ – последовательность положительных чисел, обладающая свойствами* (2.4). *Тогда последовательность $\{x_k\}$ $(k = 0, 1, 2, \ldots)$, образованная по формуле (2.5), при произвольном $x_0 \in E^n$ обладает одним из следующих свойств: либо найдется такое $k = k^*$, что $x_{k^*} \in M^*$, либо*

$$\lim_{k \to \infty} \min_{x \in M^*} ||x_k - x|| = 0, \quad \lim_{k \to \infty} f(x_k) = \min_{x \in E^n} f(x) = f^*. \tag{2.6}$$

Конкретный вариант метода ОГС типа (2.5) определяется процедурой выбора величин шага h_k. Могут быть использованы различные способы, среди которых можно выделить следующие:

а) $h_{k+1} = \dfrac{h_0}{k+1}, \quad h_0 > 0, \quad k = 0, 1, 2, \ldots;$

б) $h_{k+1} = \dfrac{h_k}{k+1}, \quad h_0 > 0, \quad k = 0, 1, 2, \ldots;$

в) $h_{k+1} = \dfrac{h_k + h_{k-1}}{2}, \quad h_0 > 0, \, h_1 = \dfrac{h_0}{2}, \, k = 1, 2, \ldots$

3. Метод ОГС со сходящимся рядом шагового множителя без нормирования градиента. Другой вариант метода ОГС с переменным шагом может быть рассмотрен, когда не проводится нормирование градиента на каждой его итерации. Пусть $\{h_k\}$ – последовательность чисел, удовлетворяющих условиям (2.4). Последовательность $\{x_k\}$ по методу ОГС определим по формуле

$$x_{k+1} = x_k - h_{k+1} g_f(x_k), \; k = 0, 1, 2, \ldots, \tag{2.7}$$

где $x_0 \in E^n$ – некоторая начальная точка, а последовательность $\{h_k\}$ удовлетворяет условиям (2.4). Такой метод ОГС назовем методом без нормирования градиента. Имеет место

Теорема 2.3. *Если в условиях теоремы* 2.2 *определить последовательность* $\{x_k\}_{k=0}^{\infty}$ *формулой* (2.7), *то возможны такие случаи:*

а) *последовательность* $\{g_f(x_k)\}_{k=0}^{\infty}$ *ограничена, тогда имеет место* (2.6)*;*

б) *последовательность* $\{g_f(x_k)\}_{k=0}^{\infty}$ *неограничена, сходимость отсутствует.*

4. Метод ОГС со сходящимся рядом шагового множителя без нормирования градиента и возможностью возврата в первоначальную точку. Заметим, что если $f(x)$ – кусочно-линейная функция с конечным числом кусков, то последовательность $\{g_f(x_k)\}_{k=0}^{\infty}$ всегда ограничена, т.е. выполняется случай а) теоремы 2.3. Отметим также, что для заданного x_0 всегда можно подобрать такое $\delta > 0$, что если $\max\limits_{k \geq 1} h_k \leq \delta$, то будет выполняться случай а) теоремы 2.3.

Определим последовательность $\{x_k\}$ по методу ОГС согласно следующей формуле:

$$x_{k+1} = \begin{cases} x_k - h_{k+1} g_f(x_k), & \text{при } \ h_{k+1} ||g_f(x_k)|| \leq c, \\ x_0, & \text{в противном случае,} \end{cases} \tag{2.8}$$

где $c > 0$ – некоторая константа.

Приведем следующие две модификации теоремы 2.3.

Теорема 2.4. *Если в условиях теоремы* 2.2 *определить последовательность* $\{x_k\}_{k=0}^{\infty}$ *формулой* (2.8), *а последовательность* $\{h_k\}$ *удовлетворяет условиям* (2.4), *то для любого начального приближения* x_0 *имеет место* (2.6).

Теорема 2.5. *Если множество минимумов* M^* *функции* $f(x)$ *содержит сферу* S_r *радиуса* $r > 0$, $h_k > 0$, $\sum\limits_{k=1}^{\infty} h_k = +\infty$ *и* $\lim\limits_{k\to\infty} \sup h_k < 2r$, *то для произвольного* $x_0 \in E^n$ *при применении ОГС в форме* (2.5) *найдется* $k(x_0)$ *такое, что* $x_{k(x_0)} \in M^*$.

5. Метод ОГС сходящийся со скоростью геометрической прогрессии. При определенных дополнительных предположениях удается получить варианты ОГС, сходящиеся со скоростью геометрической прогрессии.

Теорема 2.6. *Пусть* $f(x)$ *– выпуклая функция, определенная на* E^n*, и для всех* $x \in E^n$ *при некотором* φ $(0 \le \varphi < \pi/2)$ *выполняется неравенство*

$$(g_f(x),\ x - x^*(x)) \ge \cos\varphi \cdot ||g_f(x)|| \cdot ||x - x^*(x)||, \tag{2.9}$$

где $x^*(x)$ *– точка, принадлежащая множеству минимумов функции* $f(x)$ *и лежащая на кратчайшем расстоянии от* x*. Тогда, если при заданном* x_0 *выбрать величину* h_1*, удовлетворяющую неравенству*

$$h_1 \ge \begin{cases} ||x^*(x_0) - x_0|| \cos\varphi, & \pi/4 \le \varphi < \pi/2, \\ ||x^*(x_0) - x_0||/(2\cos\varphi), & 0 \le \varphi < \pi/4, \end{cases} \tag{2.10}$$

определить $\{h_k\}_{k=1}^{\infty}$ *в соответствие с рекуррентной формулой*

$$h_{k+1} = h_k r(\varphi), \quad k = 1,\, 2, \ldots, \tag{2.11}$$

где

$$r(\varphi) = \begin{cases} \sin\varphi, & \pi/4 \le \varphi < \pi/2, \\ 1/(2\cos\varphi), & 0 \le \varphi < \pi/4, \end{cases} \tag{2.12}$$

и вычислить $\{x_k\}_{k=1}^{\infty}$ *по формуле* (2.5)*, либо при некотором* k^* $g_f(x_{k^*}) = 0$ *и* x_{k^*} *принадлежит области минимумов, либо при всех* $k = 0,\, 1,\, 2, \ldots$ *выполняется неравенство*

$$||x_k - x^*(x_k)|| \le \begin{cases} h_{k+1}/\cos\varphi, & \pi/4 \le \varphi < \pi/2, \\ 2\cos\varphi \cdot h_{k+1}, & 0 \le \varphi < \pi/4. \end{cases}$$

Для справедливости теоремы 2.6 достаточно выполнения условия (2.9) лишь для точек последовательности $\{x_k\}_{k=0}^{\infty}$.

Таким образом, если угол φ заранее известен, то, регулируя шаг по формулам (2.11), (2.12), можно получить сходимость к минимуму со скоростью геометрической прогрессии со знаменателем $q = r(\varphi)$.

6. Метод ОГС для вытянутых поверхностей уровня минимизируемой функции. В формуле (2.9) $\cos\varphi$ характеризует степень вытянутости поверхностей уровня функции $f(x)$. Если в некоторой окрестности минимума этой поверхности выполняется (2.9), то такую функцию будем называть *существенно овражной*. При минимизации существенно овражных функций приведенный в теореме 2.6 способ регулировки шаговых множителей неприменим. В этом случае нужно использовать универсальный способ выбора шаговых множителей, указанный в теореме 2.2.

Сформулируем теорему, аналогичную теореме 2.6, непосредственно в терминах, характеризующих степень "вытянутости" поверхностей уровня.

Теорема 2.7. *Пусть выпуклая функция $f(x)$ определена на E^n, x^* – единственная точка минимума f и заданы начальное приближение x_0 и числа σ и h_1, причем $\sigma \geq \sqrt{2}$, $h_1 \geq ||x_0 - x^*||/\sigma$. Рассмотрим множество $Y = \{y\colon ||y - x^*|| \leq \sigma h_1\}$. Если для любой пары точек $x, z \in Y$, такой, что $f(x) = f(z) \neq f(x^*)$, выполняется условие*

$$||x - x^*|| \leq \sigma ||z - x^*||,$$

то последовательность $\{x_k\}_{k=0}^{\infty}$, образованная с помощью рекуррентных формул

$$x_{k+1} = x_k - h_{k+1} g_f(x_k)/||g_f(x_k)||,$$

где $h_{k+1} = h_k\sqrt{\sigma^2 - 1}/\sigma$, сходится к x^* со скоростью геометрической прогрессии:

$$||x_k - x^*|| \leq h_{k+1}\sigma,$$

за исключением случая, когда для некоторого $k = \overline{k}$ имеем $g_f(x_{\overline{k}}) = 0$, т.е. $x_{\overline{k}} = x^*$.

Заметим, что для задачи минимизации квадратичных функционалов $f(x) = (Ax, x)$ величина σ, фигурирующая в теореме 2.7, может быть взята равной $\sqrt{\rho}$, при этом получаем сходимость со скоростью геометрической прогрессии со знаменателем

$$\overline{q} = \sqrt{\sigma^2 - 1}/\sigma = \sqrt{(\rho - 1)/\rho} \approx 1 - \frac{1}{2\rho} \quad (\text{для } \rho \gg 1).$$

7. Метод ОГС с постоянным, а затем с уменьшенным в два раза шагом. Рассмотрим еще один вариант метода обобщенного градиентного спуска, когда шаговый множитель остается в течение определенного числа шагов постоянным, а затем уменьшается в два раза.

Теорема 2.8. *Пусть для выпуклой функции $f(x)$ выполняются условия теоремы 2.2, $\sigma \geq 2$. Рассмотрим при заданном x_0 следующий итеративный процесс:*

$$x_{k+1} = x_k - h_{k+1}\frac{g_f(x_k)}{||g_f(x_k)||},$$

где $h_{k+1} = h_0 \cdot 2^{-[(k+1)/N]}$. При достаточно большом h_0 и $N \geq 3\sigma^2 + 1$ выполняется неравенство

$$||x_k - x^*|| \leq 2\sigma h_{k+1}, \quad k = 0, 1, 2, \ldots$$

8. Метод ОГС со стохастическим градиентом. Введение понятия обобщенного стохастического градиента и применение метода случайного поиска привело к построению стохастического аналога ОГС. Этот метод является эффективным средством решения разнообразных задач стохастического программирования, в том числе транспортных и распределительных стохастических задач. Он принадлежит к разновидности методов случайного поиска, т.е. к таким итеративным процедурам, у которых направление движения на данном шаге определяется в результате реализации некоторого случайного (псевдослучайного) события в отличие от жестко определенных процедур в обычных градиентных методах.

При минимизации выпуклой функции $f(x)$, определенной на E^n, метод обобщенного стохастического градиента задается формулой

$$x_{k+1} = x_k - h_k(x_k)g_\omega(x_k), \quad k = 0,\, 1,\, 2, \ldots,$$

где $h_k(x_k)$ – шаговый множитель на k-м шаге, $g_\omega(x_k)$ – случайный вектор, математическое ожидание которого совпадает с обобщенным градиентом функции $f(x)$ в точке x_k (стохастический субградиент в точке x_k). Предположим для простоты, что вероятностные характеристики вектора $g_\omega(x_k)$ определяются точкой x_k и не зависят от предыстории процесса поиска, хотя обоснование сходимости проходит и для более общего случая. Пусть x^* – единственная точка минимума функции $f(x)$. Справедлива

Теорема 2.9. *Пусть выполняются условия:*

1) $\sum\limits_{k=1}^{\infty} h_k(x_k) = +\infty, \quad h_k(x_k) > 0;$

2) $\sum\limits_{k=1}^{\infty} h_k^2(x_k) < +\infty;$

3) $M_\omega||g_\omega(x_k)||^2 \leq c; \quad c > 0; \;\; k = 0, 1, \ldots$ *(M_ω – символ математического ожидания).*

Тогда с вероятностью 1

$$\lim_{k\to\infty} ||x_k - x^*|| = 0.$$

Методы ОГС дали возможность решить большое число задач производственно-транспортного планирования с применением схем декомпозиции (по переменным и по ограничениям) для задач большой размерности. Подробную информацию об этих задачах можно найти в

[183, 184, 615]. Метод ОГС также послужил основой для создания стохастического аналога обобщенного градиентного спуска [615]–[617], который имеет большое практическое применение, в частности при решении многоэтапных задач стохастического программирования. В [639] описано применение метода обобщенного стохастического градиента к решению двухэтапной стохастической транспортной задачи, связанной с определением объемов складов однородной продукции при случайном спросе.

2.3. Методы субградиентного типа с растяжением пространства

При анализе алгоритмов ОГС, сходящихся со скоростью геометрической прогрессии, существенную роль играли верхние границы q синусов углов между направлением антиградиента в данной точке и направлением из нее на точку минимума. Медленная сходимость ОГС непреодолима в рамках этого метода в овражных задачах, когда верхняя граница указанных углов близка или равна $\pi/2$.

Можно изменить ситуацию, используя линейные неортогональные преобразования пространства аргументов для улучшения обусловленности задачи. В случае, когда антиградиенты образуют угол, близкий к $\pi/2$, с направлением на точку минимума, разумно применить операцию растяжения пространства в направлении градиента для уменьшения его "поперечной" составляющей. Эти эвристические соображения послужили основой создания семейства методов субградиентного типа с растяжением пространства.

Операция растяжения пространства в направлении градиента первоначально введена Н.З. Шором как эвристическая процедура для улучшения свойств обусловленности задачи. Она реализуется посредством оператора растяжения пространства.

Для последующего изложения нам понадобятся некоторые свойства операторов растяжения пространства.

Пусть заданы вектор $\xi \in E^n$, $||\xi|| = 1$, и число $\alpha \geq 0$. Каждый вектор $x \in E^n$ однозначно представим в виде

$$x = \gamma_\xi(x)\xi + d_\xi(x) \tag{2.13}$$

при условии

$$(\xi, d_\xi(x)) = 0.$$

Из этих соотношений получаем

$$\gamma_\xi(x) = (x, \xi);\ d_\xi(x) = x - (x, \xi)\xi.$$

Оператором растяжения пространства E^n в направлении ξ с коэффициентом α назовем оператор $R_\alpha(\xi)$, действующий следующим образом на вектор x, представленный в форме (2.13):

$$R_\alpha(\xi)x = \alpha\gamma_\xi(x)\xi + d_\xi(x).$$

Из этого определения вытекает ряд свойств оператора $R_\alpha(\xi)$.

1. $R_\alpha(\xi)x = \alpha(x, \xi)\xi + [x - (x, \xi)\xi] = (\alpha - 1)(x, \xi)\xi + x.$ (2.14)

2. Оператор $R_\alpha(\xi)$ – линейный симметричный:

$$(R_\alpha(\xi)x, y) = (\alpha - 1)(x, \xi)(y, \xi) + (x, y) = (x, R_\alpha(\xi)y).$$

3. $R_{\alpha\beta}(\xi) = R_\alpha(\xi)R_\beta(\xi)$.

4. При $\alpha > 0$ $R_\alpha(\xi)R_{1/\alpha}(\xi) = R_1(\xi) = I$ (I – единичный оператор).

5. Оператор $R_0(\xi)$ является оператором проектирования на подпространство, ортогональное вектору ξ:

$$R_0(\xi)x = d_\xi(x).$$

6. Оператор $R_\alpha(\xi)$ при $n \geq 2$ имеет два собственных числа $\lambda_1 = \alpha$, $\lambda_2 = 1$; первому из них соответствует подпространство собственных векторов, порожденное вектором ξ, второму – подпространство собственных векторов, состоящее из векторов, ортогональных ξ.

7. Пусть координаты вектора ξ в некоторой ортонормированной системе координат $s = \{e_1, e_2, \ldots, e_n\}$ равны $\xi_1, \xi_2, \ldots, \xi_n$, тогда в силу (2.14) в этой системе координат преобразованию $R_n(\xi)$ соответствует матрица $R_\alpha(\xi)$ с элементами $\{r_{ij}\}$, вычисленными по формулам

$$r_{ij} = (R_\alpha(\xi)e_i, e_j) = \begin{cases} (\alpha - 1)\,\xi_i\xi_j & \text{для } \ i \neq j, \\ (\alpha - 1)\,\xi_i^2 + 1 & \text{для } \ i = j. \end{cases}$$

8. В силу формулы (2.14) вычисление вектора $R_\alpha(\xi)x$ требует $(2n+1)$ операций умножения, вычисление матриц вида $R_\alpha(\xi)A$ или $AR_\alpha(\xi)$ при заданных A, ξ и α требует $n(2n + 1)$ операций умножения.

9. Пусть x – произвольный ненулевой вектор из E^n, тогда

$$||R_\alpha(\xi)x|| = \sqrt{||x||^2 + (\alpha^2 - 1)(x, \xi)^2}, \tag{2.15}$$

или $||R_\alpha(\xi)x||^2 = ||x||^2 + (\alpha^2 - 1)(x, \xi)^2$, т.е. воздействие оператора $R_\alpha(\xi)$ при $\alpha \geq 0$ $(\alpha \leq 1)$ соответственно не уменьшает (не увеличивает) длину вектора. Этот результат получается непосредственным применением формулы (2.14), в самом деле,

$$||R_\alpha(\xi)x||^2 = (R_\alpha(\xi)x, \quad R_\alpha(\xi)x) =$$

$$= (x + (\alpha - 1)(x, \xi)\xi, \quad x + (\alpha - 1)(x, \xi)\xi) =$$

$$= ||x||^2 + 2(\alpha - 1)(x, \xi)^2 + (\alpha - 1)^2(x, \xi)^2 = ||x||^2 + (\alpha^2 - 1)(x, \xi)^2,$$

откуда и следует равенство (2.15).

10. $R_\alpha(\xi)$ в матричной форме представим в виде

$$R_\alpha(\xi) = I + (\alpha - 1)\xi\xi^T;$$

действительно,

$$(I + (\alpha - 1)\xi\xi^T)x = x + (\alpha - 1)\xi(\xi, x),$$

что соответствует определяющему соотношению (2.14).

Рассмотрим класс алгоритмов минимизации выпуклых функций, на каждом шаге которых движение в направлении обобщенного градиента будет сочетаться с операцией растяжения пространства аргументов в этом же направлении. Алгоритмы этого класса будем называть алгоритмами обобщенного градиентного спуска с растяжением пространства в направлении градиента (ОГСРП-*алгоритмами*).

Предположим, что имеются: алгоритм, позволяющий точно вычислить субградиент $g_f(x)$ минимизируемой выпуклой функции $f(x)$ в произвольной точке $x \in E^n$, а также алгоритмы вычисления последовательностей положительных чисел $\{h_k\}$ и $\{\alpha_k\}$ $(k = 1, 2, \dots)$ (шаговых множителей и коэффициентов растяжения пространства), задано начальное приближение x_0 и начальная неособая матрица $B_0 = A_0^{-1}$ (в частности, $B_0 = I$). При этих условиях определим бесконечно шаговый процесс, $(k + 1)$-й шаг которого $(k = 0, 1, 2, \dots)$ описывается следующим образом. Проведем следующие вычисления.

1) Находим значения обобщенного градиента $g_f(x_k)$. Если $g_f(x_k) = 0$, вычисления прекращаются, так как x_k дает точку минимума.

2) Находим значения обобщенного градиента в растянутом пространстве

$$g_{\varphi_k}(y_k) = B_k^* g_f(x_k) = \tilde{g}_k, \tag{2.16}$$

где $\varphi_k(y_k) = f(B_k y_k)$, $y_k = A_k x_k$, $A_k = B_k^{-1}$, B_k^* – оператор, сопряженный оператору B_k, $\tilde{g}_k$ – обобщенный градиент для функции $\varphi_k(y)$, определенной в "растянутом" пространстве.

Формула (2.16) дает возможность вычислять обобщенный градиент от функции $\varphi_k(y_k) = f(A_k^{-1}y)$, которая получается из $f(x)$ при применении линейного преобразования пространства $y = A_k x$. В тех точках, где $g_f(x)$ определен неоднозначно, формулу (2.16) следует понимать как однозначное отображение множества $G_f(x)$ на множество $G_{\varphi_k}(y)$ (если оператор B_k – неособый, то отображение взаимно однозначное).

3) Находим направление растяжения пространства

$$\xi_{k+1} = \frac{g_{\varphi_k}(y_k)}{||g_{\varphi_k}(y_k)||} = \frac{\tilde{g}_k}{||\tilde{g}_k||}. \tag{2.17}$$

4) Находим величину шага h_{k+1}.

5) Находим величину коэффициента растяжения пространства α_{k+1}.

6) Осуществляем переход в новую точку

$$x_{k+1} = x_k - B_k h_{k+1} \xi_{k+1}. \tag{2.18}$$

Формула (2.18) получается из формулы

$$y_{k+1} = y_k - h_{k+1}\xi_{k+1},$$

осуществляющей шаг обобщенного градиентного спуска для функции $\varphi_k(y)$, с последующим применением к обеим частям оператора B_k для отображения в основное пространство.

7) Находим матрицу преобразования пространства

$$B_{k+1} = A_{k+1}^{-1} = B_k R_{1/\alpha_{k+1}}(\xi_{k+1}). \tag{2.19}$$

Формула (2.19) дает возможность вычислить оператор B_{k+1}, обратный результирующему оператору

$$A_{k+1} = R_{\alpha_{k+1}}(\xi_{k+1}) \dots R_{\alpha_1}(\xi_1) A_0$$

преобразования пространства, который получается при последовательном применении операторов растяжения пространства в направлении нормированных обобщенных градиентов $\xi_1, \xi_2, \dots, \xi_{k+1}$ с коэффициентами $\alpha_1, \dots, \alpha_{k+1}$:

$$B_{k+1} = A_{k+1}^{-1} = A_k^{-1} R_{\alpha_{k+1}}^{-1}(\xi_{k+1}) = B_k R_{1/\alpha_{k+1}}(\xi_{k+1}).$$

8) Переходим к $(k+2)$-му шагу.

Каждый шаг ОГСРП требует ряда дополнительных операций по сравнению с методом обобщенного градиентного спуска. Наиболее трудоемкими из них являются операции 2), 6) и 7), каждая из которых требует порядка cn^2 арифметических операций, из них около $4n^2$ операций умножения.

Заметим, что если $B_0 = I$ и $\alpha_k = 1$ $(k = 1, 2, \dots)$, то метод ОГСРП становится эквивалентным обобщенному градиентному спуску. Метод ОГСРП является более гибким по сравнению с методом ОГС, так как его конкретные реализации зависят от двух последовательностей $\{\alpha_k\}$ и $\{h_k\}$. Применяя различные алгоритмы построения этих последовательностей, можно на единой основе строить различные модификации алгоритма ОГСРП и выбирать наиболее эффективные из них для данного класса задач.

Алгоритмы типа ОГСРП первоначально были разработаны для минимизации выпуклых функций. В дальнейшем оказалось, что некоторые их модификации применимы для нахождения локальных минимумов почти дифференцируемых функций (при этом, естественно, роль обобщенного градиента в алгоритме играет почти-градиент). В дальнейшем при исследовании сходимости алгоритмов ОГСРП в каждом конкретном случае будем оговаривать, идет ли речь о классе почти дифференцируемых функций или о подклассе выпуклых функций. При этом под обобщенным градиентом будем понимать соответственно почти-градиент либо субградиент.

Определенные варианты алгоритмов ОГСРП сходятся по функционалу со скоростью геометрической прогрессии, причем знаменатель этой прогрессии зависит от таких характеристик минимизируемой функции, которые инвариантны по отношению к невырожденным линейным преобразованиям пространства. Отметим, что для класса выпуклых функций удается построить алгоритм типа ОГСРП, который сходится по функционалу со скоростью геометрической прогрессии со знаменателем, зависящим только от размерности области определения функции.

При построении других вариантов алгоритмов, которые удалось теоретически обосновать, шаговый множитель и коэффициенты растяжения пространства выбирались таким образом, чтобы последовательность расстояний до точки минимума в соответствующих преобразованиях пространствах не возрастала. Этот принцип гарантирует сходимость со скоростью геометрической прогрессии по значению функции. Для реализации указанного принципа необходимы некоторая дополнительная информация о функции $f(x)$ – значение функции в точке минимума f^* и так называемые постоянные роста M и N.

Теорема 2.10. *Пусть $f(x)$ – почти дифференцируемая функция, при минимизации которой с помощью алгоритма ОГСРП* (2.13)–(2.19) *получается минимизирующая последовательность $\{x_k\}_{k=0}^{\infty}$. Если при этом для некоторых положительных d, α^*, δ выполняются условия*

1) $||g_f(x_k)|| \le d$;

2) $1+\delta \le \alpha_k \le \alpha^*;\ k = 0, 1, 2, \dots,$

то существует такая подпоследовательность $\{x_{k_p}\}_{p=0}^{\infty}$, $k_p < k_{p+1}$ и $c > 0$, что

$$||\tilde{g}_{k_p}|| < c\Big(\prod_{j=1}^{k_p} \alpha_j\Big)^{-1/n}, \quad p = 1, 2, \dots$$

Близкий результат можно получить для оценки скорости убывания последовательности $v_k = \min\limits_{1\le r\le k} ||\tilde{g}_r||$. Этот результат в дальнейшем будет использоваться для получения оценок скорости сходимости "рекордов" по функционалу к минимуму при использовании ОГСРП.

Теорема 2.11. *В предположениях теоремы* 2.10 *при условии $\alpha_k = \alpha > 1$ справедлива оценка*

$$v_k \le \frac{d\sqrt{k(\alpha^2-1)}}{\sqrt{\alpha^{2k/n}-1}}, \quad k = 1, 2, \dots$$

Чтобы перейти к оценке скорости сходимости метода ОГСРП по функционалу, нужно получить оценки расстояния до области минимумов в "растянутом" пространстве. Для некоторых вариантов ОГСРП эти оценки удается получить.

Теорема 2.12. *Пусть $f(x)$ – почти дифференцируемая функция, определенная в некоторой сферической окрестности S_d точки x^*, являющейся локальным минимумом $f(x)$, $S_d = \{x\colon ||x - x^*|| \le d\}$, и в области S_d почти-градиент удовлетворяет неравенству*

$$N\cdot(f(x) - f(x^*)) \le (g_f(x),\, x - x^*) \le M\cdot(f(x) - f(x^*)), \qquad (2.20)$$

где $M > N$ – положительные константы. Тогда, если в алгоритме ОГСРП принять:

1) $x_0 \in S_d$,

2) $h_{k+1} = \dfrac{2MN}{M+N} \cdot \dfrac{f(x_k) - f(x^*)}{||\tilde{g}_k||}$,

3) $1 < \alpha_{k+1} \le \dfrac{M+N}{M-N}$, $k = 0, 1, 2, \ldots$,

то для всех $k = 0, 1, 2, \ldots$

$$||A_k(x_k - x^*)|| \le d. \tag{2.21}$$

Отметим, что неравенство (2.21) эквивалентно $(G(x_k - x^*), x_k - -x^*) \le d^2$, где $G = A_k^* A_k$ – положительно определенная матрица, причем

$$\det G = (\det A_k)^2 = \Big(\prod_{i=1}^{k} \det R_{\alpha_i}(\xi_i)\Big)^2 = \Big(\prod_{i=1}^{k} \alpha_i\Big)^2.$$

Множество x удовлетворяющих неравенству $||A_k(x_k - x)|| \le d$ представляет собой эллипсоид Φ_k с центром в точке x_k. Поэтому из неравенства (2.21) следует локализация x^* в эллипсоиде Φ_k с центром в точке x_k. Отношение объемов эллипсоидов Φ_{k+1} и Φ_k задается следующим равенством

$$\frac{v(\Phi_{k+1})}{v(\Phi_k)} = \beta_k = \frac{M-N}{M+N}.$$

Для квадратичной положительно определенной функции в неравенстве (2.20) можно выбирать $M = N = 2$. Для кусочно-линейной функции, надграфик которой представляет собой конус с вершиной в точке (x^*, f^*), можно выбирать $M = N = 1$. Для этих случаев $\beta_{k+1} = \beta = 0$ и алгоритм сходится за число шагов, не превышающее n.

Отметим, что семейство алгоритмов (2.18), (2.19) содержит как частный случай алгоритмы, получившие название методы эллипсоидов.

Решение невырожденной системы n линейных уравнений с n неизвестными $(a_i, x) + b_i = 0 \quad (i = \overline{1, n})$, можно заменить нахождением минимума $f(x) = \max\limits_{1 \le i \le n} |(a_i, x) + b_i|$. Беря $f^* = 0, \quad \beta_k = 0$ и применяя метод (2.18), (2.19), получаем алгоритм, соответствующий известной конечной процедуре решения линейных алгебраических систем – методу ортогонализации градиентов.

Получены обобщения теоремы 2.12 и для некоторых классов невыпуклых функций, возникающих при решении систем нелинейных уравнений $f_i(x) = 0 \quad (i = \overline{1, n})$. Для $f(x) = \max |f_i(x)|$ можно показать, что если x^* (решение системы) – регулярная точка (т.е. функции $f_i(x)$ непрерывно дифференцируемы в этой точке и якобиан системы $I(x^*)$ отличен от нуля), то для любого $\delta > 0$ найдется достаточно малая окрестность $S_d(x^*)$, такая, что постоянные M и N в (2.20) можно выбирать соответственно $M = 1 + \delta$; $N = 1 - \delta$; $\beta = \frac{M-N}{M+N} = \delta$. Если применить предельный вариант алгоритма с $\beta = 0$ и восстановлением после каждых n итераций (большой цикл), то при обычных предположениях гладкости и регулярности для решения систем нелинейных уравнений можно получить квадратичную скорость сходимости (относительно больших циклов).

Как следствие из теорем 2.10, 2.11 и 2.12 получаем следующий результат.

Теорема 2.13. *В условиях теоремы* 2.12 *при* $\alpha_k = \alpha \quad (k = 0, 1, 2, \dots, k_p)$ *найдется такая последовательность индексов* $k_1, k_2, \dots, k_p$ *и такое положительное число* c, *что*

$$f(x_{k_p}) - f(x^*) < c \cdot \alpha^{-\frac{k_p}{n}}.$$

Кроме того,

$$\min_{1 \le i \le k} (f(x_i) - f(x^*)) \le \frac{Gd\sqrt{k(\alpha^2 - 1)}}{\sqrt{\alpha^{2k/n} - 1}},$$

где $G = \max\limits_{x \in S_d} ||g_f(x)||$.

Теорема 2.12 обобщается на случай, когда точка минимума функции $f(x)$ определена неоднозначно.

Пусть множество M^* точек минимума почти дифференцируемой функции $f(x)$ ограничено и $f(x)$ принимает на M^* значение f^*. Введем обозначение

$$\rho_k(x_k) = \min_{x^* \in M^*} ||A_k(x_k - x^*)|| = ||A_k\left(x_k - x_k^*(x_k)\right)||.$$

Точку, в которой достигается минимум, обозначим через $x_k^*(x_k) \in M^*$.

Теорема 2.14. *Пусть для всех* x, x^*, *удовлетворяющих условию* $\rho_0(x) \le d,\ x \notin M^*,\ x^* \in M^*$, *выполняются неравенства*

$$N \cdot (f(x) - f^*) \le (g_f(x),\, x - x^{**}) \le M \cdot (f(x) - f^*),$$

где x^{**} *– ближайшая к* x *точка, лежащая на луче* $y = x + t(x - x^*)$ $(t \ge \ge 0)$. *Тогда, если при применении алгоритма ОГСРП принять:*

1) $\rho_0(x_0) \le d$,

2) $h_{k+1} = \dfrac{2MN}{M+N} \cdot \dfrac{f(x_k) - f^*}{||g_{\varphi_k}(y_k)||}$,

3) $1 < \alpha_{k+1} \le \dfrac{M+N}{M-N}$,

то $\rho_{k+1}(x_{k+1}) \le d \quad (k = 0, 1, 2, \dots)$.

Теорема 2.15. *Пусть выпуклая функция $f(x)$, обладает следующим свойством: существует число $M > 1$ такое, что если*

$$\varphi(\alpha) = f((1-\alpha)x_1 + \alpha x_2), \quad 0 \le \alpha \le 1,$$

строго убывает по α, то выполняется неравенство

$$(g_f(x_1), x_1 - x_2) \le M \cdot (f(x_1) - f(x_2));$$

кроме того, $\lim\limits_{||x|| \to \infty} f(x) = +\infty$. Тогда, если при применении алгоритма ОГСРП

$$\alpha_{k+1} = \frac{M+1}{M-1}, \quad h_{k+1} = \frac{2M}{M+1} \cdot \frac{f(x_k) - \overline{f}}{||\tilde{g}_k||}$$

и $\overline{f} \ge f^*$, то последовательность $\{h_k\}$ является ограниченной и для произвольного $\varepsilon > 0$ найдется $\overline{k}$ такое, что $f(x_{\overline{k}}) \le \overline{f} + \varepsilon$ (если на некотором шаге $f(x_k) < \overline{f}$, то итерации прекращаются); если же $\overline{f} < f^*$, то последовательность $\{h_k\}$ является неограниченной.

Теорема 2.15 позволяет построить алгоритм минимизации функции $f(x)$, удовлетворяющей условию этой теоремы, при неизвестном f^*. Этот алгоритм будет состоять из последовательности этапов, на каждом из которых будет применяться ОГСРП в форме, указанной в теореме 2.15.

Вначале выбираем $x_0^1 \in E^n,\ h > 0,\ \Delta_0 > 0,\ \tilde{f}_0 = f(x_0^1)$. Пусть выполнены r этапов алгоритма. Перед $(r+1)$-м этапом имеем

$$x_0^{r+1} \in E^n,\ h > 0,\ \Delta_r > 0,\ \tilde{f}_r = f(x_0^{r+1}),$$

где x_0^{r+1} – точка, в которой получено рекордное (т.е. наименьшее) значение функционала после r этапов, $\tilde{f}_r$ – значение этого рекорда. Пусть

$$\Delta_{r+1} = \begin{cases} \Delta_r, & \text{если } \ f(x_0^{r+1}) \le f(x_0^r) - \Delta_r/2, \\ \dfrac{\Delta_r}{2}, & \text{если } \ f(x_0^{r+1}) > f(x_0^r) - \Delta_r/2 \end{cases}$$

и $\overline{f}_{r+1} = f(x_0^{r+1}) - \Delta_{r+1}$.

Приняв $x_0 = x_0^{r+1}$ и $\overline{f} = \overline{f}_{r+1}$, применяем алгоритм ОГСРП в форме, описанной в теореме 2.15. На некотором шаге алгоритма $k = \overline{k}_{r+1}$ возможен один из двух случаев:

а) $f(x_{\overline{k}_{r+1}}) \leq f(x_0^{r+1}) - \Delta_{r+1}/2$;

б) $h_{\overline{k}_{r+1}+1} > h$.

На этом $(r+1)$-й этап прекращается, запоминаются "рекордное" значение $\overline{f}_{r+1} = \min\limits_{0 \leq k \leq \overline{k}_{r+1}} f(x_k^{r+1})$ и соответствующая точка $x_{k^*_{r+1}}^{r+1} = x_0^{r+2}$.

Перейдем к $(r+2)$-му этапу.

Покажем, что $\lim\limits_{r \to \infty} f(x_0^r) = f^*$. Так как последовательность $\{f(x_0^r)\}$ является невозрастающей и ограниченной снизу, то $\lim\limits_{r \to \infty} f(x_0^r)$ существует. Из описания алгоритма и теоремы 2.15 ясно, что при достаточно малом Δ_r случай б) может иметь место только тогда, когда $\overline{f}_{r+1} < f^*$. С другой стороны, случай а) может иметь место после случая б) только конечное число раз подряд. Следовательно, $\lim\limits_{r \to +\infty} \Delta_r = 0$. Начиная с достаточно малых Δ_r (при фиксированном h), если имеет место случай б), то $f(x_0^r) - \Delta_r$ служит оценкой снизу для f^*; таким образом, имеем

$$\lim_{r \to \infty} f(x_0^r) = f^*.$$

При определенных условиях, например при решении систем нелинейных уравнений, f^* известно, но могут возникнуть трудности с оценкой M и N. При неправильном выборе констант M и N сходимость (M, N)-алгоритма ОГСРП, описанного в теореме 2.12, может оказаться слишком медленной или вообще отсутствовать. Поэтому важно иметь признаки, с помощью которых можно получить информацию о "ненормальном" ходе процесса минимизации. Наиболее простой из этих признаков основан на изучении поведения последовательности $\{h_k\}$.

Теорема 2.16. *Если параметры M, N в (M, N)-алгоритме выбраны правильно, то последовательность $\{h_k\}$ ограничена. Если при неправильном выборе параметров M и N отсутствует сходимость, то одна из последовательностей $\{h_k\}$ или $\{f(x_k)\}$ неограничена.*

Таким образом, если в процессе минимизации по схеме ОГСРП, описанной в теореме 2.12, при заданных M и N величины h_k либо $f(x_k)$ превзошли достаточно большие заданные числа $h^{\max}$ и $f^{\max}$, то следует константу M увеличить, а N уменьшить и начать процесс заново из начальной или наилучшей достигнутой точки.

2.4. Метод ОГСРП в направлении разности двух последовательных субградиентов

Среди методов недифференцируемой оптимизации особое положение по своей практической эффективности занимают методы с растяжением пространства в направлении разности двух последовательных субградиентов (или так называемые r-алгоритмы) [731]. По своей структуре они очень близки к методам ОГСРП, но между ними есть важное различие: сходящиеся к минимуму методы ОГСРП по значению минимизируемой функции в принципе не могут быть монотонными, в то же время r-алгоритмы при определенной регулировке шаговых множителей могут стать монотонными. Это связано с простым геометрическим фактом: если мы находимся на границе двух "кусков" кусочно гладкой поверхности, причем градиенты к этим кускам, вычисленные в данной точке, образуют между собой тупой угол, то любое растяжение пространства в направлении одного из градиентов (или последовательное растяжение попеременно в направлении двух указанных градиентов) не может превратить этот угол в острый, он может лишь приближаться к $\pi/2$, оставаясь тупым. В то же время растяжение пространства в направлении разности двух указанных градиентов с достаточно большим коэффициентом растяжения превращает тупой угол между градиентами в острый.

Рассмотрим теперь общую схему r-алгоритмов при минимизации почти дифференцируемой функции $f(x)$, определенной на E^n. Будем предполагать, что

$$\lim_{||x||\to\infty} f(x) = +\infty.$$

Выбираем начальное приближение $x_0 \in E^n$ и неособенную матрицу B_0 (чаще всего B_0 совпадает с единичной матрицей I_n или с диагональной матрицей D_n с положительными элементами на диагонали, с помощью которой осуществляется масштабирование переменных). Первый шаг алгоритма производим по формуле

$$x_1 = x_0 - h_0 \eta_0,$$

где $\eta_0 = B_0 B_0^* g_f(x_0)$, B^* – сопряженная матрица, $h_0 \geq 0$ – некоторый шаговый множитель, выбираемый из условия существования в точке x_1 почти-градиента $g_f(x_1)$ такого, что

$$(g_f(x_1),\, \eta_0) \leq 0.$$

При $B_0 = I_n$ имеем $\eta_0 = g_f(x_0)$, и первый шаг совпадает с итерацией субградиентного процесса.

Пусть в результате вычислений после k ($k = 1, 2, \dots$) шагов процесса получены определенные значения $x_k \in E^n$ и матрицы B_k размера $n \times n$. Опишем $(k+1)$-й шаг процесса, который состоит из вычисления ряда величин.

1) Находим значение градиента $g_f(x_k)$ (почти-градиент) функции $f(x)$ в точке x_k.

2) Находим разность двух последовательных градиентов $\Delta_k = g_f(x_k) - g_f(x_{k-1})$.

3) Находим $r_k = B_k^*(g_f(x_k) - g_f(x_{k-1}))$ – значение разности двух последовательных градиентов в преобразованном пространстве.

Переход от начального пространства к преобразованному задается формулой $y = A_k x$, где $A_k = B_k^{-1}$. Определим функцию $\varphi_k(y) = f(B_k y)$; тогда $g_{\varphi_k}(y) = B_k^* g_f(x)$. Таким образом, r_k есть разность двух почти-градиентов от функции $\varphi_k(y)$, вычисленных в точках $y_k = A_k x_k$ и $\overline{y}_k = A_k x_{k-1}$.

4) Находим вектор растяжения пространства $\xi_k = r_k / ||r_k||$.

5) Зададим величину β_k, обратную коэффициенту растяжения α_k пространства перед $(k+1)$-м шагом.

6) Вычислим $B_{k+1} = B_k R_{\beta_k}(\xi_k)$; заметим, что $B_{k+1} = A_{k+1}^{-1}$, где $A_{k+1} = R_{\alpha_k}(\xi_k) A_k$ – оператор растяжения пространства перед $(k+1)$-м шагом.

7) Находим значение градиента в преобразованном пространстве $\tilde{g}_k = B_{k+1}^* g_f(x_k)$ (почти-градиент) функции $\varphi_{k+1}(y) = f(B_{k+1} y)$, взятый в точке $\overline{y}_{k+1} = A_{k+1} x_k$.

8) Осуществляем переход в новую точку

$$x_{k+1} = x_k - h_k B_{k+1} \tilde{g}_k / ||\tilde{g}_k||, \tag{2.22}$$

который представляет собой "рабочий" шаг алгоритма, соответствующий шагу обобщенного градиентного спуска в растянутом под воздействием оператора A_{k+1} пространстве. Применив к обеим частям формулы (2.22) оператор A_{k+1}, получим

$$y_{k+1} = A_{k+1} x_{k+1} = \overline{y}_k - h_k \tilde{g}_k / ||\tilde{g}_k||,$$

где $\overline{y}_k = A_{k+1} x_k$.

9) Переход к следующему шагу или окончание работы алгоритма.

Выбор шагового множителя h_k – существенная часть алгоритма, от которого во многом зависит его практическая эффективность.

В r-алгоритме выбор h_k осуществляется из условия приближенного поиска минимума по направлению, при этом при минимизации выпуклых функций должно соблюдаться условие $h_k \geq h_k^*$ (h_k^* – значение шагового множителя, соответствующего минимуму по направлению), в общем же случае нужно следить за тем, чтобы направление обобщенного градиента в точке x_{k+1} образовало нетупой угол с направлением спуска из точки x_k.

При минимизации негладких выпуклых функций, определенных на E^n, наиболее удачными оказались следующие варианты алгоритма.

1. *Адаптивный способ регулировки шагового множителя*, когда коэффициенты растяжения пространства α_k выбираются в пределах 2–4 для шагового множителя h_k. Задается некоторое натуральное число $\overline{m}$, константы $q > 1$ и $t_0^0 > 0$. После k шагов получаем константу t_k^0. Двигаемся из точки x_k в направлении спуска с шагом, равным t_k^0, до тех пор, пока не будет выполнено условие завершения спуска по направлению, либо число шагов не станет равным $\overline{m}$. Условие завершения спуска может состоять в том, что значение функции в очередной точке не меньше, чем значение функции в предыдущей точке; другой вариант такого условия – производная по направлению спуска в данной точке неотрицательна. Если прошло $\overline{m}$ шагов, а условие завершения спуска не выполнено, то вместо t_k^0 запоминаем $t_k^1 = qt_k^0$, где $q > 1$, и продолжаем спуск в том же направлении с большим шагом. Если после очередных $\overline{m}$ шагов условие спуска не выполнено, то вместо t_k^1 берем $t_k^2 = qt_k^1$ и т.д. Так как мы предполагаем, что $\lim\limits_{||x|| \to \infty} f(x) = +\infty$, то после конечного числа шагов в определенном направлении обязательно выполнится условие завершения спуска. Константа шага $t_k^{p_k} = q_{p_k} t_k^0$ ($p \in \{0, 1, 2, \dots\}$), которая использовалась на последнем шаге, принимается в качестве начальной при спуске в новом направлении из точки x_{k+1}, полученной при завершении спуска, т.е. $t_{k+1}^0 = t_k^{p_k}$.

Может возникнуть вопрос: шаговый множитель не убывает – как же тогда получается сходящийся процесс? Сходимость r-алгоритма связана с тем, что, как правило, последовательность матриц B_k стремится к нулевой матрице при $k \to \infty$. Лишь в исключительных случаях процесс не сходится:

- это когда траектория спуска, начиная с определенного момента, лежит в некотором линейном многообразии размерности, меньшей n;

- либо когда последовательность векторов $\{x_{k+1} - x_k\}$ образует с некоторым подпространством углы, стремящиеся к 0.

В таких случаях матрица B_k может не стремиться к нулевой матрице, но при этом $B_k \tilde{g}_k / ||\tilde{g}_k||$ стремится к 0.

2. *Идеализированный вариант r-алгоритма*, когда величина шага h_k выбирается из условия минимума по направлению. Если полученное направление $B_k\tilde{g}_k$ не является направлением спуска, то $h_k = 0$, $x_{k+1} = x_k$ и в качестве $g_f(x_{k+1})$ выбирается обобщенный градиент, образующий с $B_k\tilde{g}_k$ неострый угол.

Справедлива следующая теорема.

Теорема 2.17. *Пусть $f(x)$ – почти дифференцируемая кусочно-гладкая функция такая, что $\lim\limits_{||x||\to\infty} f(x) = +\infty$, x_0 – заданное начальное приближение, $\alpha_k = \alpha > 1$ $(k = 0, 1, 2, \dots)$ и в r-алгоритме шаговый множитель выбирается из условия минимума по направлению. Тогда, если последовательность $\{x_k\}_{k=0}^{\infty}$, полученная в процессе реализации r-алгоритма, удовлетворяет условию $\lim\limits_{k\to\infty} ||x_{k-1} - x_k|| = 0$, в множестве $U = \{x\colon f(x) = f_\infty\}$ найдется точка x^* такая, что множество векторов $\tilde{G}_f(x^*)$ будет линейно зависимым (здесь $f_\infty = \lim\limits_{k\to\infty} f(x_k)$; $\tilde{G}_f(x^*)$ – множество почти-градиентов в точке x^*).*

3. *Предельный вариант r-алгоритма*, когда $\beta_k = 0$, $\alpha_k = +\infty$, а h_k $(k = 1, 2, \dots)$ находится из условия минимума по направлению.

Пусть $f(x) = (Ax, x)/2$ – положительно определенная квадратичная функция и задано начальное приближение x_0, для которого $g(x_0) = Ax_0$. Далее $x_1 = x_0 - h_0 g(x_0)$, где $h_0 = arg\min\limits_h f(x_0 - h \cdot g(x_0))$, т.е. $h_0 = \dfrac{(Ax_0, x_0)}{(A^2x_0, x_0)}$. Тогда имеем

$$x_1 = x_0 - \frac{(Ax_0,\, x_0)}{(A^2x_0,\, x_0)} Ax_0.$$

Для новой точки x_1 находим

$$g(x_1) = Ax_0 - \frac{(A^2x_0,\, x_0)}{(A^2x_0,\, Ax_0)} A^2x_0.$$

Рассмотрим разность

$$\Delta_1 = g(x_1) - g(x_0) = -\frac{(A^2x_0,\, x_0)}{(A^2x_0,\, Ax_0)} A^2x_0,$$

для которой имеем

$$(x_1,\, g(x_1) - g(x_0)) = (Ax_1, x_1 - x_0) = (g(x_1), x_1 - x_0) = 0.$$

Таким образом, разность градиентов, взятых в точках x_1 и x_0, оказывается ортогональной вектору x_1. Так как $-x_1$ – направление на точку минимума, то разность двух последовательных градиентов оказывается ортогональной направлению на точку минимума. Растяжению пространства в направлении $\Delta_1 = g(x_1) - g(x_0)$ с $\alpha = +\infty$ будет соответствовать операция проектирования градиента $g(x_1)$ на подпространство, ортогональное Δ_1. Дальнейший спуск из точки x_1 в соответствии с r-алгоритмом будет проходить в линейном многообразии, проходящем через x_1 ортогонально Δ_1. Это многообразие содержит начало координат – точку минимума $f(x)$. По индукции легко показать, что спуск из точки x_k $(k = 2, 3, \dots)$ будет проходить в многообразии, ортогональном векторам $\Delta_1, \Delta_2, \dots, \Delta_k$, проходящим через начало координат. Таким образом, для квадратичной положительно определенной функции предельный вариант r-алгоритма сходится к точке минимума не более чем за n шагов. Его можно рассматривать как проективный вариант метода сопряженных градиентов.

Предельный вариант r-алгоритма с восстановлением матрицы B_k (заменой ее единичной матрицей) после каждых n шагов применяется к задаче минимизации выпуклой дважды непрерывно дифференцируемой функции. Доказан следующий результат [715].

Теорема 2.18. *Пусть выпуклая функция $f(x)$, определенная в E^n, дважды непрерывно дифференцируема в некоторой окрестности S точки минимума x^*, причем в этой окрестности матрица вторых производных (гессиан) $H(x)$ удовлетворяет условию Липшица*

$$||H(x) - H(x')|| \leq L||x - x'||, \; x, x' \in S,$$

а в точке x^* – положительно определена. Тогда найдется такая окрестность $S' \subset S$ точки x^*, что если $x_0 \in S'$, то существует число $c > 0$, для которого

$$||x_n - x^*|| \leq c||x_0 - x^*||^2;$$

здесь x_n – точка, полученная после n шагов предельного варианта r-алгоритма ($\beta_k = 0$, h_k находятся из условия минимума по направлению $(k = \overline{0, n-1})$).

Таким образом, предельный вариант r-алгоритма, реализующий, по сути, метод проективных сопряженных градиентов, обладает по отношению к большим циклам, состоящим из n шагов r-алгоритма, квадратичной скоростью сходимости при обычных условиях гладкости и регулярности.

Рассмотрим некоторые особенности применения r-алгоритма при решении задач безусловной оптимизации в схемах декомпозиции и методе негладких штрафных функций. Как показывают практические расчеты r-алгоритм успешно используется для решения не только задач безусловной минимизации, но и задач выпуклого программирования. Пусть $f_0(x), f_1(x), \ldots, f_m(x)$ – выпуклые функции, определенные на E^n. Поставим задачу нахождения

$$\min\{f_0(x)\colon\ f_i(x) \leq 0 \quad (i = \overline{1,m})\}.$$

Для ее решения с использованием r-алгоритма наиболее часто применяется метод негладких штрафных функций. В качестве штрафной компоненты можно применить функции максимума невязок, сумму модулей невязок, корень квадратный из суммы квадратов невязок и другие точные негладкие штрафные функции, сохраняющие выпуклость. В том случае, когда субградиенты ограничений сложно вычисляются и этих ограничений много, более предпочтительна штрафная компонента в форме функции максимума, так как при вычислении субградиента такой штрафной функции требуется вычислять субградиент не более одной из функций, выражающих ограничения. Определенные трудности при использовании штрафных негладких функций может вызвать подбор штрафных коэффициентов, связанный с оценкой множителей Лагранжа. Этот вопрос обычно решается с учетом специфики решаемой задачи оптимизации.

Довольно часто при использовании схем декомпозиции по ограничениям, при вычислении двойственных оценок возникают задачи минимизации выпуклой негладкой функции $f(x)$ при простейших ограничениях $x \geq 0$. Одним из приемов решения такого рода задач является "четное" продолжение целевой функции на все пространство путем замены переменных $x_i = |y_i|$. Получаем многоэкстремальную функцию $F(y) = f(|y_1|, \ldots, |y_n|)$, однако все ее локальные минимумы глобальны. Как показала практика многочисленных расчетов с использованием r-алгоритма для функций типа $F(y)$, такой подход дает надежные результаты и в определенном смысле лучше метода штрафных функций, поскольку не требует подбора штрафных параметров.

При решении задач большой размерности вида $\min f(x)$ при условии $x \geq 0$, когда специфика задачи позволяет надеяться, что большая часть компонент оптимального решения равна нулю, желательно проводить процесс, подобный r-алгоритму, в подпространстве, соответствующем ненулевым компонентам оптимального решения, размерность которого n' может быть гораздо меньше n. Если бы удалось выделить это подпространство, то для достижения определенной точности решения пришлось бы затратить в n/n' меньше итераций. В нетривиальных случаях, конечно, невозможно заранее выделить нулевые компоненты оптимального решения. Поэтому предложена модификация r-алгоритма, в которой по определенным правилам в процессе счета происходит временная фиксация нулевых значений тех или иных координат (или их "освобождение" от этой фиксации), поэтому в определенный момент минимизация происходит по некоторому подмножеству координат.

2.5. Метод эллипсоидов

Большое теоретическое и практическое значения имеет, так называемый, метод эллипсоидов [713, 742], который можно рассматривать:

- с одной стороны, как частный случай метода ОГСРП с постоянными коэффициентами растяжения пространства и шаговым множителем, меняющимся по формуле геометрической прогрессии;

- с другой стороны, этот метод представляет собой метод последовательных отсечений, в котором область локализации решения аппроксимируется на каждом шаге эллипсоидом (см. теорему 2.12).

Предложенный первоначально для решения задач выпуклого программирования метод эллипсоидов легко обобщается на более широкий класс задач нахождения аналогов неподвижных точек векторных полей специального вида.

Пусть на E^n задано векторное поле $g(x)$, не обязательно непрерывное, $g(x) \in E^n$ $(x \in E^n)$. Рассмотрим задачу: найти такую точку x^*, что $(g(x),\ x - x^*) \geq 0$ при всех $x \in E^n$. Допустим, что эта задача имеет решение, причем известно, что $x^* \in S(x_0, R)$, где $S(x_0, R)$ – замкнутый шар с центром в точке x_0 и радиусом R.

Рассмотрим следующий итеративный алгоритм решения указанной задачи при $n > 1$. При описании этого алгоритма будем предполагать, что $g(x) \neq 0$ при $x \neq x^*$.

Перед началом вычислений имеем: $x_0 \in E^n$, $B_0 = I_n$ – единичная матрица, $h_0 = R/(n{+}1)$. Пусть на k-м шаге мы получили: $x_k \in E^n$, B_k – матрицу $n \times n$, B_k^* – сопряженная матрица, $h_k > 0$. На $(k+1)$-м шаге вычисляем:

1) значение вектора $g(x_k)$; если $g(x_k) = 0$, то x_k – искомая точка;

2) вектор направления преобразования пространства

$$\xi_k = \frac{B_k^*\, g(x_k)}{||B_k^*\, g(x_k)||}; \tag{2.23}$$

3) значение новой точки

$$x_{k+1} = x_k - h_k B_k \xi_k; \tag{2.24}$$

4) матрицу преобразования пространства

$$B_{k+1} = B_k R_\beta(\xi_k), \;\; \beta = \sqrt{(n-1)/(n+1)}, \tag{2.25}$$

где $R_\beta(\xi_k)$ – оператор растяжения пространства в направлении ξ_k с коэффициентом β;

5) значение шагового множителя

$$h_{k+1} = h_k r, \;\;\; r = n/\sqrt{n^2-1}. \tag{2.26}$$

Теорема 2.19. *Последовательность* $\{x_k\}_{k=0}^{\infty}$, *генерируемая алгоритмом* (2.23)–(2.26), *удовлетворяет неравенству*

$$||A_k(x_k - x^*)|| \le h_k(n+1), \;\; k = 0,\, 1,\, 2, \ldots,$$

где $A_k = B_k^{-1}$.

Множество точек x, удовлетворяющих неравенству

$$||A_k(x_k - x)|| \le (n+1)h_k = R \cdot \left(n/\sqrt{n^2-1}\,\right)^k,$$

представляет собой эллипсоид Φ_k, объем которого $v(\Phi_k) = \dfrac{v_0 R^n (n/\sqrt{n^2-1})^{nk}}{\det A_k}$, где v_0 – объем единичного n-мерного шара. Получаем оценку

$$\frac{v(\Phi_{k+1})}{v(\Phi_k)} = \frac{(n/\sqrt{n^2-1})^n \det A_k}{\det A_{k+1}} = \frac{(n/\sqrt{n^2-1})^n \det A_k}{\det R_\alpha(\xi_k) \det A_k} =$$

$$= \frac{1}{\alpha}\left(\frac{n}{\sqrt{n^2-1}}\right)^n = \sqrt{\frac{n-1}{n+1}}\left(\frac{n}{\sqrt{n^2-1}}\right)^n = q_n < 1. \tag{2.27}$$

Таким образом, объем эллипсоида, в котором локализуется искомая точка, в соответствии с неравенством (2.27) убывает со скоростью геометрической прогрессии со знаменателем q_n.

Рассмотрим возможность применения описанной процедуры для решения ряда задач математического программирования.

1. Задача минимизации выпуклой функции на шаре. Пусть на E^n определена выпуклая функция $f(x)$. Требуется найти минимальное значение этой функции на шаре $S(x_0, R) = \{x\colon ||x - x_0|| \le R\}$.

Пусть $x^* \in S(x_0, R)$ – искомый минимум функции $f(x)$ на шаре $S(x_0, R)$. Тогда для всех $x \in S(x_0, R)$

$$(g_f(x),\, x - x^*) \ge f(x) - f(x^*) \ge 0.$$

Для $x \notin S(x_0, R)$

$$(x - x_0,\, x - x^*) = ||x - x_0||^2 - (x - x_0,\, x^* - x_0) \ge$$

$$\ge ||x - x_0||(||x - x_0|| - ||x^* - x_0||) \ge 0.$$

Таким образом, можно построить векторное поле $g(x)$, удовлетворяющее свойству $(g(x), x - x_0) \ge 0$ $(x \in E^n)$ следующим образом:

$$g(x) = \begin{cases} g_f(x), & x \in S(x_0, R), \\ (x - x_0)/||x - x_0||, & x \notin S(x_0, R). \end{cases}$$

Применим для решения нашей задачи алгоритм (2.23)–(2.26), определив $g(x)$ указанным образом. Тогда в соответствии с формулой (2.27) получим, что объем области, в которой локализован минимум, убывает со скоростью геометрической прогрессии со знаменателем

$$q_n = \sqrt{\frac{n-1}{n+1}} \left(\frac{n}{\sqrt{n^2 - 1}} \right)^n.$$

Рассмотрим случай, когда минимум функции $f(x)$ достигается в точке x^*, являющейся внутренней точкой шара $S(x_0, R)$. Так как алгоритм (2.23)–(2.26) применительно к задаче минимизации функции $f(x)$ фактически реализует метод ОГСРП с небольшими изменениями, то для оценки скорости сходимости по функционалу можно применить теорему 2.11. Получаем следующую оценку:

$$\rho_k = \min_{1 \le i \le k} ||B_i^* g(x_i)|| \le \frac{d_k \sqrt{k(\alpha^2 - 1)}}{\sqrt{\alpha^{2k/n} - 1}}, \tag{2.28}$$

где

$$d_k = \max_{1 \le i \le k} ||g(x_i)||.$$

Пусть минимум в (2.28) достигается при $i = i(k)$. Тогда

$$\mu_k = (x_{i(k)} - x^*,\; g(x_{i(k)})) = (A_{i(k)}(x_{i(k)} - x^*),\; B_{i(k)}^* g(x_{i(k)})) \le$$

$$\le ||A_{i(k)}(x_{i(k)} - x^*)|| \frac{d_k \sqrt{k(\alpha^2 - 1)}}{\sqrt{\alpha^{2k/n} - 1}} \le h_{i(k)}(n+1) \frac{d_k \sqrt{k(\alpha^2 - 1)}}{\sqrt{\alpha^{2k/n} - 1}} \le$$

$$\le h_k(n+1) d_k \frac{\sqrt{k(\alpha^2 - 1)}}{\sqrt{\alpha^{2k/n} - 1}} = \frac{R d_k \sqrt{k(\alpha^2 - 1)}}{\sqrt{1 - \beta^{2k/n}}} q_n^{k/n}.$$

Заметим, что если $x_{i(k)} \notin S(x_0, R)$, то $g(x_{i(k)}) = \dfrac{x - x_0}{||x - x_0||}$ и, так как x^* лежит внутри $S(x_0, R)$,

$$\left(x_{i(k)} - x^*,\ \frac{x_{i(k)} - x_0}{||x_{i(k)} - x_0||}\right) \geq \delta > 0.$$

Таким образом, из того, что $\mu_k \to 0$, следует, что, начиная с достаточно больших k, $x_{i(k)} \in S(x_0, R)$. Но если $x_{i(k)} \in S(x_0, R)$, то $g(x_{i(k)}) = g_f(x_{i(k)})$ и справедливо неравенство

$$f(x_{i(k)}) - f(x^*) \leq (g_f(x_{i(k)}), x_{i(k)} - x^*) \leq \frac{Rd_k\sqrt{k(\alpha^2 - 1)}}{\sqrt{1 - \beta^{2k/n}}} q_n^{k/n}.$$

Таким образом, если минимум функции $f(x)$ достигается в точке x^*, являющейся внутренней точкой шара $S(x_0, R)$, то при использовании алгоритма вида (2.23)–(2.26) отклонение "рекорда" по функционалу от $f(x^*)$ удовлетворяет неравенству

$$r_k = \min_{1 \leq i \leq k} \Big[f(x_i) - f(x^*)\Big] \leq \frac{Rd_k\sqrt{k(\alpha^2 - 1)}}{\sqrt{1 - \beta^{2k/n}}} q_n^{k/n}.$$

Если при всех $k = 0,\ 1,\ 2, \ldots\ x_k \in S(x_0, R)$, то можно получить более точную оценку

$$f(x_{k_i}) - f(x^*)) \leq c q_n^{k_i/n}$$

для некоторой подпоследовательности $\{k_i\}_{i=1}^{\infty}$, используя теорему 2.19. При больших n величина q_n хорошо приближается следующей асимптотической формулой $q_n \approx 1 - 1/(2n)$. Отсюда следует, что, хотя r_k сходится примерно со скоростью геометрической прогрессии (если пренебречь медленно растущим множителем $\sqrt{k}$), при больших n знаменатель этой прогрессии близок к единице, т.е. практически скорость сходимости может оказаться медленной.

Положительным свойством алгоритма (2.23)–(2.26) является то, что гарантированная скорость сходимости зависит лишь от размерности пространства и не требует знания специфических особенностей функции $f(x)$.

2. Общая задача выпуклого программирования. Пусть требуется найти

$$\min f_0(x) \tag{2.29}$$

при ограничениях

$$f_i(x) \leq 0,\ \ i = 1,\ 2, \ldots, m;\ \ x \in E^n, \tag{2.30}$$

где $f_\nu(x)$ – выпуклые функции, определенные на E^n, $g_\nu(x)$ – соответствующие субградиенты ($\nu = 0, 1, \ldots, m$), причем имеется априорная информация, что оптимальная точка x^* существует и находится в шаре $S(x_0, R)$ (формально к системе ограничений (2.30) можно добавить ограничение $||x - x_0|| \le R$).

Рассмотрим поле $g(x)$, построенное следующим образом:

$$g(x) = \begin{cases} g_0(x), & \text{если} \quad \max\limits_{1 \le i \le m} f_i(x) \le 0, \\ g_{i^*}(x), & \text{если} \quad \max\limits_{1 \le i \le m} f_i(x) = f_{i^*}(x) > 0. \end{cases} \tag{2.31}$$

Покажем, что $(g(x), x - x^*) \ge 0$ при всех $x \in E^n$. Если $\max\limits_{1 \le i \le m} f_i(x) \le 0$, то $g(x) = g_0(x)$ и

$$(g(x), x - x^*) = (g_0(x), x - x^*) \ge f_0(x) - f_0(x^*) \ge 0.$$

Если $\max\limits_{1 \le i \le m} f_i(x) > 0$, то $g(x) = g_{i^*}(x)$, причем $f_{i^*}(x) > 0$, и $f_{i^*}(x^*) \le 0$. Тогда

$$(g(x), x - x^*) = (g_{i^*}(x), x - x^*) \ge f_{i^*}(x) - f_{i^*}(x^*) \ge 0.$$

Таким образом, $(g(x), x - x^*) \ge 0$ при всех $x \in E^n$. Используя это неравенство, применим для локализации точки x^* алгоритм метода эллипсоидов (2.23)–(2.26), вычисляя $g(x)$ по формуле (2.31).

Из теоремы 2.19 вытекает, что после k шагов алгоритма (2.23)–(2.26) оптимальное решение x^* будет локализовано в эллипсоиде Φ_k с центром в точке x_k объемом $v(\Phi_k) = v_0 q_n^k$, где v_0 – объем шара $S(x_0, R)$.

Отметим, что этот результат не изменится, если в формуле (2.31) вместо $g_{i^*}(x)$ брать $g_{\bar{i}}(x)$, где $\bar{i}$ – произвольный индекс, для которого $f_{\bar{i}}(x) > 0$.

3. Задача о седловой точке. Пусть задана выпукло-вогнутая функция $f(z) = f(x, y)$ двух векторных переменных $x \in E^n$, $y \in E^m$, $z = \{x, y\} \in E^n \times E^m \equiv E^{n+m}$, z^* – седловая точка этой функции, z_0 – заданное начальное приближение и априори известно, что $||z_0 - z^*|| \le R$.

Рассмотрим псевдоградиентное множество

$$G(z) = G_f^x(x, y) \times (-G_f^y(x, y)),$$

где $G_f^x(x, y)$ – множество частных субградиентов функции $f(x, y)$, рассматриваемой как функция от x при фиксированном y; $-G_f^y(x, y)$ — множество субградиентов от функций $-f(x, y)$ по y при фиксированном x.

Сформируем векторное поле $g(z)$ следующим образом:

$$g(z) = \{g^x_f(z),\ -g^y_f(z)\};\ ,\ g^x_f(z) \in G^x_f(z);\ \ g^y_f(z) \in G^y_f(z).$$

Покажем что $(g(z),\, z - z^*) \geq 0$.

Из определения седловой точки следует, что $f(x, y^*) \geq f(x^*, y^*) \geq$ $\geq f(x^*, y)$. Далее

$$0 \leq f(x, y^*) - f(x^*, y) = f(x, y^*) - f(x, y) + f(x, y) - f(x^*, y) \leq$$

$$\leq (g^x_f(z),\, x - x^*) - (g^y_f(z),\, y - y^*) = (g(z),\, z - z^*).$$

Таким образом, для локализации седловой точки z^* можем применить алгоритм (2.23)–(2.26) используя в нем псевдоградиент $\{g^x_f(z),\ -g^y_f(z)\}$ вместо $g(z)$.

4. Полиномиальные алгоритмы решения задач линейного программирования. Метод эллипсоидов сыграл конструктивную роль при построении первого полиномиального алгоритма для решения задач линейного программирования с целыми (рациональными) коэффициентами [705, 706]. Данный алгоритм имеет больше теоретическую значимость, чем практическое применение.

Пусть задана задача линейного программирования:

$$\sum_{j=1}^{n} c_j x_j \to \min, \tag{2.32}$$

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad (i = \overline{1, m}), \tag{2.33}$$

$$x_j \geq 0 \quad (j = \overline{1, n}), \tag{2.34}$$

в которой все коэффициенты c_j, b_i, a_{ij} являются целыми (рациональными) числами. Как известно, задачу линейного программирования можно свести к решению системы линейных неравенств, например, путем совместного рассмотрения прямой и двойственной задач.

Для задачи (2.32)–(2.34) построим ее двойственную:

$$\sum_{i=1}^{m} b_i u_i \to \max, \tag{2.35}$$

$$\sum_{i=1}^{m} a_{ij} u_i \geq c_j \quad (j = \overline{1, n}), \tag{2.36}$$

$$u_i \geq 0 \quad (i = \overline{1, m}). \tag{2.37}$$

По теореме двойственности, чтобы пара векторов (x^*, u^*) была решением прямой и двойственной задач (2.32)–(2.34) и (2.35)–(2.37), необходимо и достаточно выполнения следующих условий:

$$\begin{aligned}
&\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i u_i^*,\\
&\sum_{j=1}^{n} a_{ij} x_j^* \le b_i \quad (i = \overline{1,m}),\\
&\sum_{i=1}^{m} a_{ij} u_i^* \ge c_j \quad (j = \overline{1,n}),\\
&x_j^* \ge 0 \quad (j = \overline{1,n}),\\
&u_i^* \ge 0 \quad (i = \overline{1,m}).
\end{aligned} \tag{2.38}$$

Таким образом, решение задач (2.32)–(2.34) и (2.35)–(2.37) сведено к решению системы неравенств (2.38), т.е. любая задача линейного программирования может быть сведена к решению системы линейных неравенств.

Рассмотрим систему линейных неравенств с целыми коэффициентами вида

$$Ax \le b, \quad \text{или} \quad \sum_{j=1}^{n} a_{ij} x_j - b_i \le 0 \quad (i = \overline{1,m}), \tag{2.39}$$

где $A = \{a_{ij}\}$ – матрица $m \times n$ с целыми элементами a_{ij}; b – m-мерный вектор. Легко показать, что если система (2.39) имеет решение, то она имеет и "опорное" решение, т.е. имеется решение невырожденной системы линейных уравнений, матрица которой получается путем отбрасывания определенных строк и столбцов матрицы A, а правая часть в оставшихся строках сохраняется без изменения (т.е. отбрасываются несущественные ограничения и некоторые компоненты – свободные переменные – вектора x приравниваются к нулю).

Для задачи (2.39) определяется величина

$$L = \left\lfloor \sum_{i=1}^{m}\sum_{j=1}^{n} \log_2(|a_{ij}|+1) + \sum_{i=1}^{m} \log_2(|b_i|+1) + \log_2 m \cdot n \right\rfloor + 1,$$

которая характеризует длину входного кода при некоторой системе кодирования исходных параметров задачи на идеализированной ЭВМ, например машине Тьюринга. Тогда алгоритм решения задачи линейного программирования будет считаться полиномиальным, если время решения на таком компьютере будет ограничено выражением от величин L, m и n, т.е. потребуется арифметических операций порядка $O(m, n, L)$.

Ненулевые компоненты опорного решения могут быть записаны по теореме Крамера в виде отношения

$$x_j = det\,\Delta_j / det\,\Delta, \quad j \in \{1, \ldots, n\},$$

где Δ_j – j-й минор расширенной матрицы $[A, b]$, Δ – минор матрица A. Оценим $|\Delta_j|$ и, учитывая, что $|\Delta| \geq 1$, получим следующую оценку сверху для радиуса сферы с центром в начале координат, внутри которого находится решение системы (если оно существует) $R \leq 2^L$.

Рассмотрим функцию

$$\varphi(x) = \max_i \left(\sum_{j=1}^{n} a_{ij} x_j - b_i \right),$$

и пусть $\varphi^* = \min\limits_{x \in R^n} \varphi(x)$. Если $\varphi^* \leq 0$, то система (2.39) имеет решение, если же $\varphi^* > 0$, то она не имеет решения. Оказывается, что если задача (2.39) не имеет решения, то существует "зазор" – число $\alpha = \alpha(L) > 0$ такое, что выполняется неравенство $\varphi^* > \alpha$. Этот "зазор" позволяет применить для решения системы приближенный алгоритм: достаточно найти минимум $\varphi(x)$ с точностью до $\alpha/2$ по функционалу. Если при этом найденное значение функционала окажется больше $\alpha/2$, то система (2.39) противоречива. Так как необходимая точность решения задачи оценивается числом 2^{-L}, экспоненциально зависящим от длины кода задачи, то для получения полиномиального числа итераций необходим алгоритм, сходящийся со скоростью геометрической прогрессии, знаменатель которой не зависит от исходной информации. Таким условиям и удовлетворяет *метод эллипсоидов.*

Зная длину кода системы линейных неравенств (2.39), используя оценки $R \leq 2^L$ и $\alpha \geq 2 \cdot 2^{-L}$ и применяя теорему 2.19, легко получим число итераций N, которое гарантирует решение системы (2.39) с точностью до $\alpha/2$: $N \leq 6n^2L$. При этом общее число арифметических операций не превышает операций порядка $O(mn^3L)$. Эти оценки получены для точной арифметики, т.е. когда вычисления производятся с неограниченным числом разрядов.

Для "реальной" арифметики, когда вычисления проводятся с ограниченным числом разрядов (порядка длины кода L задачи), полученные оценки лишь слегка изменяются, но порядок их остается прежним: $O(mn^3L)$ при $n \leq m$.

Допустим, что, используя метод эллипсоидов, мы показали, что система вида (2.39) совместна. Теперь уже несложно построить полиномиальный алгоритм нахождения вектора решения. Для этого мы последовательно, начиная с x_1, проверяем, каким группам переменных можно придать значение 0, чтобы система при этом оставалась совместной. При присвоении переменной x_k нулевого значения длина кода редуцированной задачи не превышает длины кода первоначальной задачи. Проверка совместности осуществляется с помощью описанного выше алгоритма эллипсоидов. Такую проверку нужно осуществлять не более n раз. В редуцированной задаче отбросим несущественные ограничения. Это можно сделать, последовательно заменяя неравенства и проверяя полученную систему на несовместность. В результате получим систему уравнений с целыми коэффициентами, решение которой при добавлении нулей на места переменных, отброшенных в процессе редуцирования, дает точку, являющуюся решением системы неравенств (2.39).

5. Ускорение сходимости метода эллипсоидов. Хотя метод эллипсоидов сходится по функционалу в определенном смысле со скоростью геометрической прогрессии, требуется примерно 4,6 n^2 итераций, чтобы гарантировать уменьшение в 10 раз отклонения рекорда по функционалу от оптимального значения. Это практически медленная сходимость. Поэтому были предприняты определенные усилия, чтобы, оставаясь в рамках общей схемы метода эллипсоидов, ускорить сходимость.

При конструировании алгоритмов этого типа использованы следующие приемы.

1. *Проведение более "глубоких" отсечений.* Рассмотрим возможность проведения более "глубоких" отсечений при решении общей задачи выпуклого программирования. Допустим, что для решении задачи (2.29), (2.30) применяется алгоритм (2.23)–(2.26).

Тогда если в точке x_0 для некоторого $i = \bar{i}$ значение функции $f_{\bar{i}}(x_0) > 0$, то в качестве отсекающей гиперплоскости можно использовать следующую:

$$P(x_0) = \Big\{x\colon (g_{f_{\bar{i}}}(x_0), x - x_0) + f_{\bar{i}}(x_0) = 0\Big\}. \tag{2.40}$$

Если для всех $i = \overline{1, m}$ $f_i(x_0) \le 0$ и $f_0(x_0) > \overline{f}_0$, где $\overline{f}_0$ – достигнутый рекорд по функционалу, то в качестве отсекающей гиперплоскости можно взять

$$P(x_0) = \Big\{x\colon (g_{f_0}(x_0), x - x_0) + f_0(x_0) - \overline{f}_0 = 0\Big\}. \tag{2.41}$$

В этом случае используем полупространство

$$H(x_0) = \Big\{x\colon (g_{f_0}(x_0), x - x_0) + f_0(x_0) - \overline{f}_0 \le 0\Big\}$$

как область локализации оптимального множества задачи (2.29), (2.30).

Указанные в (2.40), (2.41) гиперплоскости отличаются от используемых в алгоритме (2.23)–(2.26) параллельным сдвигом в сторону более сильного отсечения области локализации. Величину $\overline{f}_0$ в формуле (2.41) назовем *константой сдвига;* в (2.40) $\overline{f}_0 = 0$.

2. *Построение эллипсоида минимального объема.* Описание нового эллипсоида вокруг полученной области локализации можно производить после двух или большего числа отсечений предыдущего эллипсоида. При этом следует ожидать более быстрого (в среднем на одно отсечение) уменьшения объема области локализации.

На основе указанных возможностей разработано несколько вариантов алгоритмов с кратными (в основном двойными) и "глубокими" отсечениями. Основную техническую сложность представляет нахождение эллипсоида минимального объема, описанного вокруг тел, получаемых из сферы после двух или большего числа отсечений. Заметим, что после r-кратного отсечения сферы ($r < n$) оптимальный эллипсоид сохраняет из-за свойств симметрии $n - r$ одинаковых главных осей, так что для превращения его в сферу потребуется не более чем r преобразований типа растяжения пространства.

3. *Проведение двойных отсечений.* Также предложены конкретные алгоритмы метода эллипсоидов, основанные на двойных отсечениях. Рассмотрим вопрос о скорости сходимости различных модификаций метода эллипсоидов при решении задачи минимизации выпуклой функции $f(x),\ x \in E^n$. Пусть X^* – область минимумов $f(x)$ и в E^n задана сфера $S_0 = \{x\colon ||x - x_0|| \leq r_0\}$, содержащая точки из X^*. Опуская несущественные для данного рассмотрения подробности, общую схему метода эллипсоидов опишем следующим образом.

Перед началом k-го шага ($k = 0, 1, \dots$) имеем точку $x_k \in E^n$, неособую $n \times n$-матрицу B_k и число $r_k > 0$, задающее в E^n эллипсоид $\Phi_k = \{x\colon ||B_k^{-1}(x - x_k)|| \leq r_k\}$ такой, что $x^* \in \Phi_k$ матрица B_0 – единичная и $\Phi_0 = S_0$), тогда k-й шаг состоит из двух этапов.

Этап 1 (*уточнение локализации*). В соответствии с некоторой процедурой выбираем совокупность пар $\{x_k^l, f_{kl}^s\}$ (x_k^l – вектор из E^n, f_{kl}^s – число, так называемое "значение для сдвигов", $l = 0,\ 1,\ 2, \dots, l_k$.) Каждая пара из этой совокупности определяет отсекающую гиперплоскость

$$P(x_k^l, f_{kl}^s) = \Big\{x\colon (g(x_k^l),\ x - x_k^l) + f(x_k^l) - f_{kl}^s = 0\Big\},$$

задающую в E^n полупространство вида

$$H(x_k^l, f_{kl}^s) = \Big\{x\colon (g(x_k^l),\ x - x_k^l) + f(x_k^l) - f_{kl}^s \leq 0\Big\}$$

такое, что $x^* \in H(x_k^l, f_{kl}^s)$. Тело

$$Z_k = \Phi_k \bigcap \left(\bigcap_{l=0}^{l_k} H(x_k^l, f_{kl}^s) \right) \tag{2.42}$$

оказывается "локализующим", так как, очевидно, $x^* \in Z_k$. Основное свойство, которым должна обладать указанная процедура, заключается в следующем: для тела Z_k, определенного (в силу (2.42)) выбранными в процедуре парами, существует "включающий" эллипсоид $\Phi(Z_k)$ $(\Phi(Z_k) \supset Z_k)$ такой, что

$$q(Z_k) = \frac{v(\Phi(Z_k))}{v(\Phi_k)} \leq Q < 1.$$

Этап 2 (*пересчет параметров*). Полагаем $\Phi_{k+1} = \Phi(Z_k)$, т.е. вычисляем вектор $x_{k+1} \in E^n$, неособую $n \times n$-матрицу B_{k+1} и число $r_{k+1} > 0$ такие, что

$$\Phi_{k+1} = \{x\colon ||B_{k+1}^{-1}(x - x_{k+1})|| \leq r_{k+1}\} = \Phi(Z_k).$$

Рассмотрим последовательность "рекордов" $\{f_k^r\}_{k=0}^{\infty}$, порождаемую при работе алгоритма эллипсоидов по правилу

$$f_k^l = \min\{f_{k-1}^r, \min_l\{f(x_k^l) \quad (l = \overline{0, l_k})\}\};$$
$$k = 1, 2, \ldots;$$
$$f_0^r = f(x_0).$$

Теорема 2.20. *Пусть* $x^* \in \mathrm{int} S_0$ *и* $||g(x)|| \leq L$ *при* $x \in S_0$. *Пусть выбор значений для сдвигов подчинен условию*

$$f_{kl}^s = \min\{f_{k-1}^r, \min_j\{f(x_k^j) \quad (j = \overline{0, l)}\}\} \quad (l = \overline{0, l_k}),$$

$$k = 0, 1, 2 \ldots$$

Тогда последовательность матриц $\{B_k\}_{k=0}^{\infty}$, *чисел* $\{r_k\}_{k=0}^{\infty}$ *и* $\{f_k^r\}_{k=0}^{\infty}$, *генерируемые алгоритмом эллипсоидов, удовлетворяют неравенствам*

$$f_k^r - f(x^*) \leq 2Lr_{k+1}\sqrt{\lambda(B_{k+1}B_{k+1}^*)},\ k = 0, 1, 2, \ldots,$$

где $\lambda(B_{k+1}B_{k+1}^*)$ – *минимальное собственное значение матрицы* $H_k = B_{k+1}B_{k+1}^*$.

Следствие 2.3. *В условиях теоремы 2.19 справедливы неравенства*

$$f_k^r - f(x^*) \leq 2Lr_0 \left(\prod_{i=0}^{k} q(Z_i) \right)^{\frac{1}{n}} \leq 2Lr_0 Q^{\frac{k}{n}},\ k = 0, 1, 2, \ldots$$

Глава 3

Приложения методов недифференцируемой оптимизации

Эта глава посвящена приложениям методов недифференцируемой оптимизации:

во-первых, для построения точных штрафных функций при сведении задач выпуклой условной оптимизации к задачам безусловной недифференцируемой оптимизации;

во-вторых, для установления существования полиномиальных алгоритмов решения задач оптимизации;

в-третьих, для разработки схем декомпозиции по ограничениям и по переменным решения задач математического программирования;

в-четвертых, для конкретизации алгоритмов на основе общих схем декомпозиции по ограничениям и переменным при решении задач блочной структуры (задач блочного линейного и выпуклого программирования и блочных задач дробно-линейного и дробно-выпуклого программирования);

в-пятых, для разработки схем декомпозиции по ресурсам и решения задач двухуровневого планирования и управления производственных и транспортных процессов.

Применение субградиентных методов позволили построить общие схемы и процедуры решения задач условной оптимизации, в результате которых исходные задачи сводятся к решению задач безусловной недифференцируемой оптимизации.

3.1. Общая схема применения методов недифференцируемой оптимизации

По своей сущности, методы недифференцируемой оптимизации построены для минимизации или максимизации негладких функций, для которых вместо обычного градиента определяется их субградиент или субградиентное множество в любой точке рассматриваемого пространства. Построенные субградиентные методы, основные из которых приведены в предыдущей главе, предполагают, что известны формулы и процедуры вычисления обобщенного градиента в любой точке оптимизируемой функции. Так как методы обобщенного градиента устойчивы по сходимости относительно точности вычисления субградиента, то вместо точного направления спуска в них можно использовать некоторые априорные направления (иногда, даже случайные), которые не на каждом шаге метода приведет к уменьшению значения функционала. Однако в целом, субградиентные методы позволят получить оптимальные решения задач оптимизации негладких функций с достаточной высокой точности, факт, который делает их привлекательными для решения большого диапазона сложных задач недифференцируемой оптимизации.

Широкое применение субградиентные методы нашли при построение общих схем и процедур решения задач условной минимизации, в которых используемые приемы сводит их к решению задач безусловной недифференцируемой оптимизации.

Методы недифференцируемой оптимизации применяются с целью получения более эффективных алгоритмов, схем и процедур решения известных задач математического программирования, среди которых можно выделить следующие классы:

- задачи выпуклого и дробно-выпуклого, линейного и дробно-линейного программирования;

- задачи квадратичной и полиномиальной оптимизации;

- задачи большой размерности и блочной структуры;

- задачи оптимизации на графах и транспортных сетях, задачи транспортного и производственно-транспортного типа;

- задачи булевого и дискретного программирования;

- задачи стохастического программирования и многоэтапной оптимизации.

Как правило, многие из конкретных практических задач приведенных выше классов являются задачами дифференцируемой оптимизации, для решения которых можно применять имеющиеся прямые методы и алгоритмы. Однако многие из них имеют большие размерности или специальную структуру, и поэтому для их эффективного решения применяются различные приемы сведения задач условной оптимизации к задачам недифференцируемой безусловной оптимизации, для решения которых используются субградиентные методы.

Основные приемы, которые используются при сведении задач условной оптимизации к безусловных, являются:

- построение функции Лагранжа;

- использование негладких штрафных функций;

- применение схем декомпозиции по ограничениям, переменным и ресурсам;

- использование субградиентных методов недифференцируемой оптимизации.

При применении субградиентных методов для решения задач недифференцируемой оптимизации необходимо выполнить следующие три основных этапа.

1. Используя полученную субградиентным методом текущую точку необходимо решить некоторые вспомогательные задачи оптимизации, связанные с исходной задачи. Для таких задач находится оптимальное или приближенное решение (иногда достаточно найти некоторое допустимое решение). В некоторых случаях также требуется найти и двойственное решение.

2. Для построенной функции безусловной (как правило, негладкой) оптимизации вычислить значения обобщенного градиента в текущей точке. Формулы вычисления субградиента соответствуют исходной задачи. Как правило, в них используются полученные прямые и двойственные решения подзадач условной оптимизации, связанные с исходной задачи и полученных функций оптимизации в соответствии с используемым приемом их построения. Одними из таких приемов являются построение функции Лагранжа или использование негладких штрафных функций.

3. В соответствии с примененным методом недифференцируемой оптимизации, найденным направлением движения (субградиентом) и выбранным шагом сдвига осуществить переход в новую точку.

В качестве одного из примеров возможной схемы применения методов недифференцируемой оптимизации можно привести задачу двухэтапной оптимизации

$$\max_{u \geq 0} \min_{x \in X} F(x, u) = \max_{u \geq 0} \{\Psi\,(u(x^*)) = \min_{x \in X} F(x, u)\},$$

в которой процесс вычисления разбивается на два этапа:

- первый этап состоит в решение внешней задачи

$$\max_{u \geq 0} \Psi\left(u(x)\right),$$

- второй этап состоит в решение внутренней задачи

$$\Psi(u) = \min_{x \in X} F(x, u).$$

Пусть для решения задачи максимизации функции $\Psi(u)$ применяем один из субградиентных методов, для которого выполнены k шагов и найдено решение u_k. Тогда на $(k+1)$-м шаге субградиентного метода необходимо выполнить следующие три основных этапа.

1. Найти решение $x^*(u_k)$ задачи

$$F\left(x^*(u_k), u_k\right) = \min_{x \in X} F(x, u_k),$$

для решения которой применяются алгоритмы и методы в соответствии со структурой множества X и функционала $F(x, u_k)$ при фиксированных u_k.

2. Вычислить значение обобщенного градиента $g_\Psi(u)$ функции $\Psi(u)$ в точке $u = u_k$ с учетом решения $x^*(u_k)$ по формулам, которые соответствуют функции $\Psi(u)$, исходной функции $F(x, u)$ и ассоциированной функции Лагранжа к исходной задаче оптимизации.

3. Провести переход в новую точку u_{k+1} по одной из стандартных формул

$$u_{k+1} = \max\{0, u_k + h_{k+1}\, g_\Psi(u)\},$$

где h_{k+1} – величина шага, которая выбирается в соответствии с субградиентным методом.

Приведенная выше трехэтапная схема применения субградиентных методов универсальна с точки зрения решения различных задач условной оптимизации путем построения итерационного двухэтапного процесса вычислений, когда на первом этапе применяется один из методов недифференцируемой оптимизации, а на втором этапе используются имеющиеся эффективные алгоритмы решения внутренней задачи оптимизации, которая имеет структуру в соответствии с используемым приемом ее свертывания в функционал внешней задачи и структуру всех или части ограничений исходной задачи.

С точки зрения проведения конкретных расчетов на ЭВМ, данная схема проявила свою эффективность тем, что имея стандартную программу реализации одного из субградиентных методов (в частности, это относится к известной программе RALG разработанной Н.Г. Журбенко из пакета программ ПЛАНЕР [662] реализации r-алгоритма), для решения конкретной задачи необходимо разработать свои подпрограммы вычисления значений субградиента в текущей точке и комплекса программ решения внутренних задач.

В дальнейшем такая схема берется за основу при применение методов недифференцируемой оптимизации для решения различных задач математического программирования, и в частности, в данной книге автор применяет ее для решения задач дробно-линейного и дробно-выпуклого программирования, а также для решения различных задач транспортного типа с дробными функционалами. В каждом конкретном случае выявляются те вспомогательные оптимизационные подзадачи для которых имеются эффективные методы их решения, а также определяются формулы вычисления обобщенных градиентов в текущих точках выбранного метода недифференцируемой оптимизации.

Также при разработке эффективных методов решения задач дробной оптимизации автором данной монографии учитываются полученные результаты и опыт применения субградиентных методов при решении задач линейного и выпуклого программирования, квадратичной, полиномиальной и стохастической оптимизации [470, 471, 615, 712, 722, 727, 735, 737, 738]. Приведенные ниже приемы и схемы применения методов недифференцируемой оптимизации, в следующих главах данной книге, используются при разработке различных оригинальных методов и алгоритмов решения задач дробного программирования.

3.2. Общая схема декомпозиции по ограничениям и негладкие штрафные функции

1. Схема декомпозиции по ограничениям. Функция Лагранжа эффективно может быть использована для получения двойственных оценок задач оптимизации и реализации схем декомпозиции в задачах математического программирования и, в частности, для задач выпуклого и линейного программирования. Рассмотрим этот вопрос в общем случае задачи математического программирования: определить

$$f_0^* = \inf f_0(x) \tag{3.1}$$

при ограничениях

$$f_i(x) \le 0 \quad (i = \overline{1, m}), \tag{3.2}$$

$$x \in X \subset E^n. \tag{3.3}$$

Если ограничения (3.2) и (3.3) несовместны, то полагаем $f_0^* = \infty$. Рассмотрим функцию Лагранжа

$$L(x,u) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x),$$

где $u = \{u_i\}_{i=1}^m$ – множители Лагранжа для ограничений (3.2). При $u \geq 0$ имеем неравенство

$$L(x,u) \leq f_0(x)$$

для любого допустимого решения x задачи (3.1)–(3.3).

Поэтому

$$\Psi(u) = \inf_{x \in X} L(x,u) \leq f_0^*$$

при $u \geq 0$, т.е. $\Psi(u)$ дает оценку снизу для оптимального значения целевой функции задачи (3.1)–(3.3). Чтобы получить наилучшую оценку из этого класса оценок, которые мы будем называть двойственными, необходимо вычислить величину

$$\Psi^* = \sup_{u \geq 0} \Psi(u).$$

Задачу нахождения

$$\inf_{x \in X} L(x,u)$$

будем называть внутренней, а задачу

$$\sup_{u \geq 0} \Psi(u)$$

внешней.

В общем случае указанная выше схема лежит в основе реализации принципов, использующих декомпозицию по ограничениям.

Известно, что при достаточно общих условиях внешняя задача сводится к задаче вогнутого программирования.

Теорема 3.1. *Если функции $f_0(x)$, $f_i(x)$, $i = \overline{1,m}$ непрерывны на компактном множестве $X \subset E^n$, то $\Psi(u)$ вогнута и определена при всех u. Пусть $x^*(\overline{u})$ означает точку минимума $L(x,\overline{u})$ по $x \in X$. Тогда вектор $\{f_i(x^*(\overline{u})) \quad (i = \overline{1,m})\}$ является обобщенным градиентом (суперградиентом) функции $\Psi(u)$ в точке $\overline{u}$.*

Если $x^*(\overline{u})$ определяется единственным образом, то в соответствующей точке функция $\Psi(\overline{u})$ непрерывно дифференцируема. В противном случае множество суперградиентов может содержать более одной точки, а в точке $\overline{u}$ происходит разрыв градиента.

Теорема 3.1 показывает, что если внутренняя задача решается достаточно просто, то для решения внешней задачи можно применять методы максимизации негладких вогнутых функций.

Таким образом, для общей задаче математического программирования (3.1)–(3.3), при решении внутренней задачи $\inf\limits_{x\in X} L(x,u)$ учитываются лишь ограничения $x \in X$, остальные ограничения с определенными множителями Лагранжа входят в целевую функцию $L(x,u)$. В процессе решения внешней задачи $\sup\limits_{u\geq 0} \Psi(u)$ мы находим приближенные значения оптимальных множителей Лагранжа u^*. Если $x^*(u^*)$ единственна, то оно является решением задачи (3.1)–(3.3). В противном случае решение задачи (3.1)–(3.3) следует искать среди множества $X^*(u^*)$ оптимальных решений внутренней задачи при $u = u^*$, а именно: в множестве $X^*(u^*)$ выбрать точки, которые удовлетворяют ограничениям (3.2).

Таким образом, если для решения внешней задачи $\sup\limits_{u\geq 0} \Psi(u)$ использовать один из субградиентных методов, то на его $(k+1)$-м шаге необходимо выполнить следующие три этапа.

1. При фиксированных $u = u^k$ решить задачу оптимизации по переменных x функции Лагранжа, т.е. решить задачу: найти решение $x^*(u^k)$ для которого

$$L(x^*(u^k), u^k) = \inf_{x\in X} L(x, u^k).$$

2. С учетом полученного решения $x^*(u^k)$ найти множество значений обобщенного градиента функции $\Psi(u)$ в точке u^k по общей формуле

$$G_\Psi(u^k) = \left\{ g_i(u^k) = f_i\left(x^*(u^k)\right) \quad (i = \overline{1,m}) \right\}.$$

3. Вычислить значения координат новой точки u^{k+1} по формуле, соответствующая субградиентного метода, которая в общем имеет вид

$$u_i^{k+1} = \max\{0;\ u_i^k + h_{k+1} g_i(u^k)\} \quad (i = \overline{1,m}),$$

где h_{k+1} – величина шага в субградиентном методе.

Для задачи выпуклого программирования имеет место

Теорема 3.2 (Куна-Таккера). *Пусть имеется задача выпуклого программирования: найти минимум функции*

$$f_0(x) \tag{3.4}$$

при ограничениях

$$x \in X \subset E^n,\ f_i(x) \le 0 \quad (i = \overline{1,m}), \tag{3.5}$$

где X – замкнутое выпуклое множество, $f_\tau(x)$ $(\tau = \overline{0,m})$ – выпуклые функции. Пусть $u = \{u_1, u_2, \ldots, u_m\}$, $L(x,u) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x)$ – функция Лагранжа и пусть выполняется условие Слейтера: существует такая точка $\overline{x} \in X$, что $f_i(\overline{x}) < 0$ для $i = \overline{1,m}$.

Тогда для того, чтобы x^* было оптимальным решением задачи (3.4), (3.5)*, необходимо и достаточно существование такого u^*, что $\{x^*, u^*\}$ – седловая точка функции Лагранжа на множестве $X \times \{u\colon u \ge 0\}$; при этом выполняется условие*

$$u_i^* f_i(x^*) = 0 \quad (i = \overline{1,m}).$$

Таким образом, когда X – выпуклое множество, $f_\tau(x)$ $(\tau = \overline{0,m})$ – выпуклые функции и для соответствующей задачи выпуклого программирования вида (3.4), (3.5) выполняется условие Слейтера, то в силу теоремы Куна-Таккера Ψ^* будет совпадать с f_0^*, т.е. оценка Ψ^* оказывается точной. При этом алгоритм вычисления Ψ^* будет соответствовать общей схеме декомпозиции по ограничениям. Действительно, если ограничения (3.5) разбить на две части:

1) $x \in X$;

2) $f_i(x) \le 0 \quad (i = \overline{1,m})$,

то решение исходной задаче выпуклого программирования сводится к задаче нахождения седловой точке функции Лагранжа по рассматриваемой выше схеме декомпозиции по ограничениям. Так как X – выпуклое множество, а $f_\tau(x)$ $(\tau = \overline{0,m})$ выпуклые функции, то внутренняя задача $\inf\limits_{x \in X} L(x,u)$ будет задачей выпуклого программирования.

2. Точные негладкие штрафные функции. На основании функции Лагранжа и теоремы Куна-Таккера можно построить точные негладкие функции штрафа, позволяющие свести задачу выпуклого программирования к задаче безусловной минимизации.

Рассмотрим задачу выпуклого программирования: найти

$$\min_{x \in E^n} \bigl\{ f_0(x)\colon x \in X,\ f_i(x) \le 0 \quad (i = \overline{1,m}) \bigr\}. \tag{3.6}$$

Пусть множество оптимальных решений X^* непусто, $f_0(x^*) = f_0^*$ при $x^* \in X^*$ и пара $\{x^*, u^*\}$ образует седловую точку функции Лагранжа

$$L(x, u) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x), \quad u \geq 0.$$

Как было показано в первой главе для решения задачи выпуклого программирования могут быть использованы точные негладкие штрафные функции, которые сводят задачу (3.6) к задаче безусловной оптимизации. Рассмотрим основные штрафные функции и приведем полученные задачи недифференцируемой оптимизации, которые применяются при решении задач выпуклого программирования (3.6).

1. Пусть $s = \{s_i\}_{i=1}^m \geq 0$ – вектор штрафных мультипликаторов. Построим штрафную функцию

$$S_1(x, s) = f_0(x) + \sum_{i=1}^{m} s_i \max\{0, f_i(x)\},$$

с помощью которой задача (3.6) сводится к решению задачи недифференцируемой оптимизации

$$\rho^* = \min_{x \in X} S_1(x, s). \tag{3.7}$$

2. Пусть $p > 0$ – штрафной множитель. Построим штрафную функцию

$$S_2(x, p) = f_0(x) + p \max\{0, \max_{1 \leq i \leq m} f_i(x)\},$$

с помощью которой задача (3.6) сводится к решению задачи недифференцируемой оптимизации

$$\min_{x \in X} S_2(x, p). \tag{3.8}$$

3. Пусть заданы одномерные выпуклые функции $\alpha_i(t)$ $(i = \overline{1, m})$, принимающие значение 0 при $t \leq 0$ и $\alpha_i(t) > 0$ для $t > 0$. Построим штрафную функцию

$$S_3(x, \alpha) = f_0(x) + \sum_{i=1}^{m} \alpha_i\big(f_i(x)\big),$$

с помощью которой задача (3.6) сводится к решению задачи недифференцируемой оптимизации

$$\min_{x \in X} S_3(x, \alpha), \tag{3.9}$$

где $\alpha = \{\alpha_i\}$ – вектор-функция.

4. Пусть задана одномерная выпуклая функция $\beta(t)$, принимающая значение 0 при $t \le 0$ и $\beta(t) > 0$ при $t > 0$. Построим штрафную функцию

$$S_4(x, \beta) = f_0(x) + \beta\big(\max_{1 \le i \le m} f_i(x)\big),$$

с помощью которой задача (3.6) сводится к решению задачи недифференцируемой оптимизации

$$\min_{x \in X} S_4(x, \beta). \tag{3.10}$$

Для решения задач минимизации функций $S_1(x, s), S_2(x, p), S_3(x, \alpha)$ и $S_4(x, \beta)$ можно использовать субградиентные методы негладкой оптимизации.

Рассмотрим задачу выпуклого программирования (3.6), для решения которой применяется метод негладких штрафных функций. Пусть $R(x)$ – множество допустимых решений задачи (3.6).

Построим штрафные функции для задачи (3.6) по общей формуле

$$P_t(x) = f_0(x) + S_t(x) \quad (t = \overline{1, 4})$$

и рассмотрим задачи

$$\min_{x \in X} P_t(x) \quad (t = \overline{1, 4}), \tag{3.11}$$

где $S_t(x)$ вычисляется по одной из формул:

$$S_1(x) = \sum_{i=1}^{m} s_i \ \max\{0, f_i(x)\},$$

$$S_2(x) = p \max\{0, \max_{1 \le i \le m} f_i(x)\},$$

$$S_3(x) = \sum_{i=1}^{m} \alpha_i(f_i(x)),$$

$$S_4(x) = \beta(\max_{1 \le i \le m} f_i(x)).$$

Так как функции $S_t(x) \quad (t = \overline{1,4})$ являются негладкими и имеют точки разрыва градиента, то задачи (3.11) являются задачами недифференцируемой оптимизации.

Для каждого $t = 1, 2, 3$ и 4 задача (3.11) соответствует задачам (3.7), (3.8), (3.9) и (3.10).

Так как функции $P_t(x)$, являются недифференцируемыми, то для решения любой из задач оптимизации типа $\min\limits_{x \in X} P_t(x)$ используется субградиентный метод. Рассмотрим общую схему решения задач (3.11) и приведем формулы определения субградиентов для каждой из них в отдельности.

Пусть на k-м шаге субградиентного метода минимизации функций $P_t(x)$ найдена точка x^k. Тогда на очередном $(k+1)$-м шаге субградиентного метода необходимо провести следующие три этапа.

1. Вычислим значения штрафной функции $S_t(x) \quad (t = \overline{1,4})$ в точке x^k.

2. Вычислим значения обобщенного градиента функции $P_t(x)$ в точке x^k по формуле

$$g_{P_t}(x^k) = \begin{cases} g_{f_0}(x^k), \text{ если } x^k \in R(x); \\ g_{f_0}(x^k) + g_{S_t}(x^k), \ x^k \notin R(x), \end{cases}$$

где g_{f_0} субградиенты функций $f_0(x)$ вычисленных в точке x^k, а g_{S_t} – соответственно субградиент одной из функций $S_t(x) \quad (t = \overline{1,4})$ вычисленных в точке x^k.

Отметим некоторые особенности вычисления субградиентов функций $S_t(x), \ \ t = \overline{1,4}$ в точке x^k. Пусть $x^k \notin R(x)$, а I^+ – дискретное множество индексов i, для которых $f_i(x^k) > 0$. Тогда для каждой функции в отдельности имеем:

1) $g_{S_1}(x^k) = \sum\limits_{i \in I^+} s_i \cdot g_{f_i}(x^k),$

где $g_{f_i}(x^k)$ значения субградиентов функций $f_i(x), \ i \in I^+$, вычисленных в точке x^k;

2) $g_{S_2}(x^k) = p \cdot g_{f_{i^*}}(x^k),$

где $g_{f_{i^*}}(x^k)$ значение градиента функции $f_{i^*}(x)$, вычисленной в точке x^k, а i^* значение индекса i, для которого

$$f_{i^*}(x^k) = \max_{i \in I^+} f_i(x^k);$$

3) $g_{S_3}(x^k) = \sum\limits_{i \in I^+} g_{\alpha_i} \cdot g_{f_i}(x^k)$,

где g_{α_i} – субградиенты функций α_i, а $g_{f_i}(x^k)$ – субградиенты функций $f_i(x)$, $i \in I^+$, вычисленных в точке x^k;

4) $g_{S_4}(x^k) = g_\beta \cdot g_{f_{i^*}}(x^k)$,

где g_β – субградиент функции β, $g_{f_{i^*}}(x^k)$ – субградиент функции $f_{i^*}(x)$, а i^* – значение индекса i, для которого $f_{i^*}(x^k) = \max\limits_{i \in I^+} f_i(x^k)$.

3. Вычислим значения новой точки x^{k+1} с учетом ее проектирования на множестве X, т.е.:

- находим

$$x^{k+1} = x^k - h_{k+1}\, g_{P_t}(x^k),$$

где h_{k+1} – величина шага в субградиентном методе;

- выполняем операцию проектирования точки x^{k+1} на множестве X, т.е. находим новые значения точки $x^{k+1} = \Pi R_X[x^{k+1}]$, где ΠR_X – оператор проектирования, структура которого зависит от структуры множества X.

Следует отметить, что в общей схеме декомпозиции по ограничениям для решения исходной задаче (3.1)–(3.3) применяется процесс двухэтапной оптимизации, в котором на первом этапе (внешнем) решается задача недифференцируемой оптимизации по вектору переменных u, размерность которого зависит от количества ограничений в исходной задачи. На втором этапе (внутреннем) неоднократно решается задача на множестве переменных $x \in X$, размерность которой соответствует вектору оптимизации x, а эффективность применяемого алгоритма зависит от структуры ограничений, образующие множество X.

Другое имеет место в методе штрафных функций, когда недифференцируемая оптимизация проводится на множестве переменных x с проектированием их значений на множестве X. Множество ограничений задачи $f_i(x) \le 0 \quad (i = \overline{1,m})$ учитываются только при вычисления значений обобщенных градиентов в заданной точке x. Поэтому при выборе метода штрафных функций следует учитывать размерность вектора переменных x и сложности проведения операции проектирования точки x на множестве X. Также следует отметить, что в таких методах нет необходимости на каждом шаге субградиентного метода решать оптимизационные задачи.

3.3. Особенности применения схемы декомпозиции по ограничениям для задач выпуклого и линейного программирования

1. Задачи выпуклого программирования. Рассмотрим задачу выпуклого программирования следующего вида [470, 639, 640, 724]: найти

$$f_0^* = \inf f_0(x), \tag{3.12}$$

$$x \in X \subseteq E^n, \tag{3.13}$$

$$f_i(x) \leq 0 \quad (i = \overline{1, m}), \tag{3.14}$$

где X – закрытое выпуклое множество, $f_\tau(x)$, $\tau = 0, 1, \ldots, m$ – выпуклые функции, определенные на открытом выпуклом множестве $Y \subseteq E^n$, включающее X. Для обычной функции Лагранжа $L(x, u)$ и векторе $u = \{u_1, \ldots, u_m\}$ множителей Лагранжа рассмотрим задачу: найти

$$\Psi(u) = \inf_{x \in X} L(x, u), \tag{3.15}$$

где

$$L(x, u) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x). \tag{3.16}$$

Для любого фиксированного $u \geq 0$ функция $L(x, u)$ является выпуклой и определенной по x на множестве $Y \supseteq X$. Таким образом, для любого $u \geq 0$ внутренняя задача (3.15) является задачей выпуклого программирования. Из этого следует, что если $\Psi(u)$, имеет непустое область определения ($dom\,\Psi$) в E^m, то $\Psi(u)$ является вогнутой функцией на $dom\,\Psi$. Значение

$$\sup_{u \in dom\,\Psi} \Psi(u) = \Psi^* \tag{3.17}$$

называется *нижней двойственной оценке* для f_0^* (если задача (3.12)–(3.14) не имеет допустимое решение, то ставим $f_0^* = +\infty$).

По теореме Куна-Таккера 3.2, если имеет место условие Слейтера и x^* является оптимальным решением задачи выпуклого программирования (3.12)–(3.14), то всегда существует $u^* = \{u_1^*, \ldots, u_m^*\} \geq 0$ для которого пара (x^*, u^*) является седловой точкой функции Лагранжа $L(x, u)$:

$$L(x^*, u)_{u \geq 0} \leq L(x^*, u^*)_{x \in X}$$

и $f_0^* = f(x^*) = \Psi(u^*) = \Psi^*$. В таком случае любое решение $\overline{u}^* \geq 0$ внутренней задаче (3.17) формирует u-составляющую для общей седловой точке $\{x(\overline{u}^*), \overline{u}^*\}$ функции Лагранжа (3.16).

Обозначим через $X(u)$ множество решений задачи (3.15). Предположим, что u^* являются оптимальными множителями Лагранжа, а $X(u^*)$ содержит единственную точку (решение задачи (3.15) однозначно), тогда $x(u^*)$ является решением задачи (3.12)–(3.14). Иначе, только часть решений из $X(u^*)$ удовлетворяет ограничениям (3.14). Отметим, что вогнутая функция $\Psi(u)$ является почти везде дифференцируемой на $intdom\,\Psi$. Точки недифференцируемости соответствуют неоднозначности решений $X(u)$.

Пусть $\overline{u} \in intdom\,\Psi$, и $\Psi(\overline{u}) = L(x(\overline{u}), \overline{u})$. Тогда вектор $g = \{f_i(x(\overline{u}))\}_{i=1}^m$ является суперградиентом функции Ψ в точке $\overline{u}$. Если решение $x(\overline{u}) \in X$ определено однозначно, то $\{f_i(x(\overline{u}))\}_{i=1}^m$ является градиентом функции Ψ в точке $\overline{u}$.

Пусть u^* является оптимальным вектором множителей Лагранжа, $u^* = \{u_1^*, \ldots, u_m^*\}$. Пусть $I_0 = \{i | u_i^* = 0\}$ – множество индексов, которые отмечают какие ограничения $\{f_i(x) \leq 0,\ i \in I_0\}$ являются несущественными для задачи (3.12)–(3.14) и которые в дальнейшем отбрасываются из рассматриваемой задачи при нахождения ее оптимального решения.

Если множество $X(u^*)$ содержит только одну точку $x(u^*)$, то $f_i(x(u^*)) = 0$ для $i \in I_1$, где $I_1 = \{1, \ldots, m\} \setminus I_0$, и $f_i(x(u^*)) \leq 0$ для $i \in I_0$, т.е. $x(u^*)$ является допустимой точкой. В таком случае функция $\Psi(u)$ является дифференцируемой в точке u^*, и необходимые и достаточные условия для максимизации $\Psi(u)$ на множестве $u \geq 0$ имеют вид:

$$g_{\Psi}(u^*) = \{f_i(x(u^*))\}_{i=1}^m \begin{cases} = 0, & \text{для} \quad i \in I_1, \\ \leq 0, & \text{для} \quad i \in I_0. \end{cases}$$

В общем случае, нахождение допустимого решения $x^* \in X(u^*)$ не является тривиальной задачей. Имеется некоторая процедура нахождения решений таких задач.

I. Найти u^* с достаточной хорошей точности, определить множество индексов I_0 для которых множители u_i^* равны нулю и отбросить несущественных ограничений $f_i(x) \leq 0$. Решить редуцированную задачу прямым методом, соответствующей исходной задачи. Такого типа процедуры рекомендуется использовать при решении блочных задач выпуклого программирования с достаточно большим числом блоков и при линейных ограничениях.

II. Однозначность решения задачи (3.15) (и "гладкость" функции $\Psi(u)$) может быть обеспечена путем добавления к целевой функции (3.12) строго выпуклую функцию с маленьким параметром (например, $\varepsilon \sum_{j=1}^{n} x_j^2, \varepsilon > 0$). Пусть u_ε^* оптимальные множители Лагранжа для ε-возмущенной задачи. Тогда $\tilde{x}(u_\varepsilon^*)$ (единственное решение соответствующей внутренней строго выпуклой задачи) является допустимая аппроксимация оптимального решения для исходной задачи (3.12)–(3.14).

III. Используем субградиентные или ε-субградиентные процедуры максимизации $\Psi(u)$ с небольшим шагом в окрестности u^* (нулевые координаты для u^* фиксированы) с полученными средними значениями $x(u_k)$ генерируемые соответствующим процессом получения последовательности $\{u_k\}$. Основные такие процедуры приведены в предыдущей главе.

Таким образом применение схем декомпозиции по ограничениям позволяет найти решение исходной задачи с заранее заданной величине точности $\varepsilon > 0$. Такая величина позволяет для исходной задачи выделить те значения оптимального решения, которые являются ε-устойчивыми, и те ограничения и переменные исходной задачи, которые являются несущественными и не зависят от заданных ε-возмущений задачи.

Поэтому в соответствии с первой процедуры применения схем декомпозиции по ограничениям, в отличие от точных негладких штрафных функций, разделяем решение исходной задаче выпуклого программирования на два этапа.

Первый этап состоит в сведение исходной задаче выпуклого программирования к задаче безусловной недифференцируемой оптимизации, с помощью которой определяются те устойчивые значения переменных и те несущественные ограничения, которые с заранее заданной ε-аппроксимацией можно в дальнейшем исключить из рассмотрения. В результате применения этого этапа выделяется та часть переменных x и та часть ограничений, которые являются ε-устойчивыми и в дальнейшем отбрасываются из исходной задаче.

Второй этап состоит в решение некоторой усеченной задачи выпуклого программирования намного меньшей размерности чем исходная, в которой включаются только ε-неустойчивые переменные и ограничения. Таким образом, если исходная задача имела n переменных и m ограничений, то усеченная задача как правило имеет n_0 переменных и

m_0 ограничений, где n_0 и m_0 намного меньше соответствующих исходных значений n и m.

Тогда для решения усеченной задаче выпуклого программирования: найти

$$\inf f_0(x),$$

$$f_i(x) \leq 0 \quad (i = \overline{1, m_0}),$$

$$x \in X \subset E^{n_0},$$

которая имеет намного меньших размерностей чем исходная задача, можно использовать прямые методы, в том числе и методы штрафных функций, для которых повышается эффективность их применения за счет уменьшенных размерностей исходной задачи (3.12)–(3.14).

2. Задачи линейного программирования. В работах [2, 3] схемы декомпозиции по ограничениям использованы для решения задач линейного программирования блочно-диагональной структуры, для которых если некоторые из ограничений задачи удовлетворены, тогда существует достаточно малый $\varepsilon_0 > 0$ такой, что для $\varepsilon < \varepsilon_0$, $\tilde{x}(u_\varepsilon^*)$ является оптимальным решением исходной задачи линейного программирования.

Рассмотрим задачу линейного программирования следующего вида: найти

$$f^0 = \min(c, x), \tag{3.18}$$

$$(a_i, x) - b_i \leq 0 \quad (i = \overline{1, m}), \quad x \in E^n \tag{3.19}$$

в которой $m > n$ и ε_0-возмущенную задачу: найти

$$f^\varepsilon = \min \left[(c, x) + \frac{\varepsilon}{2}(x, x) \right], \ \varepsilon > 0 \tag{3.20}$$

при ограничениях (3.19).

Пусть оптимальное решение x^* задачи (3.18), (3.19) невырожденное и единственное, которое определяется пересечением n гиперплоскостей следующего типа:

$$P_i = \{x\colon (a_i, x) - b_i) = 0\}, \ i \in \{1, \ldots, n\}.$$

Без ограничения общности, будем считать, что

$$(a_i, x^*) - b_i = 0 \quad (i = \overline{1, n}),$$

$$(a_i, x^*) - b_i < 0 \quad (i = \overline{n+1, m})$$

и векторы $a_1, \ldots, a_n$ являются линейно независимы. Рассмотрим оптимальные двойственные переменные задачи (3.18), (3.19) $\overline{u} = \{\overline{u}_1, \ldots$

$\ldots,\overline{u}_n,\ldots,\overline{u}_m\} \in E^m$, для которых в соответствии со свойствами переменных x^* следует, что

$$\overline{u}_i > 0 \quad (i=\overline{1,n}); \quad \overline{u}_i = 0 \quad (i=\overline{n+1,m}).$$

Рассмотрим значения:

$$D = \det\Gamma(a_1,a_2,\ldots,a_n) > 0$$

$$D_a^i = \det\Gamma(a_1,\ldots,a_{i-1},a_{i+1},\ldots,a_n)$$

$$D_b^i = \det\left\|\begin{array}{cc} & b_1 \\ \Gamma(a_1,\ldots,a_{i-1},a_{i+1},\ldots,a_n) & \vdots \\ & b_n \\ (a_i,a_1)\ (a_i,a_2)\ \ldots\ (a_i,a_n) & b_i \end{array}\right\|$$

где $\Gamma(r_1,\ldots,r_p)$ – матрица Грама от векторов $(r_1,\ldots,r_p)$.

Пусть

$$\varphi_\varepsilon(u) = \min_x \Big[(c,x) + \frac{\varepsilon}{2}(x,x) + (u,Ax-b)\Big]. \tag{3.21}$$

Тогда

$$f^\varepsilon = \max_{u\geq 0}\varphi_\varepsilon(u) = \varphi_\varepsilon(u_\varepsilon^*).$$

Пусть x_ε^* является оптимальным решением для (3.21) при $u = u_\varepsilon^*$. Легко видеть, что x_ε^* является оптимальным решением для ε-возмущенной задаче (3.18), (3.19).

Имеет место теорема [4]:

Теорема 3.3. *Пусть ε_0 – максимальное (не обязательно конечное) решение для системы неравенств*

$$\varepsilon D_b^i \leq \overline{u}_i D \quad (i=\overline{1,n}).$$

Тогда:

а) *при $\varepsilon\in[0,\varepsilon_0)$ решение x_ε^* не зависит от ε и $x_\varepsilon^* = x_0^* = x^*$;*

б) *если $\varepsilon_0<\infty$ и номер i_0 такой, что $\varepsilon_0 D_b^{i_0} = \overline{u}_{i_0}D$, единственный, то существует (не обязательно конечное) $\varepsilon_1>\varepsilon_0$ такое, что при $\varepsilon\in[\varepsilon_0,\varepsilon_1]$*

$$(c,x_\varepsilon^*) - (c,x^*) = \overline{u}_{i_0}\frac{D_b^{i_0}}{D_a^{i_0}}\left(1-\frac{\varepsilon_0}{\varepsilon}\right).$$

Отсюда следует что, если задача линейного программирования имеет регулярное оптимальное решение, тогда ε-возмущенная задача линейного программирования имеет такое же решение для $\varepsilon \leq \varepsilon_0$, где ε_0 является достаточно малое число. Тогда, если использовать ε-возмущенность в декомпозиционных схемах, то получим допустимые решения, которые с достаточной вероятностью стремятся к решению, которое будет оптимальным для исходной задаче. Такая вероятность растет когда ε уменьшается, так как для малых ε повышаются требования относительно точности внутреннего алгоритма схемы декомпозиции.

Использование квадратичной регуляризации приводит к необходимости решать задачи квадратичного программирования для отдельных блоков. В таком случае можно использовать конечные прямые методы квадратичного программирования, эффективность которых зависит от выбранного допустимого приближения.

Следует также отметить, что при переходе от одной итерации к другой при максимизации функции $\Psi(u)$ меняются лишь коэффициенты линейной части целевой функции, поэтому удобно в качестве начального приближения на $(k+1)$-м шаге субградиентного метода выбрать оптимальное решение $x(u^k)$, полученное на предыдущем шаге.

Аналогичные приемы используются при разработке эффективных двухуровневых алгоритмов, на каждом уровне которого решается задача недифференцируемой оптимизации.

3.4. Схемы декомпозиции по ограничениям для решения блочно-диагональных задач

Как было показано ранее, для общей задачи математического программирования (3.12)–(3.14) на базе функции Лагранжа можно построить двухуровневые схемы декомпозиции по ограничениям, на первом уровне которой решается задача недифференцируемой оптимизации при простых ограничениях. На втором уровне предполагается, что задачи имеют специальную структуру. К таким структурам относятся и блочно-диагональные задачи со связующими ограничениями или переменными.

Такие декомпозиционные методы применяются для решения блочных задач выпуклого и дробно-выпуклого сепарабельного программирования, блочных задач линейного и дробно-линейного программирования, разработанные для решения задач большой размерности и блочно-диагональной структуры. Эффективность применения такого класса методов определяется простотой или спецификой рассматриваемых задач. Такие методы строятся для каждой конкретной задачи в отдельности. Мы рассмотрим только общий принцип построения таких методов под названием схем декомпозиции по ограничениям, по переменным или по ресурсам.

Эффективность применения таких методов достигается не только за счет уменьшения размерности, снижения сложности или учета специфики рассматриваемой задачи, но и за счет сведения задачи из одного класса задач оптимизации в другой (например, нелинейные задачи сводятся к классу линейных, задачи гладкой оптимизации – к задачам негладкой оптимизации, дифференцируемые – к недифференцируемым, невыпуклые – к выпуклым, дискретные – к непрерывным).

1. Задача сепарабельного выпуклого программирования. Рассмотрим задачу выпуклого программирования, в которой переменные разбиты на p групп и для каждой из них определены выпуклые функции, независимые от других переменных. Итак, пусть $x_1, x_2, \dots, x_p$ ($x_j \in R^{n_j}$, $j = \overline{1,p}$) – группы переменных, образующие вектор $x = (x_1, x_2, \dots, x_p) \in R^n$, где $n = \sum\limits_{j=1}^{p} n_j$. Для каждой группы переменных x_j $(j = \overline{1,p})$ определены выпуклые функции: $\Phi_j(x_j), F_j^i(x_j)$ $(i = \overline{1,m})$, $W_j^i(x_j)$ $(i = \overline{1,m_j})$. Будем предполагать, что все функции определены и дифференцируемы в любой точке $x_j \in R^{n_j}$.

Рассмотрим следующую задачу выпуклого программирования:

$$\sum_{j=1}^{p} \Phi_j(x_j) \to \min, \tag{3.22}$$

$$\sum_{j=1}^{p} F_j^i(x_j) \le b_i \quad (i = \overline{1,m}), \tag{3.23}$$

$$W_j^i(x_j) \le b_j^i \quad (i = \overline{1,m_j};\ j = \overline{1,p}), \tag{3.24}$$

где b_i $(i = \overline{1,m})$ и b_j^i $(i = \overline{1,m_j};\ j = \overline{1,p})$ – заданные числа.

Так как все функции определены только для одной группы переменных x_j, то задачу (3.22)–(3.24) можно рассматривать как задачу сепарабельного программирования, что очень важно при применении методов декомпозиции. Если ввести векторы функции-столбцы

$$F_j(x_j) = \begin{pmatrix} F_j^1(x_j) \\ F_j^2(x_j) \\ \dots \\ F_j^m(x_j) \end{pmatrix}, \quad W_j(x_j) = \begin{pmatrix} W_j^1(x_j) \\ W_j^2(x_j) \\ \dots \\ W_j^{m_j}(x_j) \end{pmatrix}$$

и векторы-столбцы

$$b = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix} \quad \text{и} \quad \overline{b}_j = \begin{pmatrix} b_j^1 \\ b_j^2 \\ \dots \\ b_j^{m_j} \end{pmatrix},$$

то получим следующую задачу выпуклого сепарабельного программирования, записанную в векторной форме и имеющую блочно-диагональную структуру со связующими ограничениями:

$$\sum_{j=1}^{p} \Phi_j(x_j) \to \min, \tag{3.25}$$

$$\sum_{j=1}^{p} F_j(x_j) \le b, \tag{3.26}$$

$$W_j(x_j) \le \overline{b}_j \quad (j = \overline{1,p}). \tag{3.27}$$

Построим для задачи (3.25)–(3.27) функцию Лагранжа только на множестве ограничений (3.26)

$$L(x,u) = \sum_{j=1}^{p} \Phi_j(x_j) + \Big(u, \sum_{j=1}^{p} F_j(x_j) - b\Big) =$$

$$= \sum_{j=1}^{p} \Big(\Phi_j(x_j) + uF_j(x_j)\Big) - (u,b),$$

где $u \ge 0$ – вектор множителей Лагранжа, соответствующих ограничениям (3.26).

Тогда в соответствии со схемой декомпозиции по ограничениям рассмотрим две задачи:

$$\max_{u \geq 0} \Psi(u), \tag{3.28}$$

$$\Psi(u) = \min_{x \in R(x)} L(x, u), \tag{3.29}$$

где $R(x)$ – множество допустимых значений переменных x, удовлетворяющие ограничениям (3.27). По своему определению функция $\Psi(u)$ является кусочно-линейной вогнутой функцией.

В результате применения схемы декомпозиции по ограничениям получим двухуровневый алгоритм декомпозиции: на первом уровне решается координирующая задача (3.28) нахождения двойственных оценок u^* ограничений (3.26) путем решения безусловной задачи недифференцируемой оптимизации одним из методов ОГС; на втором уровне решается задача выпуклого сепарабельного программирования (3.29) с блочными ограничениями нахождения оптимальных решений при фиксированных значениях двойственных оценок u. Тогда решение исходной задачи (3.22)–(3.24) сводится к решению задачи (3.28) недифференцируемой оптимизации, на каждом шаге которых решается задача выпуклого программирования (3.29).

Таким образом, если предположим, что при решении задачи (3.28) на k-м шаге метода ОГС имеем точку u^k, то на $(k+1)$-м шаге необходимо выполнить следующие три основных этапа.

1. При фиксированных значениях $u = u^k$ решить задачу (3.29), которая в силу сепарабельности выпуклой целевой функции $L(x, u^k)$ распадается на p задач выпуклого программирования вида

$$\Phi_j(x_j) + \big(u^k, F_j(x_j)\big) \to \min, \tag{3.30}$$

$$W_j(x_j) \leq \bar{b}_j \tag{3.31}$$

для каждого блока в отдельности. Для задачи (3.30), (3.31) применяется любой метод выпуклого программирования нахождения оптимального решения $x_j^*(u^k)$.

2. Найти обобщенный градиент функции $\Psi(u)$ в точке u^k по формуле

$$g_i(u^k) = \sum_{j=1}^{p} F_j^i\big(x_j^*(u^k)\big) - b_i \quad (i = \overline{1, m}).$$

3. Найти координаты новой точки u^{k+1} по формуле

$$u_i^{k+1} = \max\{0, u_i^k + h_{k+1} g_i(u^k)\} \quad (i = \overline{1, m}),$$

где h_{k+1} – величина шага в методе ОГС, которая выбирается в соответствии с использованным методом недифференцируемой оптимизации.

Построенная процедура применения функции Лагранжа и методов недифференцируемой оптимизации является общей схемой декомпозиции по ограничениям. Она в дальнейшем детализируется при решении конкретных задач оптимизации, в которых, применяя прием свертывания части ограничений в целевую функцию, получаем более простые задачи для второго уровня, для решения которых можно использовать эффективные алгоритмы.

2. Схема декомпозиции по ограничениям для блочных задач линейного программирования. Рассмотрим случай, когда задача (3.22)–(3.24) является блочно-диагональной задачей линейного программирования со связующими ограничениями, т.е. имеем задачу

$$\sum_{j=1}^{p} c_j x_j \to \min, \tag{3.32}$$

$$\sum_{j=1}^{p} A_j x_j \leq b, \tag{3.33}$$

$$B_j x_j \leq b_j \quad (j = \overline{1,p}), \tag{3.34}$$

$$x_j \geq 0 \quad (j = \overline{1,p}), \tag{3.35}$$

где c_j, x_j – векторы размерности $n_j \quad (j = \overline{1,p})$; b – m-мерный вектор; b_j – векторы размерности $m_j \quad (j = \overline{1,p})$; A_j – матрицы размерности $m \times n_j \quad (j = \overline{1,p})$; B_j – матрицы размерности $m_j \times n_j \quad (j = \overline{1,p})$.

Пусть $u = \{u_i\}_{i=1}^{m}$ – вектор множителей Лагранжа связующих ограничений (3.33).

Применим к задаче (3.32)–(3.35) схему декомпозиции по ограничениям, построенную на базе функции Лагранжа и методов недифференцируемой оптимизации. В качестве "простых" ограничений выберем "блочные" ограничения (3.34). Пусть $R_j(x_j)$ – множество векторов x_j, удовлетворяющих ограничениям (3.34), (3.35) при фиксированном j (будем предполагать, что все эти множества непусты и компактны). Определим $R(x) = R_1(x_1) \times R_2(x_2) \times \cdots \times R_p(x_p)$, где x – вектор, составленный из компонент $x_1, x_2, \ldots, x_p$. Тогда функция Лагранжа принимает вид

$$L(x,u) = \sum_{j=1}^{p} c_j x_j + (u, \sum_{j=1}^{p} A_j x_j - b) = \sum_{j=1}^{p} \left(c_j + A_j^T u,\, x_j\right) - (u, b).$$

Внутренняя задача, определенная как

$$\Psi(u) = \min_{x \in R(x)} L(x, u),$$

распадается на p подзадач следующего вида: найти

$$\Psi_j(u) = \min_{x_j \in R_j(x_j)} (c_j + A_j^T u, x_j) \quad (j = \overline{1, p}). \tag{3.36}$$

Тогда

$$\Psi(u) = \sum_{j=1}^{p} \Psi_j(u) - (u, b),$$

и внешняя задача имеет вид

$$\max_{u \geq 0} \Psi(u).$$

Пусть $x^*(u) = \big(x_1^*(u), \ldots, x_p^*(u)\big)$ – вектор оптимальных значений задач (3.36) при фиксированных u. Субградиент функции Ψ в точке u задается формулой

$$g_\Psi(u) = \sum_{j=1}^{p} A_j x_j^*(u) - b. \tag{3.37}$$

Для получения оптимальных двойственных переменных u^* задачи (3.32)–(3.35), соответствующих связующим ограничениям (3.33), нужно решить задачу нахождения $\max\limits_{u \geq 0} \Psi(u)$. Так как

$$\Psi(u) = \sum_{j=1}^{p} \Psi_j(u) - (u, b) = \sum_{j=1}^{p} \min_{x_j \in R_j(x_j)} (c_j + A_j^T u, x_j),$$

то функция $\Psi(u)$ является кусочно-линейной вогнутой функцией по совокупности переменных u.

Максимизация кусочно-линейной вогнутой функции $\Psi(u)$ может быть произведена одним из методов негладкой оптимизации, например методом обобщенного градиентного спуска или его модификациями с растяжением пространства.

Рассмотрим общую схему таких методов. Пусть на k-м шаге метода недифференцируемой оптимизации решения задачи $\max\limits_{u \geq 0} \Psi(u)$ имеем точку u^k. Для перехода к новой точке $u^{k+1} > 0$ необходимо определить значения обобщенного градиента (субградиента) функции $\Psi(u)$ в точке u^k. Для этого необходимо выполнить три этапа.

1. Решаем внутреннюю задачу $\Psi(u) = \min\limits_{x \in R(x)} L(x, u)$. Для этого необходимо найти решения p задач линейного программирования (3.36) при фиксированных $u = u^k$. Таким образом, необходимо найти решения $x_j^*(u^k)$ следующих задач линейного программирования:

$$\left(c_j + A_j^T u, x_j\right) \to \min,$$

$$B_j x_j \le b_j,$$

$$x_j \ge 0,$$

для всех $j = \overline{1,p}$.

2. Находим значения суперградиента. Пусть $x^*(u^k) = \big(x_1^*(u^k), \ldots$ $\ldots, x_p^*(u^*)\big)$ – вектор оптимальных решений задач (3.36). Тогда суперградиент $g_\Psi(u^k)$ функции Ψ вычисляется по формуле (3.37), т.е. имеем

$$g_\Psi(u^k) = \sum_{j=1}^{p} A_j x_j^*(u^k) - b.$$

3. Находим новую точку u^{k+1} по формуле

$$u^{k+1} = \max\left\{0, u^k + h_{k+1} \cdot g_\Psi(u^k)\right\},$$

где h_{k+1} – величина шага, которая выбирается в соответствии с рекомендациями, приведенными в предыдущей главе для конкретного метода недифференцируемой оптимизации.

Таким образом, схема декомпозиции по ограничениям состоит в решении координирующей задачи $\max\limits_{u \ge 0} \Psi(u)$ некоторым методом недифференцируемой оптимизации, на каждом шаге которого, решаются локальные задачи линейного программирования (3.36) для каждого блока в отдельности.

Следует отметить, что в конечном итоге при $u = u^*$, решая задачи (3.36) симплекс-методом, получаем для каждого блока j решение $x_j^*(u)$ и соответствующий вектор двойственных переменных v_j^* $(j = \overline{1,p})$. Вектор $\Lambda^* = (u^*, v_1^*, \ldots, v_p^*)$ является оптимальным вектором двойственных переменных задачи (3.32)–(3.35).

Для получения решения x^* задачи (3.32)–(3.35) при известном Λ^* разработано несколько процедур. Во-первых, при использовании принципа дополняющей нежесткости можно редуцировать задачу (3.32)–(3.35) к задаче меньшего объема, отбросив те ограничения, у которых оптимальные двойственные оценки равны 0. Далее, нужно приравнять к нулю значения тех переменных, у которых при оптимальных значениях двойственных переменных соответствующее двойственное ограничение выполняется с ε-запасом при некотором $\varepsilon > 0$. Выбор ε зависит от точности решения двойственной задачи. Если после исключения указанных нулевых переменных в j-м блоке остались только базисные переменные, то эти переменные можно исключить, придав им те значения, которые получены при решении задачи (3.36) при $u = u^*$, при этом связывающие ограничения нужно соответствующим образом подкорректировать.

Редуцированная задача, как правило, оказывается задачей гораздо меньшего размера, чем первоначальная. Для ее решения можно уже использовать стандартную процедуру симплекс-метода (в невырожденном случае дело сведется к решению системы линейных уравнений).

3.5. Схема декомпозиции по переменным

Наряду со схемой декомпозиции по ограничениям для решения различных задач математического программирования могут быть использованы методы, основанные на схеме декомпозиции по переменным. Она состоит в том, что часть переменных фиксируется и решается локальная задача оптимизации относительно остальных переменных. Координирующая задача заключается в том, чтобы подобрать оптимальные значения для фиксированных переменных, т.е. такие, присоединив к которым оптимальные значения оставшихся переменных соответствующих локальных задач, получим оптимальное решение первоначальной задачи.

1. Схема декомпозиции по переменным для задач выпуклого программирования. Рассмотрим схему декомпозиции по переменным применительно к задачам выпуклого программирования следующего вида: найти

$$\min_{x,y} f_0(x, y) \tag{3.38}$$

при ограничениях

$$f_i(x, y) \le 0 \quad (i = \overline{1, m}), \tag{3.39}$$

где $x \in E^n$, $y \in E^p$ – две группы переменных; $z = \{x, y\}$ – это $(n + p)$-мерный вектор всех переменных задачи; функции $f_\tau(z)$ $(\tau = \overline{0, m})$, выпуклы по совокупности переменных, определены на E^{n+p}

и ограничения (3.39) вырезают в E^{n+p} компактное множество. Зафиксируем $x = \overline{x}$ и рассмотрим задачу выпуклого программирования относительно y:

$$\varphi(\overline{x}) = \min_y f_0(\overline{x}, y) \tag{3.40}$$

при ограничениях

$$f_i(\overline{x}, y) \le 0 \quad (i = \overline{1, m}). \tag{3.41}$$

Пусть $\varphi(\overline{x})$ – оптимальное значение целевой функции в задаче (3.40), (3.41). Если задача не имеет решения, положим $\varphi(\overline{x}) = +\infty$. Функция $\varphi(\overline{x})$ выпукла, а ее субградиент может быть получен в соответствии со следующей теоремой.

Теорема 3.4. *Пусть для задачи* (3.40), (3.41) *выполняется условие Слейтера. Обозначим вектор решения через* $y(\overline{x})$, *а вектор оптимальных множителей Лагранжа – через* $u^*(\overline{x}) = \{u_i^*(\overline{x})\}_{i=1}^m$ *(существование таких множителей вытекает из теоремы Куна-Таккера). Тогда, если точка* $\overline{x}$ *является внутренней точкой области определения* $\varphi(x)$, *в ней существует субградиентное множество* $G_\varphi(\overline{x})$, *и обобщенный градиент может быть вычислен по формуле*

$$g_\varphi(\overline{x}) = \widehat{g}^{\,x}_{L^*_{\overline{x}}}\big(\overline{x}, y(\overline{x})\big), \tag{3.42}$$

где

$$L^*_{\overline{x}}(x, y) = f_0(x, y) + \sum_{i=1}^m u_i^*(\overline{x}) f_i(x, y),$$

$\widehat{g}^{\,x}_{L^*_{\overline{x}}}\big(\overline{x}, y(\overline{x})\big)$ *– n-мерная x-составляющая такого субградиента функции* $L^*_{\overline{x}}(x, y)$, *взятого в точке* $\big(\overline{x}, y(\overline{x})\big)$, *у которого p-мерная y-составляющая равна нулю.*

Следствие 3.1. *Если функции* $f_\tau(x, y)$ *являются непрерывно дифференцируемыми* ($\tau = \overline{0, m}$), *то в предположениях теоремы* 3.4

$$g_\varphi(\overline{x}) = g^x_{L^*_{\overline{x}}}\big(\overline{x}, y(\overline{x})\big) = g^x_{f_0}\big(\overline{x}, y(\overline{x})\big) + \sum_{i=1}^m u_i^*(\overline{x}) \cdot g^x_{f_i}\big(\overline{x}, y(\overline{x})\big),$$

где $g^x_{f_i}\big(\overline{x}, y(\overline{x})\big)$ *– градиенты функций* $f_\tau(x, y)$ *по x в точке* $\big(\overline{x}, y(\overline{x})\big)$ ($\tau = \overline{0, m}$).

Это следствие того, что $L^*_{\overline{x}}$ также непрерывно дифференцируема, при этом y-составляющая ее градиента в точке $\{\overline{x}, y(\overline{x})\}$ равна нулю.

Отметим следующее свойство функции $\varphi(x)$ в предположении, что функции $f_\tau(x, y)$ $(\tau = \overline{0, m})$ непрерывно дифференцируемы: *если в некоторой точке $\overline{x}$, являющейся внутренней точкой области определения, функции $\varphi(x)$, $y(\overline{x})$ и $u^*(\overline{x})$ определены однозначно, то в этой точке $\varphi(x)$ непрерывно дифференцируема. В противном случае в этой точке наблюдается разрыв градиента.*

В случае, когда функции $\varphi(x)$ и $f_i(x, y)$ непрерывно дифференцируемы по x, обобщенный градиент функции $\varphi(x)$ в точке $x = \overline{x}$ определяется по формуле

$$g_\varphi(\overline{x}) = \sum_{i=1}^{m} u_i(\overline{x})\, g^x_{f_i}(\overline{x}, y(\overline{x})) + g^x_{f_0}(\overline{x}, y(\overline{x})),$$

где $g^x_{f_i}(\overline{x}, y(\overline{x}))$ и $g^x_{f_0}(\overline{x}, y(\overline{x}))$ – соответственно проекции обобщенных градиентов функций $f_i(x, y)$ и $f_0(x, y)$ в точке $(\overline{x}, y(\overline{x}))$ на R^n_x.

Следует отметить, что не при всех $\overline{x}$ задача (3.40), (3.41) будет разрешима. Выход из таких ситуаций заключается в использовании метода "штрафных функций", при котором задача (3.40), (3.41) заменяется следующей задачей:

$$f_0(\overline{x}, y) + M \sum_{i=1}^{m} v_i \to \min,$$

$$f_i(\overline{x}, y) - v_i \leq 0 \quad (i = \overline{1, m}),$$

$$v_i \geq 0 \quad (i = \overline{1, m}),$$

где M – достаточно большое положительное число.

Легко видеть, что эта задача имеет допустимое решение при любом $\overline{x}$ с выполнением условия Слейтера.

Таким образом, если предположить, что задача (3.38), (3.39) сведена к такому виду, что функция $\varphi(x)$ определена для любого x и имеется возможность вычисления обобщенного градиента $g_\varphi(\overline{x})$ в произвольной точке $\overline{x}$, то получим итеративный алгоритм решения задачи (3.38), (3.39), на k-м шаге которого необходимо выполнить три этапа.

1. Решить задачу (3.40), (3.41) при $\overline{x} = \overline{x}^k$ и определить вектор множителей Лагранжа $u(\overline{x}^k) = \{u_i(\overline{x}^k)\}$.

2. Вычислить обобщенный градиент $g_\varphi(\overline{x})$ по формуле (3.42).

3. Определить новые значения

$$\overline{x}^{k+1} = \overline{x}^k - h_{k+1} g_\varphi(\overline{x}^k),$$

где h_{k+1} – величина шага.

Выбор последовательности $\{h_k\}$ проводится в соответствии с рекомендациями, приведенными в предыдущей главе.

2. Схема декомпозиции по переменным для сепарабельных задач выпуклого программирования. Пусть задача (3.38), (3.39) имеет специальный вид, например является задачей выпуклого программирования с сепарабельными функциями блочной структуры вида

$$\sum_{j=1}^{p} \Phi_j(x_j) + \alpha(y) \to \min, \tag{3.43}$$

$$W_j^i(x_j) + F_j^i(y) \le b_j^i \quad (i = \overline{1, m_j};\ j = \overline{1, p}) \tag{3.44}$$

или в векторной форме

$$\sum_{j=1}^{p} \Phi_j(x_j) + \alpha(y) \to \min,$$

$$W_j(x_j) + F_j(y) \le \overline{b}_j \quad (j = \overline{1, p}),$$

где в дополнении к обозначениям для задачи (3.22)–(3.24) $y \in R^k$, $\alpha(y)$ – выпуклая функция, $F_j(y) = \{F_j^i(y)\}_{i=1}^{m_j}$ – векторы выпуклых функций.

В данном случае фиксируем переменные $y = \overline{y}$, и тогда задача (3.43), (3.44) распадается на p задач для каждого блока в отдельности.

Таким образом, на $(k+1)$-м шаге метода ОГС решения задачи

$$\min_y \Big\{ \varphi(y) = \min_x \Big\{ \sum_{j=1}^{p} \Phi_j(x_j) + \alpha(y) \Big\} \Big\}$$

при фиксированном $y = y^k$ необходимо выполнить приведенные выше три этапа схемы декомпозиции по переменным.

1. Найти решение $\{x_j^*(y^k)\}$ задачи (3.43), (3.44) при фиксированном $y = y^k$ и определить значения множителей $u_j^i(x_j^*(y^k))$ $(i = \overline{1, m_j};\ j = \overline{1, p})$ функции Лагранжа

$$L(x, y, u) = \sum_{j=1}^{p} \Big[\Phi_j(x_j) + \sum_{i=1}^{m} u_j^i \cdot \big(W_j^i(x_j) + F_j^i(y) - b_j^i \big) \Big] + \alpha(y).$$

Задача (3.43), (3.44) при $y = y^k$ распадается на p задач выпуклого программирования вида

$$\Phi_j(x_j) \to \min F_j^i(y^k), \tag{3.45}$$

$$W_j^i(x_j) \le b_j^i - F_j^i(y^k) \quad (i = \overline{1, m_j}). \tag{3.46}$$

2. Вычислить обобщенный градиент функции $\varphi(y)$ в точке $y = y^k$ по формуле
$$g_\varphi^y = \sum_{j=1}^{p} \sum_{i=1}^{m_j} u_j^i(x_j^*(y^k)) \cdot g_{ij}^F(y^k) + g^\alpha(y^k),$$
где $g_{ij}^F(y^k)$ – обобщенный (или обычный) градиент функции $F_j^i(y)$ в точке $y = y^k$, а $g^\alpha(y^k)$ – обобщенный (или обычный) градиент функции $\alpha(y)$ в точке $y = y^k$.

3. Определить новые значения
$$y^{k+1} = y^k - h_{k+1} g_\varphi^y,$$
где h_{k+1} – величина шага которая выбирается в соответствие с правилами использования метода ОГС.

3. Схема декомпозиции по переменным для задач линейного программирования. Рассмотрим задачу линейного программирования: найти
$$\min(cx + dy) \tag{3.47}$$
при ограничениях
$$Ax + By \le b, \tag{3.48}$$
$$x \ge 0, \quad y \ge 0 \tag{3.49}$$
в которой x и y – векторы переменных размерностей n и k соответственно, c, d и b – векторы заданных чисел размерностей n, k и m соответственно, A и B – матрицы размерностей $m \times n$ и $m \times k$ соответственно.

В общем виде для задачи (3.47)–(3.49) можно рассматривать две схемы декомпозиции: по переменным x и по переменным y. Тогда при фиксировании одной или другой группы переменных получим две кусочно-линейные выпуклые функции $\varphi(x) = \min\limits_y(cx + dy)$ или $\psi(y) = \min\limits_x(cx + dy)$, которые соответствуют двум группам задач линейного программирования (прямой и двойственной):

$$\text{Задача I:} \quad \left\{ \begin{array}{cc} dy + c\overline{x} \to \min, & (b - A\overline{x}, v) \to \max, \\ By \le b - A\overline{x}, & B^T v \ge d, \\ y \ge 0; & v \ge 0; \end{array} \right.$$

$$\text{Задача II:} \quad \left\{ \begin{array}{cc} cx + d\overline{y} \to \min, & (b - B\overline{y}, v) \to \max, \\ Ax \le b - B\overline{y}, & A^T v \ge c, \\ x \ge 0; & v \ge 0 \end{array} \right.$$

в которых v – вектор двойственных переменных размерностей m; A^T и B^T – транспонированные матрицы A и B соответственно.

Для задачи линейного программирования (3.47)–(3.49) схема декомпозиции по переменным x применяется в том случае, когда задача I минимизации по y исходной линейной функции $cx + dy$ или максимизации по v двойственной функции $(b - A\overline{x}, v)$ решаются значительно проще чем первоначальная задача (3.47)–(3.49). И наоборот, схема декомпозиции по переменным y применяется когда задача II минимизации по x исходной линейной функции $cx + dy$ или максимизации по v двойственной функции $(b - B\overline{y}, v)$ решаются проще чем первоначальная задача (3.47)–(3.49).

Рассмотрим особенности применения схемы декомпозиции по переменным для решения задачи (3.47)–(3.49) в зависимости от фиксирования переменных x или y. Тогда, соответственно, получаем два варианта схемы декомпозиции по переменным.

I. При фиксированных переменных $\overline{x} = \overline{x}_t$ на $(t+1)$-м шаге метода ОГС необходимо:

- найти решения $y^*(\overline{x})$ и $v^*(\overline{x})$ пары прямо-двойственных задач I;

- определить значения обобщенного градиента $g_\varphi(\overline{x})$ функции $\varphi(\overline{x})$ в точке $\overline{x} = \overline{x}_t$ по формуле

$$g_\varphi(\overline{x}) = \left\{ g_j^\varphi(\overline{x}) = (-\alpha_j, v^*(\overline{x})) + c_j \quad (j = \overline{1, n}) \right\},$$

где c_j – j-й элемент вектора c, α_j – j-й столбец размерности m матрицы A, $g_j^\varphi(\overline{x})$ – j-й элемент вектора $g_\varphi(\overline{x})$ размерности n;

- определить значения

$$\overline{x}_{t+1} = \max\{0, \overline{x}_t - h_{t+1}\, g_\varphi(\overline{x})\},$$

где h_{t+1} – величина шага в методе ОГС.

II. При фиксированных переменных $\overline{y} = \overline{y}_t$ на $(t+1)$-м шаге метода ОГС необходимо:

- найти решения $x^*(\overline{y})$ и $v^*(\overline{y})$ пары прямо-двойственных задач II;

- определить значения обобщенного градиента $g_\psi(\overline{y})$ функции $\psi(\overline{y})$ в точке $\overline{y} = \overline{y}_t$ по формуле

$$g_\psi(\overline{y}) = \left\{ g_l^\psi(\overline{y}) = (-\beta_l, v^*(\overline{y})) + d_l \quad (l = \overline{1, k}) \right\},$$

где d_l – l-й элемент вектора d, β_l – l-й столбец размерности m матрицы B, $g_l^\psi(\overline{y})$ – l-й элемент вектора $g_\psi(\overline{y})$ размерности k;

- определить значения

$$\overline{y}_{t+1} = \max\{0, \overline{y}_t - h_{t+1}\, g_\psi(\overline{y})\},$$

где h_{t+1} – величина шага в методе ОГС.

4. Схема декомпозиции по переменным для задач линейного программирования блочно-диагональной структуры со связующими переменными. Наиболее типичен для схемы декомпозиции по переменным является случай, когда при фиксированных переменных $x = \overline{x}$ или $y = \overline{y}$ задача (3.47)–(3.49) (задача I или II) распадается на блоки, каждый из которых представляет собой задачу линейного программирования сравнительно небольшой размерности или некоторой известной задачей оптимизации специальной структуры.

Рассмотрим задачу линейного программирования блочно-диагональной структуры со связующими переменными: найти

$$\min\Big(\sum_{j=1}^{p} c_j x_j + dy\Big) \tag{3.50}$$

при ограничениях

$$B_j x_j + F_j y \le b_j \quad (j = \overline{1,p}), \tag{3.51}$$

$$y \ge 0, \quad x_j \ge 0 \quad (j = \overline{1,p}), \tag{3.52}$$

где c_j, x_j – n_j-мерные векторы; d, y – k-мерные векторы; b_j – m_j-мерные векторы; B_j – $m_j \times n_j$-мерные матрицы; F_j – $m_j \times k$-мерные матрицы. Задача (3.50)–(3.52) является общей задачей линейного программирования, для решения которой можно использовать известные методы. Однако в случае больших размерностей для ее решения эффективнее использовать схемы декомпозиции по переменным.

Зафиксируем переменные $y = \overline{y}$ и рассмотрим задачу

$$\Psi(\overline{y}) = \min_{x \in R(x,\overline{y})} \Big\{\sum_{j=1}^{p} c_j x_j + d\overline{y}\Big\}, \tag{3.53}$$

где $x = \{x_1, x_2, \ldots, x_p\}$; $R(x,\overline{y}) = R_1(x_1,\overline{y}) \times \cdots \times R_p(x_p,\overline{y})$; $R_j(x_j,\overline{y}) = \{x_j\colon B_j x_j \le b_j - F_j\overline{y};\ x_j \ge 0\}$, $j = \overline{1,p}$.

Функция $\Psi(\overline{y})$ является кусочно-линейной выпуклой функцией. Построим функцию Лагранжа для задачи (3.50)–(3.52)

$$L(x,y,u) = \sum_{j=1}^{p} c_j x_j + dy + \sum_{j=1}^{p} (u_j, B_j x_j + F_j y - b_j),$$

где $u = \{u_1, \ldots, u_p\}$ – множители Лагранжа ограничений (3.51). Пусть $x^*(y)$ – оптимальное решение задачи (3.53) при фиксированном $y = \overline{y}$, а $u^*(\overline{y})$ – значения двойственных оценок данного решения. Тогда обобщенный градиент функции $\Psi(y)$ в точке $\overline{y}$ определяется по формуле

$$g^y_{L^*_{\overline{y}}}(x(\overline{y}), \overline{y}) = d + \sum_{j=1}^{p} F_j^T u_j^*(\overline{y}).$$

В силу сепарабельности целевой функции задачи (3.53) прямое и двойственное решения $x^*(\overline{y})$ и $u^*(\overline{y})$ находятся из решения следующих p прямых задач:

$$c_j x_j \to \min,$$

$$B_j x_j \le b_j - F_j \overline{y}, \quad x_j \ge 0,$$

и p двойственных задач:

$$q_j(\overline{y}) u_j = (b_j - F_j \overline{y},\, u_j) \to \max,$$

$$B_j^T u_j \ge c_j, \quad u_j \ge 0,$$

где $q_j(\overline{y}) = b_j - F_j \overline{y}$.

Обозначим множество допустимых решений двойственных задач при фиксированном $\overline{y}$ через

$$V_j(u_j) = \{u_j \colon B_j^T u_j \ge c_j, \quad u_j \ge 0\},$$

которые не зависят от переменных y.

Тогда итеративный алгоритм решения задачи (3.50)– (3.52), построенный на основе схемы декомпозиции по переменным и одного из методов негладкой оптимизации состоит в следующем: пусть, начиная с некоторого значения y^0, имеем на k-й итерации точку y^k, тогда на $(k+1)$-й итерации необходимо выполнить следующие три основных этапа.

1. Решить задачу линейного программирования для каждого блока в отдельности при фиксированных $\overline{y} = y^k$, т.е. найти $x_j^*(y^k)$ и $u_j^*(y^k)$, для которых

$$\begin{aligned} c_j x_j^*(y^*) &= \min_{x_j \in R_j(x_j, y^k)} c_j x_j = \\ &= \max_{u_j \in V_j(u_j)} (b_j - F_j y^k, u_j) = q_j(y^k) u_j^*(y^k) \quad (j = \overline{1, p}). \end{aligned}$$

2. Вычислить значение обобщенного градиента функции $\Psi(y)$ в точке y^k по формуле

$$g(u(y^k)) = d + \sum_{j=1}^{p} F_j^T u_j^*(y^k).$$

3. Вычислить y^{k+1} по формуле

$$y^{k+1} = \max\{0, y^k - h_{k+1} g(u(y^k))\},$$

где h_{k+1} – величина шага.

3.6. Схема декомпозиции по ресурсам

Метод Корнаи-Липтака [630] известен как схема декомпозиции на двух уровнях, который первоначально был разработан для решения задач линейного программирования при оптимальном распределение глобальных и эффективном использование локальных ресурсов. Далее данный метод был обобщен для задач выпуклого и нелинейного программирования как общую схему декомпозиции по ресурсам [633, 733].

Основной особенностью этого метода является процедура перераспределения ресурсов, для чего были использованы различные приемы и методы, такие как методы теории игр, задачи двойственного программирования для определения двойственных оценок глобальных ресурсов и методы недифференцируемой оптимизации. В данном параграфе приводится схема декомпозиции по ресурсам для решения задач выпуклого программирования, в которой задача перераспределения ресурсов решается субградиентными методами. Далее такая схема декомпозиции применяется для решения задач линейного и дробно-линейного программирования блочно-диагональной структуры.

1. Схема декомпозиции по ресурсам для задач выпуклого программирования. Пусть $x_1, x_2, \ldots, x_p$ ($x_j \in R^{n_j}$, $j = \overline{1,p}$) – группы переменных, образующих вектор $x = (x_1, x_2, \ldots, x_n) \in R^n$, где $n = \sum_{j=1}^{p} n_j$. Рассмотрим задачу выпуклого программирования блочно-диагональной структуры со связующими ограничениями и p диагональных блоков следующего вида:

$$\sum_{j=1}^{p} \Phi_j(x_j) \to \min, \tag{3.54}$$

$$\sum_{j=1}^{p} F_j^i(x_j) \le b \quad (i = \overline{1,m}), \tag{3.55}$$

$$W_j(x_j) \le \overline{b}_j \quad (j = \overline{1,p}). \tag{3.56}$$

Здесь Φ_j, F_j^i – выпуклые функции; W_j – выпуклые вектор-функции от соответствующих переменных; $b = (b_1, b_2, \dots, b_m)$ – фиксированный вектор глобальных ресурсов; $\overline{b}_j$ – вектор локальных ресурсов соответствующей размерности.

Рассмотрим для задачи (3.54)–(3.56) функцию Лагранжа (относительно всех ограничений)

$$L(x,\lambda,\mu) = \sum_{j=1}^{p} \Phi_j(x_j) + \sum_{j=1}^{p}\sum_{i=1}^{m} \lambda^i \Big(F_j^i(x_j) - b_i\Big) + \sum_{j=1}^{p} \mu_j \Big(W_j(x_j) - \overline{b}_j\Big),$$

в которой $\lambda = \{\lambda^i \quad (i = \overline{1,m})\}$ и $\mu = \{\mu_j \quad (j = \overline{1,p})\}$ – множители Лагранжа соответственно для ограничений (3.55) и (3.56).

Сущность метода Корнаи-Липтака заключается в осуществление первоначального распределения глобальных ресурсов b_i между всеми p блоками с последующей оценке эффективности такого распределения (использования глобальных и локальных ресурсов) через двойственных оценок (множителей Лагранжа). Таким образом оценки $\lambda^i \quad (i = \overline{1,m})$ характеризуют эффективность использования глобальных ресурсов, одновременно с имеющимися локальными ресурсами. Как видно из функции Лагранжа $L(x,\lambda,\mu)$ оценки λ^i являются одинаковыми для каждого блока $j = \overline{1,p}$, поэтому основная цель перераспределения глобальных ресурсов является получение одинаковых оценок их эффективного использования для каждого блока в отдельности.

Представим вектор ресурсов b в виде суммы:

$$b = z_1 + z_2 + \dots + z_p, \tag{3.57}$$

а ограничения (3.55) заменим на

$$F_j^i(x_j) \le z_j^i \quad (i = \overline{1,m};\; j = \overline{1,p}), \tag{3.58}$$

где $z_j = (z_j^1, z_j^2, \dots, z_j^m)$ – вектор, показывающий распределение ресурсов j-му блоку.

Используя данные соотношения, задача (3.54)-(3.56) разлагается на p подзадач (локальные задачи):

$$\Phi_j(x_j) \to \ \min, \tag{3.59}$$

$$F_j^i(x_j) \ \le z_j^i \quad (i = \overline{1,m}), \tag{3.60}$$

$$W_j(x_j) \ \le \bar{b}_j. \tag{3.61}$$

Для каждой задачи (3.59)–(3.61) построим свою функцию Лагранжа следующего вида

$$L_j(x_j, \lambda_j, \mu_j) = \Phi_j(x_j) + \sum_{i=1}^{m} \lambda_j^i \Big(F_j^i(x_j) - z_j^i\Big) + \mu_j \Big(W_j(x_j) - \bar{b}_j\Big),$$

в которой λ_j^i $(i = \overline{1,m})$ – множители Лагранжа ограничений (3.60).

С учетом сделанных распределений (3.57) и полученных новых подзадач (3.59)-(3.61) определим следующую задачу:

$$\Phi(z) = \sum_{j=1}^{p} \alpha_j(z_j) \to \min,$$

$$\sum_{j=1}^{p} z_j^i \le b_i \quad (i = \overline{1,m}),$$

в которой $\alpha_j(z_j)$ – алгоритмическая функция, значения которой определяется в результате решения задачи (3.59)-(3.61) с учетом решения $x_j^*(z_j)$ и распределения z_j. Для этой задачи функция Лагранжа имеет вид:

$$L(z, \lambda) = \sum_{j=1}^{p} \left(\alpha_j(z_j) + \sum_{i=1}^{m} \lambda^i z_j^i\right) - \sum_{i=1}^{m} \lambda^i b_i.$$

Пусть $\Phi(z) = \sum\limits_{j=1}^{p} \Phi_j(x_j^*(z_j))$, где $x_j^*(z_j)$ – оптимальное решение задачи (3.59)–(3.61) при заданном z_j (фиксированное распределение ресурсов); для каждой из локальных задач выполняется условие Слейтера; $\{\lambda_j^i(z_j)\}$ – оптимальные значения множителей Лагранжа, соответствующие ограничениям (3.60). Легко показать, что $\Phi(z)$ – выпуклая функция, а ее субградиент определяется по формуле

$$g_\Phi(z) = \left\{-\lambda_j^i(z_j) \quad (j = \overline{1,p};\ i = \overline{1,m})\right\}.$$

Первоначальная задача (3.54)–(3.56) сводится к нахождению минимума выпуклой функции $\Phi(z)$ при линейных ограничениях (3.57) (координирующая задача), т.е.

$$\Phi(z) \ \to \min, \tag{3.62}$$

$$\sum_{j=1}^{p} z_j \leq b. \tag{3.63}$$

Учитывая простоту этих ограничений, для решения координирующей задачи можно применить обобщенный градиентный спуск с проектированием на линейные многообразия.

Таким образом, исходная задача (3.54)–(3.56) больших размерностей и блочно-диагональной структуры сводится с помощью двухуровневого приема к решению координирующей (внешней) задачи (3.62), (3.63) и локальных (внутренних) задач (3.59)–(3.61) по следующей схеме декомпозиции по ресурсам с применением методов недифференцируемой оптимизации, вычисления по которой проводится на двух уровнях.

На первом уровне проводится перераспределение ресурсов b путем решения задачи недифференцируемой оптимизации одним из субградиентным методом. Таким образом, на каждом шаге субградиентного метода решения задачи (3.62), (3.63) на множестве вектора переменных $z = (z_1, z_2, \ldots, z_p)$ размерностью $m \times p$ осуществляется новое распределение глобальных ресурсов.

На втором уровне осуществляется оценка проведенного перераспределения ресурсов путем решения локальных задач (3.59)–(3.61) на множестве переменных (x_j, λ_j).

Таким образом, на $(k+1)$-м шаге субградиентного метода решения задачи (3.62), (3.63) необходимо выполнить три этапа.

1. Решить задачи (3.59)–(3.61) при фиксированных $z_j^i = z_j^{i^k}$ и найти значения двойственных оценок $\lambda_j^i(z_j^k)$.

2. Найти значения обобщенного градиента по формуле $g_j^{i^k}(z) = -\lambda_j^{i^k}(z_j^{i^k})$ $(j = \overline{1,p};\ i = \overline{1,m})$.

3. Найти новую точку

$$\widetilde{z}_j^{i^{k+1}} = \max\{0, z_j^{i^k} - h_{k+1} g_j^{i^k}(z)\} =$$

$$= \max\{0, z_j^{i^k} + h_{k+1} \lambda_j^{i^k}(z_j^{i^k})\} \quad (j = \overline{1,p};\ i = \overline{1,m}).$$

Провести проектирование точки $\widetilde{z}_j^{i^{k+1}}$ на линейном многообразии $\sum\limits_{j=1}^{p} z_j = b$, т.е. найти новые значения $z_j^{i^{k+1}}$ в соответствии с правилом

$$z_j^{i^{k+1}} = \min\left\{\widetilde{z}_j^{i^{k+1}}, b_i - \sum_{r\neq j}\widetilde{z}_r^{i^{k+1}}\right\} \quad (j=\overline{1,p};\ i=\overline{1,m}).$$

2. Схема декомпозиции по ресурсам для задач линейного программирования. Для задачи линейного программирования блочно-диагональной структуры декомпозиционная схема по ресурсам имеет такую же структуру. В данном случае изменяются только локальные задачи (3.59)–(3.61). Пусть имеем задачу линейного программирования (3.32)–(3.35). Тогда в схеме декомпозиции по ресурсам координирующей является задача (3.62), (3.63), а локальные имеют вид задач линейного программирования с фиксированными значениями переменных z_j:

$$\Phi_j(x_j) = c_j x_j \to \min,$$

$$B_j x_j \le z_j,$$

$$x_j \ge 0.$$

Находим оптимальные решения $x_j^*(z_j)$ и двойственные оценки $\lambda_j(z_j)$ эффективности использования распределенных глобальных ресурсов z_j. Тогда

$$\Phi(z) = \sum_{j=1}^{p}\Phi_j(x_j^*(z_j)) = \sum_{j=1}^{p} c_j x_j^*(z_j)$$

является кусочно-линейной выпуклой функцией по переменным z, а ее обобщенный градиент определяется в соответствии с функцией Лагранжа

$$L(x,z) = \sum_{j=1}^{p}\Big[(c_j + \lambda_j A_j + \beta_j B_j)x_j - (\lambda_j z_j + \beta_j b_j)\Big]$$

и в данном случае имеет вид $g_\Phi(z) = \{-\lambda_j\}_{j=1}^{p}$. Остальные вычисления соответствуют указанных выше основных трех этапов. Отличие состоит в том, что задачи (3.59)–(3.61) являются линейными.

3.7. Применение схем декомпозиции для решения специальных задач

1. Транспортные задачи. Рассмотрим обычную линейную транспортную задачу:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min, \tag{3.64}$$

$$\sum_{j=1}^{n} x_{ij} \le a_i \quad (i = \overline{1,m}), \tag{3.65}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,n}), \tag{3.66}$$

$$x_{ij} \ge 0 \quad (i = \overline{1,m};\ j = \overline{1,n}), \tag{3.67}$$

в которой m – количество поставщиков, n – количество потребителей, a_i – объемы поставки, b_j – объемы потребления, c_{ij} – расстояния перевозки, а x_{ij} – объемы перевозки между i-м поставщиком и j-м потребителем.

Предположим, что a_i и b_i – положительные числа и $\sum\limits_{i=1}^{m} a_i \ge \sum\limits_{j=1}^{n} b_j$.

Для решения задачи (3.64)–(3.67) можно применить декомпозиционные схемы и субградиентные методы. В случае, когда $m \ll n$, а n достаточно большое число, то имеем так называемую ленточную матрицу $C = \{c_{ij}\}$, которая представляет собой матрицу расстояний от небольшого числа поставщиков к большому числу потребителей. В таком случае получаем транспортную задачу большой размерности, для решения которой можно эффективно применить декомпозиционные схемы и субградиентные методы.

Схема декомпозиции по ограничениям. Рассмотрим для начала схему декомпозиции по ограничениям. Для этого построим функцию Лагранжа для задачи (3.64)–(3.67), на базе ограничений (3.65).

$$L(x,u) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\,x_{ij} + \sum_{i=1}^{m} u_i\left(\sum_{j=1}^{n} x_{ij} - a_i\right) =$$
$$= \sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij} + u_i)x_{ij} - \sum_{i=1}^{m} a_i u_i$$

и рассмотрим задачу

$$L^*(u) = \min_{x \in R(x)} L(x,u) \tag{3.68}$$

в которой $R(x)$ – ограниченное выпуклое многогранное множество значений переменных x, удовлетворяющих ограничениям (3.66), (3.67), а $u = (u_1, u_2, \ldots, u_m)$ – множители Лагранжа для ограничений (3.65), $u \geq 0$.

Для решения задачи (3.64)–(3.67) применяем следующий алгоритм.

1. Решаем внешнюю задачу максимизации функции $L^*(u)$ при ограничениях $u \geq 0$ субградиентным методом, на $(t+1)$-м шаге которого необходимо выполнить основные три этапа.

а) Решить внутреннюю задачу (3.68) при фиксированных значениях $u = u^t$, следующего вида

$$\sum_{j=1}^{n} \sum_{i=1}^{m} (c_{ij} + u_i^t) x_{ij} \to \min, \tag{3.69}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,n}), \tag{3.70}$$

$$x_{ij} \geq 0, \tag{3.71}$$

которая имеет сепарабельный функционал (3.69) и независимые (блочные) ограничения (3.70) для каждого j. Тогда решение задачи (3.69)–(3.71) однозначно определяется по формуле

$$x_{ij}^*(u^t) = \begin{cases} b_j, & \text{для } i = i_j^* \quad (j = \overline{1,n}); \\ 0, & \text{для } i \neq i_j^* \quad (j = \overline{1,n}), \end{cases}$$

где i_j^* индекс для которого имеем

$$c_{i_j^* j} + u_{i_j^*}^t = \min_i (c_{ij} + u_i^t);$$

б) Вычислить значения субградиента $g_i(u^*)$ функции $L^*(u)$ в точке $u = u^t$ по формуле

$$g_i(u^t) = \sum_{j=1}^{n} x_{ij}^*(u^t) - a_i \quad (i = \overline{1,m}),$$

где $\{x_{ij}^*(u^t)\}$ оптимальное решение задачи (3.69)–(3.71) при $u = u^t$;

в) Найти новые значения

$$u_i^{t+1} = \max\left\{0, u_i^t + h_{t+1}\, g_i(u^t)\right\} \quad (i = \overline{1,m}),$$

где h_{t+1} – величина шага в субградиентном методе.

2. Пусть u^* – оптимальное решение задачи максимизации $L^*(u)$, а x^* – оптимальное решение задачи (3.69)–(3.71) при $u = u^*$. Находим значения

$$\Delta_{ij} = c_{kj} + u_k^* - c_{ij} - u_i^*,$$

для всех i и j, для которых $x_{ij}^* = 0$, где k – значение индексов i, для которых $x_{kj} = b_j \quad (j = \overline{1,n})$.

Параметры Δ_{ij} соответствуют по своей структуре признаку оптимальности плана x транспортной задачи в методе потенциалов, а именно

$$\Delta_{ij} = v_j - u_i - c_{ij} \leq 0,$$

в котором в данном случае $v_j = c_{kj} + u_k^*$.

3. Находим множество J тех индексов j, для которых хотя бы для одного $i \quad |\Delta_{ij}| \leq \varepsilon$, где $\varepsilon > 0$ – достаточно малое число. Множество J соответствует тем значениям индексов j, для которых объемы перевозок фиксированы, т.е. имеем $x_{kj}^* = b_j$, но имеются некоторые другие возможные варианты поставки с примерно такими же оценки эффективности, т.е. $c_{kj} + u_k^* \approx c_{ij} + u_i$, для некоторых индексов i.

4. Находим новые значения коэффициентов a_i', соответствующие нераспределенных объемов поставки

$$a_i' = a_i - \sum_{j \notin J} x_{ij}^*$$

и находим множество I тех значений индексов i для которых $a_i' > 0$. Множество I соответствует тем поставщикам, которые имеют нераспределенные объемы грузов, т.е. a_i' – положительные числа.

5. По алгоритму методов потенциалов решаем редуцированную транспортную задачу

$$\sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \to \min;$$

$$\sum_{j \in J} x_{ij} \leq a_i', \quad i \in I;$$

$$\sum_{i \in I} x_{ij} = b_j, \quad j \in J;$$

$$x_{ij} \geq 0, \quad i \in I, j \in J,$$

для которой $||I|| \approx ||J||$, т.е. ее матрица расстояний является почти квадратной и небольших размерностей.

Полученное решение редуцированной задачи дополняет оптимальное решение x^* задачи (3.69)–(3.71) для тех индексов i и j для которых выполнялись неравенства $|\Delta_{ij}| \leq \varepsilon$.

Схема декомпозиции по переменным. Рассмотрим теперь особенности применения схемы декомпозиции по переменным для решения транспортной задачи (3.64)–(3.67). Для этого построим ее двойственную

$$\sum_{j=1}^{n} b_j v_j - \sum_{i=1}^{m} a_i u_i \to \max, \tag{3.72}$$

$$v_j - u_i \leq c_{ij} \quad (i = \overline{1,m};\ j = \overline{1,n}), \tag{3.73}$$

$$u_i \geq 0 \quad (i = \overline{1,m}), \tag{3.74}$$

где u_i, v_j – двойственные переменные (потенциалы), соответствующие ограничениям (3.65) и (3.66). При фиксированных u_i, каждое v_j ограничено сверху соотношениями

$$v_j \leq c_{ij} + u_i \quad (i = \overline{1,m}).$$

Так как $b_j > 0$, то для получения максимума по v_j в (3.72) при фиксированных u_i должно выполняться соотношение

$$v_j = \min_{1 \leq i \leq m} (c_{ij} + u_i). \tag{3.75}$$

Такими соотношениями должны быть связаны потенциалы $u_i \ (i = \overline{1,m})$ и $v_j \ (j = \overline{1,n})$ в оптимальном плане двойственной транспортной задачи, при этом все ограничения (3.73) выполняются автоматически. Отсюда вытекает, что двойственная задача (3.72)–(3.74) сводится к следующей задаче безусловной оптимизации: найти

$$\max_{u \geq 0} \varphi(u) = \max_{\{u_i\}} \left(\sum_{j=1}^{n} b_j \min_{1 \leq i \leq m} (c_{ij} + u_i) - \sum_{i=1}^{m} a_i\, u_i \right), \tag{3.76}$$

где $\varphi(u)$ – функция полученная в результате подстановки в (3.72) вместо v_j выражение (3.75), т.е. $\varphi(u) = \sum_{j=1}^{n} b_j \min_{1 \leq i \leq m} (c_{ij} + u_i) - \sum_{i=1}^{m} a_i u_i$.

Функция $\varphi(u)$ определена для любого $u \geq 0$, кусочно-линейна и вогнута. Разрыв градиента функции $\varphi(u)$ соответствуют разрывом градиентов одной из функций $\psi_j(u) = \min_{1 \leq i \leq m} (c_{ij} + u_i)$, которые происходят при таких значениях u, когда минимум в выражении для $\psi_j(u)$ достигается сразу при двух и большем числе значений i.

В результате получаем следующий алгоритм схемы декомпозиции по переменным решения транспортной задачи (3.72)–(3.74).

1. Решаем внешнюю задачу максимизации функции $\varphi(u)$ при $u \geq 0$ субградиентным методом на $(t+1)$-м шаге которого необходимо выполнить основные три этапа:

а) решить внутреннюю задачу

$$\sum_{j=1}^{n} b_j v_j \to \max, \tag{3.77}$$

$$v_j \leq c_{ij} + u_i \tag{3.78}$$

при фиксированных $u_i = \overline{u}_i^t$, решение которой определяется однозначно

$$v_j^t = \min_{1 \leq i \leq m} (c_{ij} + \overline{u}_i^t) = c_{i_j^* j} + \overline{u}_{i_j^*}^t,$$

для которого соответствует решение прямой задачи

$$x_{i_j^* j}^*(u^t) = \begin{cases} b_j, \text{ для } i = i_j^* \quad (j = \overline{1,n}), \\ 0, \quad \text{для } i \neq i_j^* \quad (j = \overline{1,n}); \end{cases}$$

б) вычислить значения субградиента $g_i(\overline{u}^t)$ функции $\varphi(u)$ в точке $u = \overline{u}^t$ по формуле

$$g_i(\overline{u}^t) = \sum_{j=1}^{n} x_{ij}^*(\overline{u}^t) - a_i \quad (i = \overline{1,m}),$$

где $\{x_{ij}^*(\overline{u}^t)\}$ оптимальное решение задачи (3.77), (3.78) при $u = \overline{u}^t$;

в) найти новые значения

$$\overline{u}_i^{t+1} = \max\left\{0, \overline{u}_i^t + h_{t+1} g_i(\overline{u}^t)\right\} \quad (i = \overline{1,m}),$$

где h_{t+1} – величина шага в субградиентном методе.

2. Пусть u^* оптимальное решение задачи (3.76), а v^* решение задачи (3.77), (3.78) при $u = u^*$. Находим значения

$$\Delta_{ij} = v_j^* - u_i^* - c_{ij} = c_{kj} + u_k^* - u_i^* - c_{ij}$$

для всех i и j, для которых $x_{ij}^* = 0$, где k – значения индексов, для которых $x_{kj} = b_j$ $(j = \overline{1,n})$, т.е. $v_j^* = c_{kj} + u_k^*$.

3. Находим множество J тех индексов j, для которых $|\Delta_{ij}| \leq \varepsilon$, где $\varepsilon > 0$ – достаточно малое число. Множество J соответствует тем значениям j, для которых ограничения (3.78) ε-неустойчивы, т.е. ограничения являются существенными.

4. Находим новые значения коэффициентов a'_i определенных по формулам

$$a'_i = a_i - \sum_{j \notin J} x^*_{ij} \quad (i = \overline{1, m})$$

и находим множество I тех значений индексов i для которых $a'_i > 0$.

5. По алгоритму метода потенциалов решаем редуцированную прямую транспортную задачу

$$\sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \to \min;$$

$$\sum_{j \in J} x_{ij} \leq a'_i, \quad i \in I;$$

$$\sum_{i \in I} x_{ij} = b_j, \quad j \in J;$$

$$x_{ij} \geq 0, \qquad i \in I,\, j \in J,$$

для которой $||I|| \approx ||I||$, т.е. ее матрица расстояний является почти квадратной.

Полученное решение редуцированной задачи дополняет оптимальное решение задачи (3.77), (3.78).

Следует отметить, что схема декомпозиции по ограничениям и схема декомпозиции по переменным для решения транспортной задачи совпадает со схемой декомпозиции по переменным для двойственной ей задачи. В результате их применения получаем те же решения u^* и $\overline{u}^*$, для которых значение функционала совпадает $L^*(u^*) = \varphi(\overline{u}^*) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x^*_{ij}$.

Задача (3.76), вытекающая из транспортной задачи, фактически являлась той самой первой моделью, на основе которой академик Н.З. Шор отрабатывал методы субградиентного спуска и показал их практическую эффективность при решении задач большой размерности [725].

Особенности применения такого метода при решении сетевой транспортной задачи состоит в том, что вместо исходной матрицы расстояний

$\{c_{ij}\}$, на каждом шаге субградиентного метода находятся кратчайшие пути с учетом пересчета длин дуг на соответствующую величину u_i.

2. Обобщенная распределительная задача линейного программирования. Обобщенная распределительная задача линейного программирования, которая имеет собственную терминологию, это задача следующего вида: найти

$$\min \sum_{(i,j)\in P} c_{ij} x_{ij} \tag{3.79}$$

при ограничениях

$$\sum_{(i,j)\in P} p_{ij}^{(\alpha)} x_{ij} - T^{(\alpha)} \le 0, \quad \alpha \in A; \tag{3.80}$$

$$\sum_{\{j/(i,j)\in P\}} x_{ij} = d_i, \quad i \in I = \{1, \dots, N\}; \tag{3.81}$$

$$x_{ij} \ge 0, \quad \forall\ (i,j) \in P. \tag{3.82}$$

Каждый индекс i соответствует индексу "продукта", индекс j является индексом "технологии", множество P это множество пар (i,j), которые соответствуют i-му "продукту" произведенным j-м "технологией" . Множество I будем называть множеством заказов, а множество A назовем множество технологических ограничений (технологические коэффициенты $p_{ij}^{(\alpha)}$ означают количество ресурса $\alpha \in A$ необходимое для производства единицы i-го продукта j-м технологией). Каждый заказ из ограничений (3.81) соответствует части ограничений (3.82), формирует блок, который ассоциируется с вектором $x_i = \{x_{ij}/(i,j) \in P\}$, а технологические ограничения (3.80) являются связующими ограничениями.

Рассмотрим схему декомпозиции по ограничениям, для чего построим функцию Лагранжа следующим образом:

$$L(x,u) = \sum_{(i,j)\in P} \Big(c_{ij} + \sum_{\alpha\in A} u_\alpha p_{ij}^{(\alpha)}\Big) x_{ij} - \sum_{\alpha\in A} u_\alpha T^{(\alpha)} \tag{3.83}$$

и получаем внутреннюю задачу: найти

$$\Psi(u) = \sum_{i=1}^{N} \psi_i(u) - \sum_{\alpha} u_\alpha T^{(\alpha)}, \tag{3.84}$$

где для каждого $i \in I$ имеем

$$\psi_i(u) = \min \sum_{\{j/(i,j)\in P\}} \Big(c_{ij} + \sum_{\alpha\in A} p_{ij}^{(\alpha)} u_\alpha\Big) x_{ij} \tag{3.85}$$

при ограничениях

$$\sum_{\{j/(i,j)\in P\}} x_{ij} = d_i, \quad x_{ij} \geq 0 \ \forall j, \ (i,j) \in P. \tag{3.86}$$

Пусть $x_{ij}(u)$ – решение (в случае неоднозначности, произвольное) задачи (3.85), (3.86). Легко видеть, что $x_{ij}(u) = \{x_{ij}(u)\}$ можно определить формулами:

$x_{ij}(u) = d_i$, если $j = j_i^*(u)$, где $j_i^*(u)$ некоторый (например, наименьший) индекс минимизации по j

$$c_{ij}(u) = c_{ij} + \sum_{\alpha\in A} u_\alpha p_{ij}^{(\alpha)};$$

$x_{ij}(u) = 0$ для всех $j \neq j_i^*(u)$.

Тогда,

$$\psi_i(u) = d_i \min_{\{j,(i,j)\in P\}} \Big[c_{ij} + \sum_{\alpha\in A} u_\alpha p_{ij}^{(\alpha)}\Big] \quad (i = \overline{1,N}\,)$$

и

$$\Psi(u) = \sum_{i=1}^{N} d_i \min_{\{j,(i,j)\in P\}} \Big[c_{ij} + \sum_{\alpha\in A} u_\alpha p_{ij}^{(\alpha)}\Big] - \sum_{\alpha\in A} u_\alpha T^{(\alpha)} \text{ для } u \geq 0;$$

$$g_\Psi(u) = \{g_\Psi^{(\alpha)}(u)\}_{\alpha\in A} = \Big\{\sum_{(i,j)\in P} x_{ij}(u) p_{ij}^{(\alpha)} - T^{(\alpha)}\Big\}_{\alpha\in A} =$$

$$= \Big\{\sum_{i\in I} d_i p_{ij_i^*}^{(\alpha)}(u) - T^{(\alpha)}\Big\}_{\alpha\in A}.$$

Задачи типа (3.79)–(3.82) часто встречаются на практике, в частности при решении задач производственно-транспортного планирования.

3. Решение нелинейных задач большой размерности специальной структуры. Эффективность недифференцируемой оптимизации покажем при решении нелинейных выпуклых задач большой размерности специальной структуры. В таких случаях структура включает только ”линейные” переменные, которые рассматриваются как сепарабельные для большой части всего множества переменных. Поэтому естественно использовать схемы декомпозиции по переменных разделенные на ”линейные” и ”нелинейные” части в зависимости от ситуации.

Рассмотрим задачу выпуклого программирования:
минимизировать
$$f(x,y) = \{(c,x) + f(y)\} \tag{3.87}$$

при ограничениях

$$Ax + b(y) \le 0,\ b(y) = \{b_i(y)\}_{i=1}^m, \tag{3.88}$$

$$x \ge 0, \tag{3.89}$$

где $y \in E^k$, $x \in E^n$, c – n-мерный вектор, A – матрица размерностей $m \times n$, а f и b_i являются выпуклыми функциями, определенные на E^k. Для простоты, предположим, что внутренняя задача: найти

$$\Psi(\overline{y}) = \inf_x [(c,x) + f(\overline{y})] \tag{3.90}$$

при ограничениях
$$Ax + b(\overline{y}) \le 0 \tag{3.91}$$

имеет непустое множество $X(\overline{y})$ оптимальных решений при произвольном $\overline{y} \in E^k$. Пусть $u \in E^n$ является вектором множителей Лагранжа для задачи (3.87)–(3.89). Для функции Лагранжа вида:

$$L(x,y,u) = (c,x) + f(y) + (u,Ax) + (u,b(y))$$

определяем
$$L_{\overline{y}}(x,u) = L(x,\overline{y},u)$$

для фиксированном $y = \overline{y}$, которая является функцией Лагранжа для задачи (3.90), (3.91). Пусть $\{x(\overline{y}) \in X(\overline{y}),\ u(\overline{y}) \in U(\overline{y})\}$ является пара оптимальных прямых и двойственных переменных, формирующие точку Куна-Таккера для задачи линейного программирования (3.90), (3.91). Тогда
$$\Psi(\overline{y}) = (c, x(\overline{y})) + f(\overline{y}),$$

где $\Psi(\overline{y})$ является выпуклой функцией, определенной на E^k. Субградиент для Ψ в точке $\overline{y}$ получим по формуле

$$g_{\Psi}(\overline{y}) = g_f(\overline{y}) + (u(\overline{y}), g_b(\overline{y})),$$

где $g_b(y) = \{g_{b_1}(y), \ldots, g_{b_m}(y)\}$. Функция Ψ является дифференцируемой почти везде, за исключением точек, где f или некоторые $b_i(y)$ являются недифференцируемые, и точки $\overline{y}$, где $u(\overline{y})$ неединственное.

Для нахождения оптимального y используется метод субградиентного типа недифференцируемой оптимизации на $(t+1)$-м шаге которого необходимо выполнить три этапа.

1. При фиксированных значениях $y = y^t$ решить задачу линейного программирования по переменным x

$$(c, x) \to \min,$$

$$Ax \le b(y^t),$$

$$x \ge 0$$

и найти ее оптимальные прямые и двойственные решения $x(y^t)$ и $u(y^t)$.

2. Найти значения обобщенного градиента $g_\psi(y^t)$ функции $\psi(y)$ в точке $y = y^t$ по формуле

$$g_\psi(y^t) = g_f(y^t) + (u(y^t), g_b(y^t)).$$

3. Найти новые значения

$$y^{t+1} = y^t - h_{t+1} \cdot g_\psi(y^t),$$

где h_{t+1} – величина шага в субградиентном методе.

3.8. Блочная задача сепарабельного выпуклого программирования

Рассмотрим задачу выпуклого программирования, в которой система ограничений имеет блочно-диагональную структуру, а функционал является сепарабельной функцией

$$F_1(x) = \sum_{j=1}^{p} \varphi_j(x_j) \to \min, \tag{3.92}$$

$$\sum_{j=1}^{p} g_j^i(x_j) \le a_i \quad (i = \overline{1, m}), \tag{3.93}$$

$$h_j^l(x_j) \le b_j^l \quad (l = \overline{1, m_j};\ j = \overline{1, p}). \tag{3.94}$$

Переменные данной задачи разбиты на p групп $x_j = (x_j^1, x_j^2, \ldots$ $\ldots, x_j^{n_j}) \in R^{n_j}$ $(j = \overline{1,p})$, которые образуют вектор $x = (x_1, x_2, \ldots, x_p) \in$ $\in R^n$, где $n = \sum\limits_{j=1}^{p} n_j$. На множестве каждой группы переменных x_j определены непрерывные функции $\varphi_j,\ g_j^i\ \ (i = \overline{1,m}),\quad h_j^l\ \ (l = \overline{1,m_j}) :$ $R^{n_j} \to R$.

Предположим, что функции $\varphi_j,\ g_j^i$ и h_j^l выпуклы на группе переменных $x_j\quad (j = \overline{1,p})$, а $a_i\quad (i = \overline{1,m}),\ \ b_j^l\quad (l = \overline{1,m_j};\ j = \overline{1,p})$ – заданные числа.

В задаче (3.92)–(3.94) можно использовать и другие сепарабельные функционалы:

$$F_2(x) = \max_j \varphi_j(x_j) \tag{3.95}$$

или же

$$F_3(x) = \prod_{j=1}^{p} \varphi_j(x_j). \tag{3.96}$$

Приведенную ниже схему решения задачи (3.92)–(3.94) можно с таким же успехом применять и для минимизации функционала (3.95) или (3.96) при ограничениях (3.93), (3.94).

Предположим, что функции $\varphi_j(x_j)$ такие, что функционалы $F_1(x)$, $F_2(x)$ и $F_3(x)$ являются выпуклыми или вогнутыми, без уточнения необходимых условий, которым они должны удовлетворять.

В общем случае задачи минимизации функционалов $F_1(x)$, $F_2(x)$ и $F_3(x)$ при ограничениях (3.93) и (3.94) являются задачами выпуклого программирования, однако, в зависимости от структуры ограничений, возможны различные варианты их решения.

Рассмотрим каждый возможный вариант постановки данных задач в отдельности.

Пусть $p = 1,\quad n_1 \geq 1,\quad m \geq 1$ и $m_1 = 0$. Тогда для всех трех задач получим задачу выпуклого программирования

$$\varphi(x) \to \min,$$

$$g_i(x) \leq a_i \quad (i = \overline{1,m}),$$

для решения которой можно использовать известные методы и алгоритмы выпуклой оптимизации.

Пусть теперь $p \geq 2,\quad n_j \geq 2,\quad m = 0$ и $m_j \geq 1$. Тогда получим задачи минимизации сепарабельных функционалов $F_1(x), F_2(x)$ и $F_3(x)$ на множестве блочно-диагональных ограничений без связующих ограничений, а именно: найти минимум функционала $F_1(x)$ ($F_2(x)$ или $F_3(x)$) при ограничениях (3.94).

Поскольку данные функционалы являются сепарабельными и ограничения имеют только блочно-диагональную структуру, то они сводятся к решению p задач выпуклой оптимизации следующего вида:

$$\varphi_j(x_j) \to \min, \tag{3.97}$$

$$h_j^l(x_j) \leq b_j^l \quad (l = \overline{1, m_j}), \tag{3.98}$$

отдельно для каждого $j = \overline{1, p}$.

Полученные решения x_j^* формируют общее решение $x^* = (x_1^*, x_2^*, \ldots$ $\ldots, x_p^*)$, которое дает оптимальные значения для всех трех функционалов $F_1(x^*)$, $F_2(x^*)$ и $F_3(x^*)$. Таким образом, для решения задач $\min F_r(x)$ $(r = \overline{1, 3})$ достаточно решить p задач типа (3.97), (3.98).

Рассмотрим общий случай задачи (3.92)–(3.94), т.е. когда $p \geq$ ≥ 2, $\quad n_j \geq 2$, $\quad m \geq 1$ и $m_j \geq 1$. Тогда задача (3.92)–(3.94) будет общей задачей выпуклого программирования блочно-диагональной структуры со связующими ограничениями. Рассмотрим возможные схемы декомпозиции для решения таких задач для каждого функционала в отдельности.

Метод решения задачи (3.92)–(3.94) с использованием схемы декомпозиции по ограничениям и субградиентных методов приведен в п. 3.4.

Для задачи минимизации функционала (3.95) или (3.96) при ограничениях (3.93), (3.94) получим следующие вариации предложенного метода.

Для данных задач функции Лагранжа имеют вид

$$L_2(x, u) = \max_j \varphi_j(x_j) + \sum_{j=1}^{p} \sum_{i=1}^{m} u_i g_j^i(x_j) - \sum_{i=1}^{m} u_i a_i,$$

и

$$L_3(x, u) = \prod_{j=1}^{p} \varphi_j(x_j) + \sum_{j=1}^{p} \sum_{i=1}^{m} u_i g_j^i(x_j) - \sum_{i=1}^{m} u_i a_i.$$

Исходя из приведенной в п. 3.4. общей схемы декомпозиции по ограничениям с применением субградиентных методов, для этих функций Лагранжа, при фиксированных значениях $u = u^t$ необходимо решить одну из задач

$$\max_{u \geq 0} \{ L_2^*(u) = \min_{x \in S_1 \times \cdots \times S_p} L_2(x, u) \} \tag{3.99}$$

или

$$\max_{u\geq 0}\{L_3^*(u) = \min_{x\in S_1\times\dots\times S_p} L_3(x,u),\} \tag{3.100}$$

где S_j – множество значений переменных x_j, удовлетворяющих ограничениям (3.94) для каждого j в отдельности. Так как функционалы в данном случае не являются сепарабельными, то для нахождения обобщенного градиента функции

$$L_2^*(u) = \max_{u\geq 0} L_2(x,u)$$

или

$$L_3^*(u) = \max_{u\geq 0} L_3(x,u)$$

применяется приближенный алгоритм. А именно, на каждом шаге субградиентного метода решение задачи (3.99) или (3.100) для функционалов $L_2(x,u^t)$ или $L_3(x,u^t)$ сводится к решению задач типа

$$\varphi_j(x_j) + \sum_{i=1}^{m} u_i^t\, g_j^i(x_j) \to \min, \tag{3.101}$$

$$h_j^l(x_j) \leq b_j^l \quad (l = \overline{1,m_j}), \tag{3.102}$$

для каждого блока в отдельности, которые являются задачами выпуклого программирования.

Обобщенный градиент функций $L_2^*(u)$ и $L_3^*(u)$ в точке $u = u^t$ в задачах (3.99) и (3.100) определяются по формуле

$$g_i(u^k) = \sum_{j=1}^{p} g_j^i(x_j^*(u^t)) - a_i \quad (i = \overline{1,m}),$$

где $x_j^*(u^t)$ $(j = \overline{1,p})$ – решения задач (3.101), (3.102).

3.9. Блочная задача выпуклого программирования со связующими переменными и ограничениями

Пусть имеем p групп переменных $x_j = (x_j^1, x_j^2, \dots, x_j^{n_j}) \in R^{n_j}$, образующих вектор $x = (x_1, x_2, \dots, x_p) \in R^n$, где $n = \sum\limits_{j=1}^{p} n_j$. На множестве каждой группы переменных x_j определим следующие непрерывные функции: φ_j, g_j^i $(i = \overline{1,m})$, h_j^l $(l = \overline{1,m_j})$: $R^{n_j} \to R$.

Пусть также имеются переменные $y = (y_1, y_2, \ldots, y_k) \in R^k$, на которых определены функции α, $F_0^i \quad (i = \overline{1,m})$, $F_j^l \quad (l = \overline{1,m_j};\ j = \overline{1,p}) : \ R^k \to R$. Предположим, что функции φ_j, $g_j^i \quad (i = \overline{1,m})$, h_j^l $(l = \overline{1,m_j})$ выпуклы на множестве значений переменных x_j, а функции α, $F_0^i \quad (i = \overline{1,m})$, $\quad F_j^l \quad (l = \overline{1,m_j};\ j = \overline{1,p})$ выпуклы на множестве значений переменных y.

Рассмотрим задачу выпуклого программирования на множестве переменных (x, y), в которой система ограничений имеет блочно-диагональную структуру со связующими переменными и ограничениями:

$$F(x,y) = \sum_{j=1}^{p} \varphi_j(x_j) + \alpha(y) \to \min, \tag{3.103}$$

$$\sum_{j=1}^{p} g_j^i(x_j) + F_0^i(y) \le a_i \quad (i = \overline{1,m}), \tag{3.104}$$

$$h_j^l(x_j) + F_j^i(y) \le b_j^l \quad (l = \overline{1,m_j}; \;\; j = \overline{1,p}), \tag{3.105}$$

где $a_i \;\; (i = \overline{1,m})$ и $b_j^l \;\; (l = \overline{1,m_j};\ j = \overline{1,p})$ – заданные числа.

Через $M(x, y)$ обозначим множество значений переменных (x, y), удовлетворяющих ограничениям (3.103)–(3.105), и предположим, что оно ограничено, и представляет собой некоторый компакт.

Задача (3.103)–(3.105) является задачей выпуклого программирования, для решения которой можно построить итеративный алгоритм с использованием субградиентных методов, применяя схему декомпозиции по переменным y, а при решении задачи на множестве переменных x – схему декомпозиции по ограничениям. Также для решения задачи на множестве переменных x, которые при фиксированных $y = \overline{y}$ имеют блочно-диагональные структуры со связующими ограничениями, можно применить схему декомпозиции по ресурсам (схема Корнаи-Липтака). В каждом из таких случаях алгоритмы решения задачи (3.103)–(3.105) будут иметь два уровня, на каждом из которых используются субградиентные методы.

Рассмотрим алгоритм более подробно.

Первый (внешний) *уровень*. Применяем схему декомпозиции по переменным. Тогда для фиксированных переменных $y = \overline{y}$ получим задачу:

$$F(x,\overline{y}) = \sum_{j=1}^{p} \varphi_j(x_j) + \alpha(\overline{y}) \to \min, \tag{3.106}$$

$$\sum_{j=1}^{p} g_j^i(x_j) \le \overline{a}_i(\overline{y}) \quad (i = \overline{1,m}), \tag{3.107}$$

$$h_j^l(x_j) \le b_j^l(\overline{y}) \quad (l = \overline{1,m_j};\ j = \overline{1,p}), \tag{3.108}$$

где $\overline{a}_i(\overline{y}) = a_i - F_0^i(\overline{y})$, $\overline{b}_j^l(\overline{y}) = b_j^l - F_j^l(\overline{y})$ $(l = \overline{1,m_j};\ j = \overline{1,p})$ – постоянные коэффициенты.

На множестве значений переменных x определим функцию

$$\Phi(\overline{y}) = \min_{x \in R(\overline{y})} F(x, \overline{y}).$$

Здесь $R(\overline{y})$ – множество значений переменных x, удовлетворяющих ограничениям (3.107), (3.108) при фиксированных значениях переменных $y = \overline{y}$.

Тогда на внешнем уровне итеративного алгоритма решения задачи (3.103)–(3.105) необходимо минимизировать выпуклую функцию $\Phi(\overline{y})$ при ограничениях $\overline{y} \ge 0$. Для этого используем метод обобщенного градиентного спуска, на t-м шаге которого необходимо выполнить три этапа.

1. Решить задачу (3.106)–(3.108) при фиксированных $y = \overline{y}^t$, т.е. найти оптимальное решение $x(\overline{y}^t)$ и двойственные оценки $\{u_i(\overline{y}^t)\}$ и $\{v_j^l(\overline{y}^t)\}$ соответственно для ограничений (3.107) и (3.108).
2. Определить значения обобщенного градиента функции $\Phi(y)$ в точке $y = \overline{y}^t$ по формуле

$$G(\overline{y}^t) = G_\alpha - \sum_{i=1}^{m} u_i(\overline{y}^t) G_0^i(\overline{y}^t) - \sum_{j=1}^{p} \sum_{i=1}^{m} v_j^l(\overline{y}^t) G_j^l(\overline{y}^t),$$

   где G_α – обобщенный градиент функции $\alpha(y)$, а $G_0^i(\overline{y}^t)$ и $G_j^l(\overline{y}^t)$ – обобщенные градиенты функций $F_0^i(y)$ и $F_j^l(y)$ в точке $y = \overline{y}^t$.
3. Вычислить новые значения

$$\overline{y}^{t+1} = \overline{y}^t - \gamma_{t+1} G(\overline{y}^t),$$

   где γ_{t+1} – величина шага.

В первом пункте внешнего уровня итеративного алгоритма необходимо решить задачу (3.106)–(3.108). Так как эта задача имеет блочно-диагональную структуру со связующими ограничениями, то для ее решения можно использовать два приема декомпозиции:

- схема декомпозиции по ограничениям,
- схема декомпозиции по ресурсам.

Решение задачи (3.106)–(3.108) составляет *второй* (внутренний) *уровень* итеративного алгоритма.

1) *Схема декомпозиции по ограничениям.* Для этого рассмотрим следующие две задачи:

$$\max_{u \geq 0} L^*(u)$$

и

$$L^*(u) = \min_{x \in S(x)} L(x, u),$$

где $u = \{u_i\}$ – множители Лагранжа функции $L(x, u)$, определенной по формуле

$$L(x, u) = \sum_{j=1}^{p} \left[\varphi_j(x_j) + \sum_{i=1}^{m} u_i g_j^i(x_j) \right] + \alpha(\overline{y}) - \sum_{i=1}^{m} u_i \bar{a}_i(\overline{y}),$$

а $S(x)$ – множество значений переменных x, удовлетворяющих ограничениям (3.108).

Тогда, на $(s+1)$-м шаге субградиентного метода максимизации функции $L^*(u)$ необходимо провести три этапа.

1. Решить задачу выпуклого программирования

$$\sum_{j=1}^{p} \left[\varphi_j(x_j) + \sum_{i=1}^{m} u_i^s g_j^i(x_j) \right] + \alpha_0(u^s, \overline{y}) \to \min, \tag{3.109}$$

$$h_j^l(x_j) \leq b_j^l(\overline{y}) \quad (l = \overline{1, m_j};\ j = \overline{1, p}), \tag{3.110}$$

где $\alpha_0(u^s, \overline{y}) = \alpha(\overline{y}) - \sum\limits_{i=1}^{m} u_i^s \overline{a}_i(\overline{y})$ при фиксированных $u = u^s$.

Тогда для каждого $j = \overline{1, p}$ необходимо решить следующие задачи выпуклого программирования

$$\varphi_j(x_j) - \lambda^s \psi_j(x_j) + \sum_{i=1}^{m} u_i^s g_j^i(x_j) \to \min,$$

$$h_j^l(x_j) \leq b_j^l(\overline{y}) \quad (l = \overline{1, m_j}).$$

Пусть $x_j^*(u^s)$ – решение каждой задачи в отдельности, а $v_j^{l*}(u^s)$ – соответствующие значения множителей Лагранжа для этих решений. Тогда $x^*(u^s) = \{x_j^*(u^s)\}$ является решением задачи (3.109), (3.110).

2. Определить значения обобщенного градиента функции $L^*(u)$ в точке $u = u^s$ по формуле

$$G(u^s) = \Big\{ \sum_{j=1}^{p} g_j^i(x_j^*(u^s)) - \overline{\alpha}_i(\overline{y}) \Big\}.$$

3. Вычислить новые значения

$$u^{s+1} = \max\{0, u^s + \gamma_{s+1} G(u^s)\},$$

где γ_{s+1} – величина шага.

Таким образом, в результате решения задачи (3.103)–(3.105) по указанному алгоритму получим оптимальные значения переменных (x, y) и двойственных оценок u и $\{v_j\}$.

Заметим, что если в задаче (3.103)–(3.105) имеются некоторые отклонения от общей постановки, то для ее решения можно использовать другие схемы итеративного алгоритма. Рассмотрим некоторые вариации структуры системы ограничений.

Пусть $m = 0$, т.е. связующие ограничения (3.104) отсутствуют. Тогда, если использовать схему декомпозиции по переменным, то задачу (3.106)–(3.108) можно решить для каждого блока в отдельности. В конечном итоге получим одноуровневый алгоритм. Аналогично имеет место, когда $m > 0$, но все функции $g_j^i(x_j)$ тождественно равны нулю на множестве переменных x (т.е. отсутствуют связующие ограничения по x).

Если же $m_j = 0 \quad (j = \overline{1, p})$, т.е. блочно-диагональные ограничения (3.105) отсутствуют, то получим задачу, для которой достаточно применить схему декомпозиции по переменным. В этом случае эффективность данного метода достигается за счет специальной структуры ограничений (3.107) при фиксированных значениях y (транспортного типа, сетевая структура и т.п.).

2) *Схема декомпозиции по ресурсам*

На втором уровне алгоритма применяется схема декомпозиции по ресурсам для решения задач (3.106)–(3.108) при фиксированных $y = \overline{y}$. Так как на первом уровне фиксируются переменные $y = \overline{y}$, то система ограничений (3.107), (3.108) имеет блочно-диагональную структуру на множестве переменных x со связующими ограничениями.

Представим ресурсы $\overline{a}_i(y)$ в виде следующих равенств

$$\overline{a}_i(\overline{y}) = z'_{i1}(\overline{y}) + z'_{i2}(\overline{y}) + \cdots + z'_{ip}(\overline{y}) \quad (i = \overline{1, m}). \tag{3.111}$$

В векторной форме равенство (3.111) имеет вид

$$b'(\overline{y}) = z'_1(\overline{y}) + z'_2(\overline{y}) + \cdots + z'_p(\overline{y}),$$

где $z'_j(\overline{y})$ – вектор-столбец выделенных ресурсов j-му блоку.

Тогда задача (3.106)–(3.108) разлагается на p подзадач выпуклого программирования типа

$$\Phi_j(x_j(z'_j(\overline{y}))) = \varphi_j(x_j) \to \min, \tag{3.112}$$

$$g^i_j(x_j) \leq z'_{ij}(\overline{y}) \quad (i = \overline{1,m}), \tag{3.113}$$

$$h^l_j(x_j) \leq b^l_j(\overline{y}) \quad (l = \overline{1,m_j}). \tag{3.114}$$

Пусть $x_j(z'_j(\overline{y}))$ – оптимальное решение задачи (3.112)–(3.114) при заданном $z'(\overline{y})$, а $u_{ij}(z'_{ij}(\overline{y}))$, $v_j(z'_{ij}(\overline{y}))$ – значения множителей Лагранжа, соответствующие ограничениям (3.113) и (3.114), где функция Лагранжа определяется по формуле:

$$L_j(x_j, u_j, v_j) = \varphi_j(x_j) + \sum_{i=1}^{m} u_{ij}\left[g^i_j(x_j) - z'_{ij}(\overline{y})\right] + \sum_{l=1}^{m_j} v_l(h^l_j(x_j) - b^l_j(\overline{y})).$$

Определим следующую функцию:

$$\Phi(x(z'(\overline{y}))) = \sum_{j=1}^{p} \Phi(x_j(z'_j(\overline{y}))),$$

которая является выпуклой. Субградиент этой функции определяется по формуле:

$$G_\Phi(z'(\overline{y})) = \left\{g_j(z'_j(\overline{y})) = -u_j(z'_j(\overline{y}))\right\}. \tag{3.115}$$

Таким образом задача (3.106)–(3.108) сводится к решению задачи минимизации функционала $\Phi(x(z'(\overline{y})))$ при линейных ограничениях (3.111), т.е. координирующая задача внутреннего уровня имеет вид

$$\Phi(x(z'(\overline{y}))) \to \max,$$

$$\sum_{j=1}^{p} z'_{ij}(\overline{y}) = \overline{a}_i(y) \quad (i = \overline{1,m}).$$

Для решения этой задачи можем использовать субградиентный метод с проектированием на линейные многообразия. На k-м шаге выбранного метода необходимо выполнить три этапа.

1. При фиксированных $z'_j(\overline{y}) = z'^k_j(\overline{y})$ решить задачу (3.112)–(3.114) и найти оптимальные решения $x_j(z'_j(\overline{y}))$ и соответствующие двойственные оценки $u_j(z'_j(\overline{y}))$ ограничений (3.113).

2. Определить значения обобщенного градиента функции $\Phi(x(z'(\overline{y})))$ в точке $z'_j(\overline{y}) = z'^k_j(\overline{y})$ по формуле (3.115).

3. Вычислить новые значения $z'^{k+1}_j(\overline{y})$ по формулам выбранного субградиентного метода с учетом выполнения линейных уравнений (3.111).

Глава 4

Задачи дробно-линейного программирования

В данной главе рассматривается задача дробно-линейного программирования, для которой приводится основные теоретические результаты и различные алгоритмы ее решения: прямой и модифицированный симплекс-метод, метод сведения к задаче линейного программирования, параметрический метод и методы декомпозиции. Также для решения задач дробно-линейного программирования приводятся полиномиальные алгоритмы и субградиентный метод.

Для решения блочно-диагональных задач дробно-линейного программирования предлагаются декомпозиционный метод Данцига-Вулфа, схемы декомпозиции по ограничениям, переменным и ресурсам с применением субградиентных методов, в результате которых исходная задача сводится к решению линейных задач для каждого блока в отдельности.

4.1. Задачи дробно-линейного программирования

Рассмотрим следующую задачу дробно-линейного программирования:

$$F(x) = \frac{C(x)}{D(x)} = \frac{\sum\limits_{j=1}^{n} c_j x_j}{\sum\limits_{j=1}^{n} d_j x_j} \to \max, \tag{4.1}$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \quad (i = \overline{1,m}), \tag{4.2}$$

$$x_j \ge 0 \quad (j = \overline{1,n}). \tag{4.3}$$

Обозначим через R(x) многогранное выпуклое множество, заданное ограничениями (4.2)–(4.3), т.е.

$$R(x) = \left\{ x \Big/ \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad (i = \overline{1,m});\ x_j \ge 0 \quad (j = \overline{1,n}) \right\},$$

и предположим, что $R(x)$ ограничено.

Задача дробно-линейного программирования (4.1)–(4.3) является наиболее изученной и для ее решения используются различные численные методы [32, 36, 148, 359, 433, 438], [488]–[495], [733], которые, как правило, основаны на предположении, что $D(x) > 0$ для любых $x \in R(x)$, и следующих утверждениях.

Теорема 4.1. *На любом прямолинейном отрезке, принадлежащем многограннику $R(x)$, дробно-линейная функция $F(x)$ изменяется монотонно.*

Теорема 4.2. *Дробно-линейная функция $F(x)$ достигает минимума (максимума) только в крайней точке многогранника $R(x)$. Если минимум (максимум) достигается в нескольких крайних точках, то он достигается и на их выпуклой оболочке.*

Для задачи дробно-линейного программирования (4.1)–(4.3) имеет место общая теория двойственности линейного программирования [2, 46, 110, 152, 457, 460, 511]. Однако для задач дробно-линейного программирования двойственные задачи строятся неоднозначно, но все они являются эквивалентными.

Двойственная задача наиболее общего вида строится следующим образом:

1) каждому ограничению системы ограничений (4.2) ставится в соответствие две двойственные переменные u_i и v_i – первая для числителя целевой функции, а вторая для знаменателя;

2) целевая функция двойственной задачи имеет вид

$$W(u,v)=\frac{W_1(u)}{W_2(v)}=\frac{\sum\limits_{i=1}^{m} b_i u_i}{\sum\limits_{i=1}^{m} b_i v_i}\to \min; \tag{4.4}$$

3) ограничения двойственной задачи строятся для каждой переменной $x_j \quad (j=\overline{1,n})$ исходной задачи и имеют нелинейный вид

$$\Bigl(c_j-\sum_{i=1}^{m} a_{ij}u_i\Bigr)\sum_{i=1}^{m} b_i v_i-\Bigl(d_j-\sum_{i=1}^{m} a_{ij}v_i\Bigr)\sum_{i=1}^{m} b_i u_i\le 0 \quad (j=\overline{1,n}); \tag{4.5}$$

4) ограничения на неотрицательности переменных

$$\sum_{i=1}^{m} b_i v_i>0,\ u_i\ge 0,\ v_i\ge 0 \quad (i=\overline{1,m}). \tag{4.6}$$

Для задач (4.1)–(4.3) и (4.4)–(4.6) имеют место следующие основные теоремы теории двойственности.

Теорема 4.3. *Если x – допустимое решение задачи дробно-линейного программирования* (4.1)–(4.3), *а* (u,v) *допустимое решение двойственной задачи* (4.4)–(4.6), *то имеет место неравенство*

$$F(x)\le W(u,v).$$

Теорема 4.4. *Если x^* – оптимальное решение задачи* (4.1)–(4.3), *а* (u^*,v^*) *– оптимальное решение двойственной задачи* (4.4)–(4.6), *то*

$$F(x^*)=W(u^*,v^*)$$

и обратно, если для допустимых планов x^* исходной задачи дробно-линейного программирования и допустимого решения (u^*,v^*) двойственной задачи значение целевых функций этих задач совпадают $F(x^*)=W(u^*,v^*)$, то x^* является оптимальным решением для исходной задачи, а (u^*,v^*) – оптимальное решение двойственной задачи.

Таким образом условие $F(x^*) = W(u^*, v^*)$ является необходимым и достаточным условием оптимальности решений x^* и (u^*, v^*).

Теорема 4.5. *Для того, чтобы допустимые решения x^* для задачи дробно-линейного программирования* (4.1)–(4.3) *и* (u^*, v^*) *для двойственной задачи* (4.4)–(4.6) *были оптимальными необходимо и достаточно выполнение следующих двух групп равенств*

$$\text{(I)} \qquad x_j^* \cdot \Big[\Big(c_j - \sum_{i=1}^m a_{ij}u_i^*\Big)\sum_{i=1}^m b_i v_i^* - \Big(d_j - \sum_{i=1}^m a_{ij}v_i^*\Big)\sum_{i=1}^m b_i u_i^*\Big] = 0,$$

$$\text{(II)} \quad u_i^* \cdot \Big(\sum_{j=1}^n a_{ij}x_j^* - b_i\Big) = 0 \quad \textit{и} \quad v_i^* \cdot \Big(\sum_{j=1}^n a_{ij}x_j^* - b_i\Big) = 0.$$

Эти равенства являются аналогом условий дополняющих нежесткостей второй основной теоремы двойственности линейного программирования.

Для задачи дробно-линейного программирования (4.1)–(4.3) функцию Лагранжа можно построить двумя способами

$$L_1(x,u) = F(x) + \sum_{i=1}^m u_i\Big(\sum_{j=1}^n a_{ij}x_j - b_i\Big), \tag{4.7}$$

$$L_2(x,u) = \Big\{C(x) + \sum_{i=1}^m u_i\Big(\sum_{j=1}^n a_{ij}x_j - b_i\Big)\Big\}\Big/D(x). \tag{4.8}$$

Тогда решение задачи дробно-линейного программирования (4.1)–(4.3) может быть сведено к нахождению седловой точки функции Лагранжа (4.7) или (4.8), т.е.

$$L_r(x^*, u^*) = \max_{u\ge 0}\min_{x\ge 0} L_r(x,u) = \min_{x\ge 0}\max_{u\ge 0} L_r(x,u),\ r = 1, 2.$$

Теорема 4.6. *Если пара (x^*, u^*) является седловой точкой функции $L_1(x,u)$, то пара $(x^*, \overline{u}^*)$, где $\overline{u}_1^* = u_1^* D(x^*)$, есть седловая точка функции $L_2(x,u)$ и наоборот.*

Разработанные методы решения обычной задачи дробно-линейного программирования (4.1)–(4.3) могут быть разделены на четыре группы:

- модификации симплекс-метода [32, 222, 223, 226, 305], [356]–[358], [460, 493, 494, 508, 513, 531, 592, 635, 655];

- методы "линеаризации", т.е. сведение к решению задачи линейного программирования с изменением системы ограничений исходной задаче дробно-линейного программирования [32, 133, 493, 494, 593, 601, 628, 673, 733];

- параметрические методы, т.е. сведение к решению последовательности задач параметрического линейного программирования без изменения системы ограничений исходной задаче дробно-линейного программирования [4, 32, 85, 171, 280, 493, 494, 509, 525, 545, 546, 592, 733];

- полиномиальные алгоритмы [24, 593, 685, 733];

- декомпозиционные методы [16, 17, 112, 318, 320, 336, 624, 636], [663]–[667], [669, 670, 672, 674, 675, 733].

Как отмечается в [3, 32, 80, 99, 381, 493, 494, 733], методы первых трех групп имеют близкую вычислительную сложность, а приоритет отдается методу, позволяющему эффективнее решать конкретную дробно-линейную задачу.

Задачу (4.1)–(4.3) можно рассматривать как частный случай следующих обобщенных задач дробно-линейного программирования:

- сумма двух дробно-линейных функций

$$\min_{x\in R(x)} \left\{ F(x) = \frac{C(x)}{D(x)} + \frac{P(x)}{Q(x)} \right\}; \tag{4.9}$$

- сумма линейной и дробно-линейной функцией

$$\min_{x\in R(x)} \left\{ F(x) = \frac{C(x)}{D(x)} + P(x) \right\}, \tag{4.10}$$

где

$$P(x) = \sum_{j=1}^{n} p_j x_j + p_0, \ Q(x) = \sum_{j=1}^{n} q_j x_j + q_0.$$

Для задачи (4.9) в работе [631] предлагаются методы ее сведения к решению задачи (4.10). Задача (4.10) рассмотрена в работах [109, 249, 355, 431, 631, 671, 700], и для случаев, когда $D(x) > 0$ и $P(x) > 0$ предлагаются модификации симплекс-метода.

4.2. Основные методы решения задач дробно-линейного программирования

Как было отмечено ранее, разработанные методы решения задач дробно-линейного программирования могут быть разделены на четыре группы: модификации симплекс-метода, методы "линеаризации", параметрические методы и полиномиальные алгоритмы. Среди них особенно выделяется группа алгоритмов параметрического метода, которые являются наиболее универсальными и могут быть использованы для решения задач дробно-линейного программирования с различной структурой системы ограничений. При сведении задач дробно-линейного программирования к задачам линейного программирования алгоритмы параметрического метода оставляют без изменений структуру системы ограничений.

В данном параграфе, кроме известных ранее методов решения задач дробно-линейного программирования, приводится модификация параметрического метода и полиномиальные алгоритмы типа Кармаркара-Дикина. Предлагаемая модификация параметрического метода позволяет применять программное обеспечение линейного программирования при решении практических задач дробно-линейного программирования. На ее основе в дальнейшем будут построены алгоритмы решения задач дробно-линейного программирования специальной структуры: блочно-диагональные задачи, транспортные задачи, а также другие задачи специальной структуры, для которых в линейном случае имеются эффективные алгоритмы их решения. В конце параграфа приводятся два полиномиальных алгоритма решения общих задач дробно-линейного программирования, основанных на методе внутренних точек Дикина и проективного алгоритма Кармаркара-Барнеса решения задач линейного программирования.

Рассмотрим более детально каждую группу.

4.2.1. Модификации симплекс-метода

Пусть задана задача дробно-линейного программирования

$$F(x) = \frac{C(x)}{D(x)} = \frac{\sum\limits_{j=1}^{n} c_j x_j + c_0}{\sum\limits_{j=1}^{n} d_j x_j + d_0} \to \max, \tag{4.11}$$

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \quad (i = \overline{1,m}), \tag{4.12}$$

$$x_j \geq 0 \quad (j = \overline{1,n}), \tag{4.13}$$

для которой выполняется предположение, что $D(x) > 0$ для любого $x \in R(x)$ и многогранное выпуклое множество $R(x)$ ограничено. Тогда применение симплекс-метода основывается на следующих свойствах задачи (4.11)–(4.13):

- минимальное значение дробно-линейного функционала $F(x)$ достигается в крайней точке многогранника $R(x)$;
- вдоль любого ребра многогранника $R(x)$ функция $F(x)$ является монотонной;
- если производная дробно-линейного функционала $F(x)$ по направлению вдоль ребра многогранника $R(x)$, идущего к новой крайней точке, является отрицательной, то значение целевой функции в этой точке будет меньше по сравнению со значением целевой функции в текущей точке;
- если в некоторой крайней точке значения всех производных по направлению вдоль исходящих из нее ребер неотрицательны, то текущая точка является оптимальной.

Из изложенного следует, что для реализации симплекс-метода решения задачи дробно-линейного программирования необходимо найти эффективный способ вычисления производных по направлению вдоль всех ребер.

Частные производные дробно-линейной функции

$$F(x) = \frac{C(x)}{D(x)} = \frac{\sum\limits_{j=1}^{n} c_j x_j}{\sum\limits_{j=1}^{n} d_j x_j}$$

по каждой переменной x_j имеют вид

$$\frac{\partial F}{\partial x_j} = \frac{c_j D(x) - d_j C(x)}{D^2(x)} = \frac{1}{D^2(x)} \cdot \begin{vmatrix} c_j & C(x) \\ d_j & D(x) \end{vmatrix} = \Delta_j \cdot \frac{1}{D^2(x)},$$

где $\Delta_j = \begin{vmatrix} c_j & C(x) \\ d_j & D(x) \end{vmatrix}$. Так как $\dfrac{1}{D^2(x)} > 0$, то знак производной зависит от знака определителя Δ_j.

Таким образом, если $\Delta_j > 0$, то $\dfrac{\partial F}{\partial x_j} > 0$ и значение функции $F(x)$ растет с увеличением значении переменной x_j. Для симплекс-метода, при решении задачи дробно-линейного программирования на максимум, это означает ввод переменной x_j в базис, т.е. переход к новому базису (к новой крайней точке), в котором $x_j > 0$ и значение функции будет больше.

Если же $\Delta_j < 0$, то $\dfrac{\partial F}{\partial x_j} < 0$ и значение функции $F(x)$ уменьшается с увеличением значении переменной x_j. Для симплекс-метода, при решении задачи дробно-линейного программирования на минимум, это означает ввод переменной x_j в базис, т.е. переход к новому базису (к новой крайней точке), в котором $x_j > 0$ и значение функции будет меньше.

В результате анализа знака частных производных, получим следующие критерии оптимальности:

1) если все $\dfrac{\partial F}{\partial x_j} \leq 0 \quad (j = \overline{1,n})$ для опорного плана, то в соответствующей точке x достигается максимальное значение дробно-линейной функции;

2) если все $\dfrac{\partial F}{\partial x_j} \geq 0 \;\; (j = \overline{1,n})$, то достигается минимальное значение дробно-линейной функции.

Возможны два варианта реализации симплекс-метода решения задачи дробно-линейного программирования: прямого и модифицированного.

1. Алгоритм прямого симплекс-метода. Рассмотрим для начала особенности применения прямого симплексного метода, состоящего из обыкновенных или модифицированных жордановых исключений.

Пусть имеется задача дробно-линейного программирования в стандартном виде

$$F(x) = \frac{C(x)}{D(x)} = \frac{\sum\limits_{j=1}^{n} c_j x_j + c_0}{\sum\limits_{j=1}^{n} d_j x_j + d_0} \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad (i = \overline{1,m}),$$

$$x_j \geq 0 \quad (j = \overline{1,n}).$$

Для данной задачи составим первоначальную симплекс-таблицу в сокращенном виде. В отличие от линейной задачи в исходной симплекс-таблицу решения задачи дробно-линейного программирования для целевой функции включается не одна строка, а две – соответственно для числителя и знаменателя (см. табл. 4.1). Жордановыми исключениями

Таблица 4.1

<table>
<tr><td colspan="2" rowspan="3">Коэффициенты базисных переменных</td><td colspan="2" rowspan="2">Базис</td><td>c_1</td><td>c_2</td><td>$\dots$</td><td>c_q</td><td>$\dots$</td><td>c_n</td><td rowspan="4">$b\downarrow$</td></tr>
<tr><td>d_1</td><td>d_2</td><td>$\dots$</td><td>d_q</td><td>$\dots$</td><td>d_n</td></tr>
<tr><td rowspan="2">$x_\delta\downarrow$</td><td rowspan="2">$P_\delta\downarrow$</td><td>x_1</td><td>x_2</td><td>$\dots$</td><td>x_q</td><td>$\dots$</td><td>x_n</td></tr>
<tr><td>$c_\delta\downarrow$</td><td>$d_\delta\downarrow$</td><td>P_1</td><td>P_2</td><td>$\dots$</td><td>P_q</td><td>$\dots$</td><td>P_n</td></tr>
<tr><td></td><td></td><td></td><td></td><td>a_{11}</td><td>a_{12}</td><td>$\dots$</td><td>a_{1q}</td><td>$\dots$</td><td>a_{1n}</td><td>b_1</td></tr>
<tr><td></td><td></td><td></td><td></td><td>a_{21}</td><td>a_{22}</td><td>$\dots$</td><td>a_{2q}</td><td>$\dots$</td><td>a_{2n}</td><td>b_2</td></tr>
<tr><td></td><td></td><td></td><td></td><td>$\dots$</td><td>$\dots$</td><td>$\dots$</td><td>$\dots$</td><td>$\dots$</td><td>$\dots$</td><td>$\dots$</td></tr>
<tr><td></td><td></td><td></td><td></td><td>a_{p1}</td><td>a_{p2}</td><td>$\dots$</td><td>$\boxed{a_{pq}}$</td><td>$\dots$</td><td>a_{pn}</td><td>b_p</td></tr>
<tr><td></td><td></td><td></td><td></td><td>$\dots$</td><td>$\dots$</td><td>$\dots$</td><td>$\dots$</td><td>$\dots$</td><td>$\dots$</td><td>$\dots$</td></tr>
<tr><td></td><td></td><td></td><td></td><td>a_{m1}</td><td>a_{m2}</td><td>$\dots$</td><td>a_{mq}</td><td>$\dots$</td><td>a_{mn}</td><td>b_m</td></tr>
<tr><td colspan="4">Строка числителя</td><td>$-c_1$</td><td>$-c_2$</td><td>$\dots$</td><td>$-c_q$</td><td>$\dots$</td><td>$-c_n$</td><td>c_0</td></tr>
<tr><td colspan="4">Строка знаменателя</td><td>$-d_1$</td><td>$-d_2$</td><td>$\dots$</td><td>$-d_q$</td><td>$\dots$</td><td>$-d_n$</td><td>d_0</td></tr>
<tr><td colspan="4">Δ-строка</td><td>Δ_1</td><td>Δ_2</td><td>$\dots$</td><td>Δ_q</td><td>$\dots$</td><td>Δ_n</td><td>F_0</td></tr>
</table>

отыскивается оптимальный план, при этом коэффициенты числителя и знаменателя подвергаются всем преобразованиям. Для нахождения первоначального опорного плана (т.е. для приведения задачи дробно-линейного программирования к каноническому виду) можно использовать задачу линейного программирования в которой необходимо максимизировать числитель задачи дробно-линейного программирования при тех же ограничениях. При этом жордановым исключениям преобразуются и коэффициенты d_j.

В результате этих преобразований на k-й итерации симплекс-метода имеем некоторый опорный план x^k, для которого коэффициенты системы ограничений принимают новые значения a_{ij}^k и b_i^k, числителя и знаменателя – соответственно новые значения c_j^k и d_j^k (см. табл. 4.2).

В таблице 4.2 вектора с номерами $r_1, r_2, \dots, r_m$ формируют базис, а вектор $x^k = (x_{r_1}, x_{r_2}, \dots, x_{r_m})$ является базисным решением и формирует опорный план дробно-линейной задачи. Тогда если значения числителя и знаменателя дробно-линейной функции для этого плана обозначим через $C(x^k)$ и $D(x^k)$, то критерий оптимальности текущего опорного плана будет

$$\Delta_j^k = \begin{vmatrix} c_j^k & C(x^k) \\ d_j^k & D(x^k) \end{vmatrix} = D(x^k)c_j^k - C(x^k)d_j^k \geq 0 \ \ (j = \overline{1,n}). \qquad (4.14)$$

Таблица 4.2

Коэффициенты базисных переменных		Базис		c_1	c_2	$\dots$	c_q	$\dots$	c_n	b
				d_1	d_2	$\dots$	d_q	$\dots$	d_n	
		$x_\delta \downarrow$	$P_\delta \downarrow$	x_1	x_2	$\dots$	x_q	$\dots$	x_n	
$c_\delta \downarrow$	$d_\delta \downarrow$			P_1	P_2	$\dots$	P_q	$\dots$	P_n	
c_{r_1}	d_{r_1}	x_{r_1}	P_{r_1}	a_{11}^k	a_{12}^k	$\dots$	a_{1q}^k	$\dots$	a_{1n}^k	b_1^k
c_{r_2}	d_{r_2}	x_{r_2}	P_{r_2}	a_{21}^k	a_{22}^k	$\dots$	a_{2q}^k	$\dots$	a_{2n}^k	b_2^k
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
c_{r_p}	d_{r_p}	x_{r_p}	P_{r_p}	a_{p1}^k	a_{p2}^k	$\dots$	1	$\dots$	a_{pn}^k	b_p^k
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
c_{r_m}	d_{r_m}	x_{r_m}	P_{r_m}	a_{m1}^k	a_{m2}^k	$\dots$	a_{mq}^k	$\dots$	a_{mn}^k	b_m^k
Числитель				c_1^k	c_2^k	$\dots$	c_q^k	$\dots$	c_n^k	$C(x^k)$
Знаменатель				d_1^k	d_2^k	$\dots$	d_q^k	$\dots$	d_n^k	$D(x^k)$
Δ_j^k				Δ_1^k	Δ_2^k	$\dots$	Δ_q^k	$\dots$	Δ_n^k	$F(x^k)$

Поэтому если $\Delta_j^k \geq 0$ для всех $j = \overline{1,n}$, то текущий опорный план является оптимальным, а если хотя бы один, например $\Delta_q^k < 0$, то соответствующий вектор P_q вводится в базис по обычным правилам прямого симплекс-метода. Следует отметить, что если соответствующий вектор P_q для ввода в базис не имеет положительных коэффициентов, то задача дробно-линейного программирования имеет асимптотические (неограниченные) решения.

Таким образом нахождение нового опорного плана x^{k+1} при решении задачи дробно-линейного программирования осуществляется по следующим правилам:

- выделяем вектор P_q для ввода в базис, исходя из наименьшей отрицательной оценки Δ_q^k, т.е. находим

$$\Delta_q^k = \min_{\Delta_j^k < 0} \Delta_j^k;$$

- находим вектор P_p для вывода его из базиса, вычислив

$$\frac{b_p^k}{a_{pq}^k} = \min_{a_{iq}^k > 0} \left\{ \frac{b_i^k}{a_{iq}^k} \right\}$$

в результате определяем разрешающий элемент a_{pq}^k;

- значения элементов новой симплекс-таблицы находятся по формулам:

$$a_{pq}^{k+1} = \frac{1}{a_{pq}^k},$$

$$a_{pj}^{k+1} = \frac{a_{pj}^k}{a_{pq}^k} \quad (j = \overline{1,n};\ j \neq q),$$

$$a_{iq}^{k+1} = -\frac{a_{iq}^k}{a_{pq}^k} \quad (i = \overline{1,m};\ i \neq p),$$

$$b_p^{k+1} = \frac{b_p^k}{a_{pq}^k},$$

$$a_{ij}^{k+1} = \frac{a_{ij}^k a_{pq}^k - a_{pj}^k a_{iq}^k}{a_{pq}^k} \quad (i = \overline{1,m};\ j = \overline{1,n};\ i \neq p;\ j \neq q),$$

$$c_j^{k+1} = \frac{c_j^k a_{pq}^k - a_{pj}^k c_q^k}{a_{pq}^k} \quad (j = \overline{1,n};\ j \neq q),$$

$$d_j^{k+1} = \frac{d_j^k a_{pq}^k - d_q^k a_{pj}^k}{a_{pq}^k} \quad (j = \overline{1,n};\ j \neq q),$$

$$b_i^{k+1} = \frac{b_i^k a_{pq}^k - a_{iq}^k b_p^k}{a_{pq}^k} \quad (i = \overline{1,m};\ i \neq p),$$

$$C(x^{k+1}) = \frac{C(x^k) a_{pq}^k - c_q^k b_p^k}{a_{pq}^k},$$

$$D(x^{k+1}) = \frac{D(x^k) a_{pq}^k - d_q^k b_p^k}{a_{pq}^k}.$$

В результате получаем новую симплекс-таблицу (см. табл. 4.3). Полученный опорный план x^{k+1} проверяем на оптимальность, т.е. находим значения оценок Δ_j^{k+1} по формуле (4.14). Если все $\Delta_j^{k+1} \geq 0$, то текущий план оптимален. В противном случае, проводим очередную итерацию прямого симплекс-метода.

Следует отметить, что коэффициенты c_j и d_j при базисных переменных не участвуют в дальнейших расчетах. Поэтому при дальнейших расчетах они в симплекс-таблице не заносятся.

Таблица 4.3

Базис		x_1	x_2	$\dots$	x_q	$\dots$	x_n	b
x_δ^{k+1}	P_δ^{k+1}	P_1	P_2	$\dots$	P_q	$\dots$	P_n	
x_{r_1}	P_{r_1}	a_{11}^{k+1}	a_{12}^{k+1}	$\dots$	a_{1q}^{k+1}	$\dots$	a_{1n}^{k+1}	b_1^{k+1}
x_{r_2}	P_{r_2}	a_{21}^{k+1}	a_{22}^{k+1}	$\dots$	a_{2q}^{k+1}	$\dots$	a_{2n}^{k+1}	b_2^{k+1}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
x_{r_q}	P_{r_q}	a_{p1}^{k+1}	a_{p2}^{k+1}	$\dots$	a_{pq}^{k+1}	$\dots$	a_{pn}^{k+1}	b_p^{k+1}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
x_{r_m}	P_{r_m}	a_{m1}^{k+1}	a_{m2}^{k+1}	$\dots$	a_{mq}^{k+1}	$\dots$	a_{mn}^{k+1}	b_m^{k+1}
Числитель		c_1^{k+1}	c_2^{k+1}	$\dots$	c_q^{k+1}	$\dots$	c_n^{k+1}	$C(x^{k+1})$
Знаменатель		d_1^{k+1}	d_2^{k+1}	$\dots$	d_q^{k+1}	$\dots$	d_n^{k+1}	$D(x^{k+1})$
Δ_j^{k+1}		Δ_1^{k+1}	Δ_2^{k+1}	$\dots$	Δ_q^{k+1}	$\dots$	Δ_n^{k+1}	$F(x^{k+1})$

Следовательно, алгоритм решения задачи дробно-линейного программирования прямым симплексным методом отличается от алгоритма решения обычной задачи линейного программирования лишь способом вычисления оценок для проверки критерия оптимальности текущего опорного плана, которые находятся по формулам (4.14).

2. Алгоритм модифицированного симплекс-метода. Рассмотрим особенности применения модифицированного симплекс-метода. Для решения задач дробно-линейного программирования. Пусть B – текущий допустимый базис. Находим оценки $u = \{u_i\}$ и $v = \{v_i\}$ соответственно числителя и знаменателя по формулам

$$u = c_\delta B^{-1} \text{ и } \ v = d_\delta B^{-1},$$

где c_δ и d_δ – векторы коэффициентов числителя и знаменателя при базисных переменных, а B^{-1} – матрица, обратная матрице B. Используя оценки u и v, критерий оптимальности текущего базиса B можно записать в виде

$$\Delta_j = \left(c_j - \sum_{i=1}^{m} a_{ij}u_i\right)D(x) - \left(d_j - \sum_{i=1}^{m} a_{ij}v_i\right)C(x) \le 0 \quad (j = \overline{1,n}),$$

где $C(x)$ и $D(x)$ – соответствующие значения числителя и знаменателя в текущем базисе, x – опорное решение, соответствующее текущему базису. Тогда, как и в прямом симплекс-методе, если $\Delta_j \le 0$ для всех $j = \overline{1,n}$, то текущий базис является оптимальным. Если некоторый $\Delta_q > 0$, то q-й вектор вводится в базис. Далее выполняется одна итерация модифицированного симплекс-метода по обычным правилам.

Итак, модифицированный симплексный метод решения задач дробно-линейного программирования включает следующие этапы.

1. Приведение задачи дробно-линейного программирования к стандартному виду.
2. Нахождение первоначального опорного плана задачи дробно-линейного программирования (можно использовать модифицированный симплекс-метод решения расширенной задачи линейного программирования на максимум числителя задачи дробно-линейного программирования).
3. Вычисление матрицы B^{-1}, обратной матрице B, составленной из векторов-столбцов базиса.
4. Вычисление векторов $u = c_\delta \cdot B^{-1}$; $v = d_\delta \cdot B^{-1}$.
5. Вычисление оценок переменных
   $\Delta_j = (c_j - uP_j)D(x) - (d_j - vP_j)C(x) \quad (j = \overline{1, n})$,
   где $C(x)$ и $D(x)$ – соответственно значения числителя и знаменателя в текущем базисе.
6. Проверка полученного опорного плана на оптимальность. Если все $\Delta_j \leq 0$, то опорный план является оптимальным; если же среди Δ_j имеются положительные, то план неоптимальный и переходим к пункту 7.
7. Выбор среди положительных оценок Δ_j наибольшей. Пусть это будет Δ_q. Тогда вычисляют координаты вектора P_q в данном базисе по формуле $\widetilde{P}_q = B^{-1}P_q$. Если среди координат вектора $\widetilde{P}_q$ нет положительных, то задача дробно-линейного программирования имеет асимптотическое решение. Если же есть положительные, то вектор P_q вводят в базис и по правилам симплекс-метода находят разрешающую строку и новый опорный план, а также новую матрицу $\widetilde{B}^{-1}$, обратную матрице $\widetilde{B}$, составленную из координат векторов базиса и переходят к п. 4 для проверки полученного опорного плана на оптимальность.

Этапы 3–7 составляют одну итерацию (один шаг) модифицированного симплекс-метода.

Следует отметить, что если имеется оптимальный базис B, и B^{-1} обратная к матрице соответствующая этому базису, то оптимальное решение может быть определено по формуле

$$x^* = B^{-1} \cdot b,$$

где b – первоначальный вектор правых частей системы ограничений.

Таким образом, особенности применения методов линейного программирования при решении задач дробно-линейного программирования состоят только в вычислении оценок Δ_j для проверки критерия оптимальности текущего базиса. Все остальные вычисления проводятся по обычным схемам симплексного метода (прямого или модифицированного).

4.2.2. Метод "линеаризации" решения задач дробно-линейного программирования

Идея метода "линеаризации" состоит в том, что на основе предположения $D(x) > 0$ задача дробно-линейного программирования (4.11)–(4.13) сводится к решению задачи линейного программирования, к которой добавляется одна дополнительная переменная и одно ограничение.

Рассмотрим задачу дробно-линейного программирования (4.11)–(4.13), для которой $R(x)$ ограниченный многогранник, а $D(x) > 0$ для любого $x \in R(x)$. Введя новую положительную переменную

$$r = 1/D(x) \tag{4.15}$$

и неотрицательные переменные

$$y_j = rx_j \quad (j = \overline{1,n}) \tag{4.16}$$

задачу (4.11)–(4.13) запишем в виде

$$Z(y,r) = \sum_{j=1}^{n} c_j y_j + c_0 r \to \max, \tag{4.17}$$

$$\sum_{j=1}^{n} a_{ij} y_j - b_i r \le 0 \quad (i = \overline{1,m}), \tag{4.18}$$

$$\sum_{j=1}^{n} d_j y_j + d_0 r = 1, \tag{4.19}$$

$$y_j \ge 0 \quad (j = \overline{1,n}), \tag{4.20}$$

$$r \ge 0. \tag{4.21}$$

Из задачи (4.17)–(4.21) следует, что для любого допустимого плана (y,r) величина $r > 0$ поэтому в оптимальном плане (y^*,r^*) получим $r^* > 0$. Имеет место следующая теорема.

Теорема 4.7. *Если* (y^*, r^*) *является оптимальным решением линейной задачи* (4.17)–(4.21), *то план* $x^* = \{x_j^* = y_j^*/r^* \quad (j = \overline{1,n})\}$ *оптимален для дробно-линейной задачи* (4.11)–(4.13).

Следует отметить, что оптимальные значения функций цели задач (4.11)–(4.13) и (4.17)–(4.21) совпадают, т.е. имеем

$$F(x^*) = Z(y^*, r^*).$$

Рассмотрим некоторую особенность применения метода "линеаризации". Как правило, в практических задачах знаменатель $D(x)$ принимает достаточно большие значения для допустимых планов задачи (4.11)–(4.13) и поэтому переменная r принимает малые значения (близкие к нулю). Поэтому в таких задачах формулу (4.15) целесообразно изменить и переменную r определять по формуле $r = \gamma/D(x) - \varepsilon$, где γ и ε – положительные числа, значения которых подобраны таким образом, чтобы выполнялось условие $r \geq 0$ для любых значений $x \in R(x)$. Определяя переменные y_j по формуле $y_j = x_j(r + \varepsilon) \quad (j = \overline{1,n})$ задачу (4.17)–(4.21) запишем в виде

$$\sum_{j=1}^{n} c_j y_j + c_0 r + c_0 \varepsilon \to \max,$$

$$\sum_{j=1}^{n} a_{ij} y_j - b_i r \leq \varepsilon b_i \quad (i = \overline{1,m}),$$

$$\sum_{j=1}^{n} d_j y_j + d_0 r = \gamma - \varepsilon d_0,$$

$$r_j \geq 0, \ y_j \geq 0 \quad (j = \overline{1,n}).$$

Применение метода "линеаризации" для решения практических задач дробно-линейного программирования на ЭВМ позволяет использовать программное обеспечение линейного программирования.

Рассмотрим двойственную задачу для случая сведения задачи дробно-линейного программирования к эквивалентной линейной задачи. Пусть исходная задача (4.11)–(4.13) сведена к задаче линейного программирования вида (4.17)–(4.21). Построим по общим правилам линейного программирования двойственную к задаче (4.17)–(4.21). Введем двойственные переменные $u_i \quad (i = \overline{1,m})$ для ограничений (4.18) и переменную λ для ограничения (4.19). Тогда получим двойственную задачу

$$W = \lambda \to \min, \tag{4.22}$$

$$\sum_{i=1}^{m} a_{ij} u_i + d_j \lambda \geq c_j \quad (j = \overline{1, n}), \tag{4.23}$$

$$-\sum_{i=1}^{m} b_i u_i + d_0 \lambda \geq c_0, \tag{4.24}$$

$$u_i \geq 0 \quad (i = \overline{1, m}). \tag{4.25}$$

Для взаимно двойственных задач линейного программирования (4.17)–(4.21) и (4.22)–(4.25) имеют место основные теоремы двойственности.

Теорема 4.8. *Если* (y, r) *– любой допустимый план прямой задачи* (4.17)–(4.21)*, а* (u, λ) *– любой допустимый план двойственной задачи* (4.22)–(4.25)*, то имеет место неравенство*

$$Z(y, r) \leq \lambda.$$

Теорема 4.9. *Если* (y^*, r^*) *и* (u^*, λ^*) *– допустимые планы соответственно для прямой и двойственной задач и* $Z(y^*, r^*) = \lambda^*$*, то* (y^*, r^*) *– оптимальный план прямой задачи* (4.17)–(4.21)*, а* (u^*, λ^*) *– оптимальный план двойственной задачи* (4.22)–(4.25)*, и обратно.*

Теорема 4.10. *Для того, чтобы допустимые решения* (y^*, r^*) *прямой задачи* (4.17)–(4.21)*, и* (u^*, λ^*) *для двойственной задачи* (4.22)–(4.25) *были оптимальными, необходимо и достаточно выполнение следующих двух групп равенств:*

(I) *– для неизвестных прямой задачи и ограничений двойственной задачи*

$$y_j^* \cdot \Big(\sum_{i=1}^{m} a_{ij} u_i^* + d_j \lambda^* - c_j \Big) = 0 \quad (j = \overline{1, n}),$$

$$r^* \cdot \Big(-\sum_{i=1}^{m} b_i u_i^* + d_0 \lambda^* - c_0 \Big) = 0,$$

(II) *– для переменных двойственной задачи и ограничений прямой задачи*

$$u_i^* \cdot \Big(\sum_{j=1}^{n} a_{ij} y_j^* - b_i r^* \Big) = 0 \quad (i = \overline{1, m}),$$

$$\lambda^* \cdot \Big(\sum_{j=1}^{n} d_j y_j^* + d_0 r^* - 1 \Big) = 0.$$

4.2.3. Параметрический метод решения задач дробно-линейного программирования

При применении метода "линеаризации" для решения задач дробно-линейного программирования к ограничениям задачи добавляется одна связующая переменная и одно связующее ограничение, что неэффективно в тех случаях, когда система ограничений задачи имеет специфическую структуру. В таких случаях целесообразно использовать параметрические методы решения задач дробно-линейного программирования, суть которых заключается в том, что исходная задача сводится к решению конечной последовательности линейных задач на множестве ограничений исходной задачи.

Ниже приводим один из алгоритмов параметрического метода. Рассмотрим задачу дробно-линейного программирования (4.1)–(4.3), для которой $D(x) > 0$ для любого $x \in R(x)$ и $R(x)$ – ограниченное множество. Тогда минимальное и максимальное значение функции $F(x)$ достигается в крайних точках многогранника $R(x)$. Таким образом целевая функция $F(x)$ ограничена снизу и сверху на выпуклом многограннике $R(x)$, т.е. существуют такие значения $\underline{\lambda}$ и $\overline{\lambda}$, для которых $\underline{\lambda} \leq F(\underline{x}) \leq F(x) \leq F(\overline{x}) \leq \overline{\lambda}$, где $\underline{x}$ и $\overline{x}$ – оптимальные планы задачи дробно-линейного программирования, в которых достигаются минимальное и максимальное значения функции $F(x)$.

Рассмотрим следующую задачу параметрического линейного программирования:

$$\max_{x \in R(x)} \{Z(x,\lambda) = C(x) - \lambda D(x)\}, \tag{4.26}$$

где $\lambda \in [\underline{\lambda}, \overline{\lambda}]$.

Алгоритм параметрического линейного программирования позволяет решить задачу (4.26) за конечное число итераций, т.е. находится конечная последовательность опорных планов $x^1, x^2, \ldots, x^p$ и соответствующие им интервалы оптимальности $[\underline{\lambda}, \lambda_1], [\lambda_1, \lambda_2], \ldots, [\lambda_p, \overline{\lambda}]$, где $\lambda_r \quad (r = \overline{1,p})$ – критические точки функции $f(\lambda)$, которая определяется по формуле

$$f(\lambda) = \max_{x \in R(x)} Z(x,\lambda). \tag{4.27}$$

Теорема 4.11. *Пусть $D(x) > 0$ для любого $x \in R(x)$. Тогда функция $f(\lambda)$, определенная по формуле (4.27), является кусочно-линейной, выпуклой и монотонно убывающей.*

Замечание 4.1. *Оптимальным решением задачи* (4.1)–(4.3) *является опорный план* x^r, *для которого* $Z(x, \mu) = 0$ (*т.е.* $F(x^r) = \mu$), *где* μ – *корень уравнения* $f(\lambda) = 0$.

Так как функция $f(\lambda)$ является убывающей, то найдется такое значение $\mu \in [\underline{\lambda}, \overline{\lambda}]$, которое представляет собой корень уравнения $f(\lambda) = 0$. На этой основе можно сформулировать следующую теорему, являющуюся признаком оптимальности опорного плана x задачи (4.1)–(4.3).

Теорема 4.12. *Для того чтобы оптимальный план* x^r *задачи* (4.26) *был оптимальным решением задачи* (4.1)–(4.3), *необходимо и достаточно, чтобы существовало такое значение* $\lambda = \mu \in [\lambda_r, \lambda_{r+1}]$, *для которого* $Z(x^r, \mu) = 0$.

Таким образом, решение задачи (4.1)–(4.3) сводится к нахождению корня уравнения $f(\lambda) = 0$. На основе теоремы 4.12 это эквивалентно нахождению опорного плана x^r задачи (4.26), для которого $Z(x^r, \mu) = 0$, а $\mu \in [\lambda_r, \lambda_{r+1}]$. Другими словами, решение задачи (4.1)–(4.3) сводится к последовательному решению параметрической задачи (4.26) с измененным значением параметра λ до тех пор, пока не найдется такой план x^r, для которого $\mu = F(x^r)$ является корнем уравнения $f(\lambda) = 0$.

Рассмотрим алгоритм параметрического метода нахождения оптимального решения задачи (4.1)–(4.3), т.е. корня уравнения $f(\lambda) = 0$. Так как корень этого уравнения принадлежит некоторому интервалу оптимальности, а соответствующий опорный план является оптимальным решением задачи (4.1)–(4.3), то идея такого алгоритма основана на том, что при последовательном решении задачи (4.26) в качестве значения параметра λ берется корень уравнения, соответствующий линейной функции опорного плана, и проверяется, принадлежит ли этот корень интервалу оптимальности. Однако эту проверку можно не проводить, если алгоритм нахождения корня уравнения $f(\lambda) = 0$ построить по следующей схеме.

Пусть имеем некоторый оптимальный план x^r задачи (4.26), полученный при фиксированном значении параметра λ. Тогда находим корень уравнения $f_r(\lambda) = Z(x^r, \lambda) = 0$, который определяется по формуле $\lambda = \mu_r = F(x^r)$. Графически это означает, что находим точку пересечения оси λ с касательной графика функции $f_r(\lambda)$ в некоторой точке отрезка, определенного интервалом оптимальности опорного плана x^r. Далее подставляя значение $\lambda = \mu_r$ в задачу (4.26), находим ее оптимальное решение. Тогда если μ_r принадлежит интервалу оптимальности опорного плана x^r, то решение задачи (4.26) не изменится, т.е. мы получили бы новый опорный план x^{r+1}, для которого точка $\lambda = \mu_r$ остается корнем уравнения $Z(x^{r+1}, \mu) = 0$. Таким образом, $\mu_r = \mu_{r+1}$ и $F(x^r) = F(x^{r+1})$.

На основе сказанного изложенный алгоритм параметрического метода решения задач дробно-линейного программирования сводится к следующим шагам.

Предварительный (0-*й*) *шаг.* Берем $\lambda = \mu_0 = \underline{\lambda}$, где $\underline{\lambda}$ – нижнее ограничение функции $F(x)$ на выпуклом многогранном множестве $R(x)$. Находим оптимальное решение задачи линейного программирования

$$\max_{x \in R(x)} \sum_{j=1}^{n} q_j x_j, \tag{4.28}$$

где $q_j = c_j - \lambda d_j \quad (j = \overline{1, n})$.

Общий(k-*й*) *шаг.* Пусть имеем некоторый оптимальный план x^{k-1} задачи (4.28), полученный на предыдущем шаге алгоритма.

Тогда:

- определяем значение параметра $\lambda = \mu_k = F(x^{k-1})$, т.е. находим точку M_0 пересечения оси λ с графиком функции $f_{k-1}(\lambda) = Z(x^{k-1}, \lambda)$;
- находим оптимальное решение x^k задачи (4.28) при $\lambda = \mu_k$;
- если $F(x^k) = F(x^{k-1})$, то план x^k является оптимальным решением задачи (4.1)–(4.3), если $F(x^k) \neq F(x^{k-1})$, то повторяем этот шаг, исходя из плана x^k.

Имеют место следующие утверждения:

Теорема 4.13. *Пусть x^{k-1} и x^k – два последовательных плана задачи* (4.26), *для которых $\mu_k = F(x^k) = F(x^{k-1})$. Тогда эти планы являются оптимальными для задачи* (4.1)–(4.3), *и если μ_k не является критической точкой функции $f(\lambda)$, то они совпадают.*

Теорема 4.14. *Пусть $\mu_0, \mu_1, \ldots, \mu_{k-1}, \mu_k$ – последовательность точек, полученных в результате работы алгоритма, а планы $x^0, x^1, \ldots, x^{k-1}, x^k$ – соответственно оптимальные решения задачи* (4.28) *при $\lambda = \mu_r$ $(r = \overline{1, k})$. Тогда $F(x^0) < F(x^1) < \cdots < F(x^{k-1}) = F(x^k)$.*

Теорема 4.15. *Последовательный процесс решения задачи* (4.1)–(4.3) *по алгоритму параметрического метода является конечным, и за конечное число шагов получим значение $\lambda = \mu_k \in [\lambda_k, \lambda_{k+1}]$, для которого $Z(x^k, \mu_k) = 0$.*

Замечание 4.2. *В качестве первоначального значения параметра* λ *может быть выбрано любое значение* $\mu_0 \in [\underline{\lambda}, \overline{\lambda}]$. *Если* μ_0 *находится справа от корня уравнения* $f(\lambda) = 0$, *то следующее значение* μ_1 *будет больше значения* μ_0, *а все последующие* μ_r $(r = \overline{1,k})$ *монотонно убывают, т.е.* $\mu_0 > \mu_1 < \mu_2 < \cdots < \mu_k$.

Как уже отмечалось, в отличие от других методов параметрический метод решения задач дробно-линейного программирования сохраняет структуру системы ограничений. Это очень важно, когда система ограничений имеет специальную структуру, например блочно-диагональную.

Рассмотрим задачу дробно-линейного программирования:

$$F(x) = \frac{C(x)}{D(x)} = \frac{\sum_{j=1}^{n} c_j x_j}{\sum_{j=1}^{n} d_j x_j} \to \min, \tag{4.29}$$

$$B_j x_j = b_j \quad (j = \overline{1,n}), \tag{4.30}$$

$$x_j \geq 0 \quad (j = \overline{1,n}), \tag{4.31}$$

в которой c_j, d_j, x_j – n_j-мерные векторы, b_j – m_j-мерные векторы, а B_j – $m_j \times n_j$-мерные матрицы.

Система ограничений задачи (4.29)–(4.31) имеет блочно-диагональную структуру без связующих ограничений. Однако поскольку функция (4.29) не является сепарабельной, то задачу (4.29)–(4.31) невозможно решить для каждого блока в отдельности. Если же для ее решения применить параметрический метод, то на каждом шаге задачу минимизации функционала $z(x, \lambda) = C(x) - \lambda D(x)$ при ограничениях (4.30), (4.31) и фиксированном λ можно решить для каждого блока в отдельности.

4.2.4. Полиномиальные алгоритмы решения задач дробно-линейного программирования

Первый полиномиальный алгоритм, основанный на модификациях метода эллипсоидов, для решения задач линейного и дробно-линейного программирования был предложен Л.Г. Хачианом в работах [293, 705, 706]. Однако на большинстве практических задач этот алгоритм уступает по быстродействию симплекс-методу. Поэтому достаточно остро стоял вопрос о построении алгоритма решения задач линейного программирования, конкурентоспособного по отношению с симплекс-методом. Таким алгоритмом стал полиномиальный алгоритм Кармаркара [285] и его дальнейшие модификации [22]–[26], [35, 42, 210, 218, 220, 227, 254, 286, 287, 303, 377, 390, 416, 420, 520, 648]. Близкий алгоритм к одной из модификаций [35] алгоритма Кармаркара был предложен И.И. Дикиным [610] еще в 1967 г.

Одна из версий алгоритма Кармаркара была использована для разработки полиномиального алгоритма решения задач дробно-линейного программирования [24]. Далее мы приведем две новые модификации полиномиальных алгоритмов Кармаркара-Барнеса-Дикина для решения задач дробно-линейного программирования [733].

Рассмотрим следующую задачу дробно-линейного программирования:

$$F(x) = \frac{C(x)}{D(x)} = \frac{\sum\limits_{j=1}^{n} c_j x_j}{\sum\limits_{j=1}^{n} d_j x_j} \to \min, \tag{4.32}$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad (i = \overline{1,m}), \tag{4.33}$$

$$x_j \geq 0 \quad (j = \overline{1,n}). \tag{4.34}$$

Введем обозначения: $x = (x_1, x_2, \ldots, x_n)$; $c = (c_1, c_2, \ldots, c_n)$; $d = (d_1, d_2, \ldots, d_n)$; $b = (b_1, b_2, \ldots, b_m)$; $A = \{a_{ij}\}$; $R(x) = \{x\colon Ax = b, x \geq 0\}$.

Предположим, что $D(x) > 0$ для любого $x \in R(x)$. Функция Лагранжа имеет вид

$$L(x, \lambda) = \frac{C(x)}{D(x)} - (\lambda, Ax - b), \tag{4.35}$$

где $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ – множители Лагранжа.

Пусть $y = (y_1, y_2, \ldots, y_n)^T$ – допустимое решение задачи (4.32)–(4.34), для которого $y_j > 0$ $(j = \overline{1,n})$. Если $0 < R < 1$, то эллипсоид

$$\sum_{j=1}^{n} \frac{(x_j - y_j)^2}{y_j^2} \leq R^2$$

принадлежит положительному ортанту в E^n.

Рассмотрим задачу:

$$F(x) = \frac{C(x)}{D(x)} \to \min, \tag{4.36}$$

$$Ax = b, \tag{4.37}$$

$$\sum_{j=1}^{n} \frac{(x_j - y_j)^2}{y_j^2} \le R^2. \tag{4.38}$$

Пусть $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ – множители Лагранжа ограничений (4.37), а $B = diag(y_1, y_2, \ldots, y_n)$. Для любого x, удовлетворяющего ограничениям (4.37), имеет место

$$F(y) - F(x) = \frac{C(y)}{D(y)} - \frac{C(x)}{D(x)} =$$

$$= \frac{C(y)D(x) - D(y)C(x) - A^T\lambda(x-y)}{D(x)D(y)} =$$

$$= \frac{(C(y)d - D(y)c - A^T\lambda)^T(x-y)}{D(x)D(y)} =$$

$$= \frac{||B(C(y)d - D(y)c - A^T\lambda)||}{D(x)D(y)} ||B^{-1}(x-y)||. \tag{4.39}$$

Выражение (4.39) выполняется в виде строгого равенства, если выполняется равенство

$$\frac{B(C(y)d - D(y)c - A^T\lambda)}{D(x)D(y)} = \gamma B^{-1}(x-y) \tag{4.40}$$

для некоторого γ и если $||B^{-1}(x-y)|| = R$.

Находим

$$\gamma = \frac{||B(C(y)d - D(y)c - A^T\lambda)||}{RD(x)D(y)}$$

и подставляем в (4.40)

$$\frac{||B(C(y)d - D(y)c - A^T\lambda)||}{D(x)D(y)} = \frac{||B(C(y)d - D(y)c - A^T\lambda)||}{RD(x)D(y)} B^{-1}(x-y).$$

Тогда

$$x = y - R\frac{B^2(C(y)d - D(y)c - A^T\lambda)}{||B(C(y)d - D(y)c - A^T\lambda)||}.$$

Так как $Ax = Ay = b$, то из (4.40) получим

$$\lambda B^2(C(y)d - D(y)c - A^T\lambda) = 0,$$

откуда

$$\lambda = (AB^2A^T)^{-1}AB^2(C(y)d - D(y)c).$$

Из (4.39) получим

$$F(x) \geq F(y) - R\frac{||B(C(y)d - D(y)c - A^T\lambda)||}{D(x)D(y)}.$$

Рассмотрим две вычислительные схемы полиномиального алгоритма типа Дикина-Барнеса решения задачи дробно-линейного программирования.

Алгоритм Барнеса. *Первоначальный* 0*-й шаг.* Пусть $x^0 > 0$, удовлетворяющее $Ax^0 = b$, будет первоначальным допустимым решением задачи (4.32)–(4.34).

Общий k-й шаг. Пусть x^k – некоторое решение, полученное на предыдущем шаге. Тогда:

- определяем $B_k = diag(x_1^k, x_2^k, \ldots, x_n^k)$,
- находим $\lambda_k = (AB_k^2A^T)^{-1}AB_k^2(C(x^k)d - D(x^k)c)$,
- вычисляем новые значения $x^{k+1} > 0$ по формуле

$$x^{k+1} = x^k - R\frac{B_k^2(C(y^k)d - D(x^k)c - A^T\lambda_k)}{||B_k(C(x^k)d - D(x^k)c - A^T\lambda_k)||}.$$

Алгоритм Дикина. *Первоначальный* 0*-й шаг.* Пусть $x^0 > 0$, удовлетворяющее $Ax^0 = b$, будет первоначальным допустимым решением задачи (4.32)–(4.34).

Общий k-й шаг. Пусть x^k – некоторое решение, полученное на предыдущем шаге. Тогда:

- находим направление спуска $\{u_i^k\}$, соответствующее множителям Лагранжа для функции

$$\frac{C(x)}{D(x)} + \sum_{i=1}^m u_i\left(b_i - \sum_{j=1}^n a_{ij}x_j\right) + v\left(\sum_{j=1}^n \frac{(x_j - y_j)^2}{y_j^2} - 1\right)\Big/2$$

  из решения системы уравнений

$$\sum_{t=1}^m b_{st}^k u_t = p_s^t \quad (s = \overline{1,m}),$$

  где

$$b_{st}^k = \sum_{j=1}^{n} (x_j^k)^2 a_{sj} a_{tj} \quad (s = \overline{1,m};\ t = \overline{1,m}),$$

$$p_s^k = \sum_{j=1}^{n} (x_j^k)^2 a_{sj} q_j^k \quad (s = \overline{1,m}),$$

$$q_j^k = c_j D(x^k) - d_j C(x^k) \quad (j = \overline{1,n}),$$

- вычисляем величины

$$\delta_j^k = \sum_{j=1}^{k} a_{ij} u_i - q_j^k \quad (j = \overline{1,n}),$$

$$z_j^k = (x_j^k)^2 \delta_j^k \quad (j = \overline{1,n}),$$

$$\Phi_k = \sum_{j=1}^{n} (x_j^k)^2 (\delta_j^k)^2 \quad (j = \overline{1,n}),$$

- выбираем величину шага λ_k по формуле

$$\lambda_k = \min(\rho\mu_k, \lambda_k'),$$

где

$$\lambda_k' = 1/\sqrt{\Phi_k},\ 0.5 \le \rho < 1,$$

а

$$\mu_k = \min_{j \in J} \frac{x_j^k}{z_j^k},$$

J – множество индексов j, для которых $\delta_j^k < 0$,

- вычисляем новые значения $x_j^{k+1} > 0$ по формуле

$$x_j^{k+1} = x_j^k + \lambda_k z_j^k \quad (j = \overline{1,n}),$$

- если $\sqrt{\Phi_k} < \varepsilon$, то процесс прекращается.

4.2.5. Субградиентный метод решения задач дробно-линейного программирования

Рассмотрим следующую задачу дробно-линейного программирования:

$$F(x)=\frac{C(x)}{D(x)}=\frac{\sum\limits_{j=1}^{n} c_j x_j}{\sum\limits_{j=1}^{n} d_j x_j}\to \min, \tag{4.41}$$

$$\sum_{j=1}^{n} a_{ij}x_j \le b_i \quad (i=\overline{1,m}), \tag{4.42}$$

$$x_j \ge 0 \quad (j=\overline{1,n}). \tag{4.43}$$

Обозначим через $R(x)$ многогранное выпуклое множество, заданное ограничениями (4.42), (4.43), т.е.

$$R(x)=\left\{x:\ \sum_{j=1}^{n} a_{ij}x_j \le b_i \quad (i=\overline{1,m});\ x_j\ge 0 \quad (j=\overline{1,n})\right\},$$

и предположим, что $R(x)$ ограничено.

Для решения задачи (4.41)–(4.43) используем субградиентный метод, в котором обобщенный градиент определяется относительно параметрической задачи

$$\max_{x\in R(x)} Z(x,\lambda)=\sum_{j=1}^{n}(c_j-\lambda d_j)x_j.$$

Тогда на $(k+1)$-м шаге значения обобщенного градиента в точке $x^k=\{x_j^k\}$ определяются по формуле

$$g_j(x^k)=\begin{cases} c_j-\lambda_k d_j, & \text{если}\ \ \Delta_{i_0}\le 0 \quad (j=\overline{1,n});\\ a_{i_0 j}, & \text{если}\ \ \Delta_{i_0}>0 \quad (j=\overline{1,n}),\end{cases}$$

где $\Delta_{i_0}=\max\limits_{1\le i\le m}\Delta_i=\max\limits_{1\le i\le m}\left\{\sum\limits_{j=1}^{n} a_{ij}x_j^k-b_i\right\}$, а

$$\lambda_k=\begin{cases} F(x^k), & \text{если}\ \ D(x^k)>0;\\ 0, & \text{в противном случае.}\end{cases}$$

Алгоритм решения задачи (4.41)–(4.43) методом обобщенного градиента состоит в следующем. Пусть на $(k+1)$-м шаге имеем точку $\{x_j^k\}$. Тогда:

- вычисляем значение функционала (4.41) в точке $\{x_j^k\}$, т.е. находим

$$\lambda_k = \begin{cases} F(x^k), & \text{если} \quad D(x^k) > 0; \\ 0, & \text{если} \quad D(x^k) \le 0, \end{cases}$$

- находим

$$\Delta_i = \sum_{j=1}^{n} a_{ij} x_j^k - b_i \quad (i = \overline{1,m}),$$

  т.е. в заданной точки $\{x_j^k\}$ находим невязки каждого ограничения,

- находим значение Δ_{i_0} максимальной невязки и номер ограничения для которого оно достигается, т.е.

$$\Delta_{i_0} = \max_{1 < i \le m} \Delta_i,$$

- если $\Delta_{i_0} \le 0$, то это означает, что точка $\{x_j^k\}$ является допустимой для системы ограничений (4.42). В этом случае значение обобщенного градиента определяется по формуле

$$g_j(x^k) = c_j - \lambda_k d_j \quad (j = \overline{1,n}).$$

  Это означает, что в качестве направления движения в методе обобщенного градиента берется антиградиент целевой дробно-линейной функции, если $D(x^k) > 0$ и антиградиент линейной функции ограничения i_0, если $D(x^k) \le 0$, т.е. происходит движение в сторону уменьшения значения линейной или дробно-линейной функции,

- если $\Delta_{i_0} > 0$, то это означает, что точка $\{x_j^k\}$ не является допустимой для системы ограничений (4.42). В этом случае значение обобщенного градиента определяется по формуле

$$g_j(x^k) = a_{i_0 j} \quad (j = \overline{1,n}).$$

  Это означает, что в качестве направления движения в субградиентном методе берется антиградиент ближайшего ограничения i_0, т.е. в данном случае происходит движение в сторону допустимой области,

- переходим к новой точке $\{x_j^{k+1}\}$ по формуле

$$x_j^{k+1} = \max\{0,\, x_j^k - h_{k+1} g_j(x^k)\} \quad (j = \overline{1,n}),$$

где h_{k+1} – величина шага.

В заключение можно отметить, что при решении различных практических задач дробно-линейного программирования было использовано программное обеспечение, реализующее параметрический метод и метод "линеаризации" с дальнейшем применением программ линейного программирования. Полиномиальные алгоритмы решения задач дробно-линейного программирования были реализованы с целью проведения экспериментальных расчетов.

4.3. Задачи дробно-линейного программирования блочной структуры

4.3.1. Блочная задача дробно-линейного программирования

Рассмотрим задачу дробно-линейного программирования, в которой система ограничений имеет блочно-диагональную структуру:

$$F(x) = \frac{C(x)}{D(x)} = \frac{\sum_{j=1}^{p} c_j x_j}{\sum_{j=1}^{p} d_j x_j} \to \min, \tag{4.44}$$

$$\sum_{j=1}^{p} A_j x_j \le b, \tag{4.45}$$

$$B_j x_j \le b_j \quad (j = \overline{1,p}), \tag{4.46}$$

$$x_j \ge 0 \quad (j = \overline{1,p}), \tag{4.47}$$

где c_j, d_j, x_j – векторы размерности n_j; b – m-мерный вектор; b_j – векторы размерности m_j; A_j – матрицы размерности $m \times n_j$; B_j – матрицы размерности $m_j \times n_j$.

Задача (4.44)–(4.47) имеет блочно-диагональную структуру со связующими ограничениями. Предположим, что в задаче имеются достаточно большое количество блоков, а также размерности каждого блока или структура его ограничений, позволяет решать эффективно возникающие "блоковые" подзадачи.

Для решения блочной задачи дробно-линейного программирования (4.44)–(4.47) большой размерности предлагается использовать декомпозиционные методы. Разработаны четыре класса декомпозиционных методов:

- метод декомпозиции Данцига-Вулфа;
- схема декомпозиции по ограничениям;
- схема декомпозиции по переменным;
- метод декомпозиции по ресурсам.

Рассмотрим каждый метод в отдельности.

4.3.2. Метод декомпозиции Данцига-Вулфа

Рассмотрим метод декомпозиции Данцига-Вулфа для решения задачи дробно-линейного программирования блочно-диагональной структуры. Обозначим через $R_j(x_j)$ множество допустимых решений каждого блока ограничений (4.46) т.е.

$$R_j(x_j) = \{x_j \in E^{n_j} : \ B_j x_j \leq b_j, \ x_j \geq 0\}.$$

Каждое множество $R_j(x_j)$ представляет собой многогранное множество в пространстве E^{n_j} и предположим, что оно ограничено и известны все его крайние точки. Обозначим через $x_j^r \ \ (r = \overline{1, k_j})$, вектор координатов всех крайних точек многогранника $R_j(x_j)$. Как известно, любая точка многогранника $R_j(x_j)$ может быть представлена в виде линейной выпуклой комбинации всех его крайних точек. Таким образом для каждого j-го блока ограничений (4.46)

$$x_j = \sum_{r=1}^{k_j} \lambda_j^r x_j^r \quad (r = \overline{1, k_j}), \tag{4.48}$$

$$\sum_{r=1}^{k_j} \lambda_j^r = 1, \ \lambda_j^r \geq 0 \quad (r = \overline{1, k_j}). \tag{4.49}$$

Подставим (4.48) в (4.44) и (4.45), а условия (4.49) добавляются вместо ограничений (4.46).

Тогда получим задачу:

$$\frac{\sum_{j=1}^{p}\sum_{r=1}^{k_j} c_j x_j^r \lambda_j^r}{\sum_{j=1}^{p}\sum_{r=1}^{k_j} d_j x_j^r \lambda_j^r} \to \min,$$

$$\sum_{j=1}^{p}\sum_{r=1}^{k_j} A_j x_j^r \lambda_j^r \leq b,$$

$$\sum_{r=1}^{k_j} \lambda_j^r = 1 \quad (j = \overline{1,p}),$$

$$\lambda_j^r \geq 0 \quad (j = \overline{1,p};\ r = \overline{1,k_j}).$$

Введем обозначения:

$f_j^r = A_j x_j^r$ – векторы-столбцы размерности m,

$\alpha_j^r = c_j x_j^r$ – коэффициенты числителя,

$\beta_j^r = d_j x_j^r$ – коэффициенты знаменателя.

Тогда в новых обозначениях получим следующую задачу дробно-линейного программирования:

$$\frac{\alpha(\lambda)}{\beta(\lambda)} = \frac{\sum_{j=1}^{p}\sum_{r=1}^{k_j} \alpha_j^r \lambda_j^r}{\sum_{j=1}^{p}\sum_{r=1}^{k_j} \beta_j^r \lambda_j^r} \to \min, \tag{4.50}$$

$$\sum_{j=1}^{p}\sum_{r=1}^{k_j} f_j^r \lambda_j^r \leq b, \tag{4.51}$$

$$\sum_{r=1}^{k_j} \lambda_j^r = 1 \quad (j = \overline{1,p}), \tag{4.52}$$

$$\lambda_j^r \geq 0 \quad (j = \overline{1,p};\ r = \overline{1,k_j}). \tag{4.53}$$

Для решения задачи дробно-линейного программирования (4.50)–(4.53) используем модифицированный симплекс-метод.

Пусть B – текущий базис; B^{-1} – соответствующая обратная матрица текущего базиса; λ_B – значение переменных базиса B; $\alpha(\lambda_B)$ и $\beta(\lambda_B)$ – соответственно значения числителя и знаменателя функционала (4.50) в текущем базисе. Следует заметить, что базис B имеет размерность $(m+p) \times (m+p)$.

Тогда на очередном шаге модифицированного симплекс-метода выполняем следующие вычисления:

1. Находим значения двойственных оценок ограничений (4.51) и (4.52) соответствующие числителю и знаменателю по формулам

$$u = \alpha_B B^{-1} \text{ и } v = \beta_B B^{-1},$$

где α_B и β_B – векторы размерности $m+p$ коэффициентов числителя и знаменателя соответствующие базисным переменным. Координаты векторов двойственных оценок разделены на две части $u = (u^1, u^2)$, $v = (v^1, v^2)$, где $u^1 = (u_1^1, u_2^1, \ldots, u_m^1)$, $v^1 = (v_1^1, v_2^1, \ldots, v_m^1)$ – соответствуют ограничениям (4.51), а $u^2 = (u_1^2, u_2^2, \ldots, u_p^2)$, $v^2 = (v_1^2, v_2^2, \ldots, v_p^2)$ – соответствуют ограничениям (4.52).

2. Проверяем критерий оптимальности текущего базиса B по формуле:

$$\Delta_j^r = \beta(\lambda_B)(\alpha_j^r - u^1 f_j^r) - \alpha(\lambda_B)(\beta_j^r - v^1 f_j^r) - \delta_j \geq 0 \quad (j = \overline{1,p};\ r = \overline{1,k_j}),$$

где

$$\delta_j = \alpha(\lambda_B) \cdot v_j^2 - \beta(\lambda_B) \cdot u_j^2 \quad (j = \overline{1,p}).$$

Как правило, в соответствие с модифицированным симплекс-методом, значения коэффициентов Δ_j^r необходимо вычислить для всех небазисных переменных, т.е. для всех крайних точек многогранников $R_j(x_j) \quad (j = \overline{1,p})$.

Однако в соответствии с декомпозиционным методом Данцига-Вулфа этого делать не обязательно. Для этого используется процедура генерации столбцов.

Для того, чтобы проверить критерий оптимальности $\Delta_j^r \geq 0$ находим

$$\min_{j,r} \Delta_j^r = \min_{j,r} \left\{\beta(\lambda_B)(\alpha_j^r - u^1 f_j^r) - \alpha(\lambda_B)(\beta_j^r - v^1 f_j^r) - \delta_j\right\} =$$

$$= \min_{j,r}\{\beta(\lambda_B) \cdot (c_j x_j^r - u^1 A_j x_j^r) - \alpha(\lambda_B) \cdot (d_j x_j^r - v^1 A_j x_j^r) - \delta_j\} =$$

$$= \min_{j,r}\{[\beta(\lambda_B) \cdot (c_j - u^1 A_j) - \alpha(\lambda_B) \cdot (d_j - v^1 A_j)] x_j^r - \delta_j\} =$$

$$= \min_{j}\{\min_{x_j \in R_j(x_j)} \Phi_j(x_j) - \delta_j\},$$

где

$$\Phi_j(x_j) = \beta(\lambda_B) \cdot (c_j - u^1 A_j) - \alpha(\lambda_B) \cdot (d_j - v^1 A_j).$$

Таким образом задача вычисления значений коэффициентов Δ_j^r сводится к решению p задач линейного программирования типа:

$$\Phi_j(x_j) \to \min, \tag{4.54}$$

$$B_j x_j \le b_j, \tag{4.55}$$

$$x_j \ge 0 \tag{4.56}$$

для каждого блока в отдельности.

Пусть оптимальное решение x_j^* задачи (4.54)–(4.56) соответствует некоторой крайней точке $x_j^{l_j}$, а $\Phi_j(x_j^{l_j})$ значение функционала (4.54) в этой точке.

Находим

$$\Delta_t^{l_t} = \min_j \{\Phi_j(x_j^{l_j}) - \delta_j\}.$$

Если $\Delta_t^{l_t} < 0$, то текущий базис является не оптимальным. Поэтому, переходим к новому базису, для чего в базис вводится вектор, который будет соответствовать новой переменной $\lambda_t^{l_t}$. С учетом значений координат крайней точки $x_t^{l_t}$, по соответствующим формулам определяются значения коэффициентов $\alpha_t^{l_t}, \beta_t^{l_t}$ и вектора $f_t^{l_t}$. Полученные коэффициенты $\alpha_t^{l_t}$ и $\beta_t^{l_t}$ используются в дальнейшем для вычисления значений числителя и знаменателя в новом базисе, а вектор $(f_t^{l_t}, 0, 0, \ldots, 1, \ldots, 0)$, где значение 1 соответствует t-му блоку (t-му многограннику) для которого найдена оптимальная крайняя точка $x_t^{l_t}$, умноженный на обратную матрицу B^{-1} вводится в новый базис.

Вектор для вывода из базиса находится по общим правилам модифицированного симплекс-метода.

Если $\Delta_t^{l_t} \ge 0$, тогда текущий базис является оптимальным. Пусть λ^{r*} является оптимальным планом задачи (4.50)–(4.53). Тогда план $x^* = (x_1^*, x_2^*, \ldots, x_p^*)$, где

$$x_j^* = \sum_{r=1}^{k_j} \lambda_j^{r*} x_j^r \quad (j = \overline{1,p})$$

будет оптимальным решением задачи (4.44)–(4.47).

Следует отметить, что так как для оптимального плана $\{\lambda^{r*}\}$ выполняются условия

$$\sum_{r=1}^{k_j} \lambda_j^{r*} = 1 \quad (j = \overline{1,p})$$

и $\lambda_j^{r*} \ge 0 \quad (j = \overline{1,p};\ r = \overline{1,k_j})$, то в оптимальном решении задачи (4.50)–(4.53) будет входить по крайней мере по одной крайней точки из каждого многогранника $R_j(x_j)$. Данная крайняя точка и будет соответствовать оптимальным значениям компонент плана $\{x_j^*\}$ для j-го блока. Если же для некоторого многогранника j в оптимальном плане $\{\lambda_j^{r*}\}$ будет входить несколько крайних точек, т.е. $\lambda_j^{r*} > 0$ и $\lambda_j^{r*} \neq 1$, для некоторых значений индексов r, то соответствующая линейная комбинация этих крайних точек и будет давать нам решения x_j^* для i-го блока.

4.3.3. Схема декомпозиции по ограничениям

Для решения задачи дробно-линейного программирования (4.44)–(4.47) используем схему разложения по ограничениям. Построим функцию Лагранжа на множестве ограничений (4.45)

$$L(x,u) = \left(C(x) + \left(u, \sum_{j=1}^{p} A_j x_j - b\right)\right) \Big/ D(x)$$

и рассмотрим задачу

$$\max_{u \geq 0} \{L^*(u) = \min_{x \in R(x)} L(x,u)\}, \tag{4.57}$$

где $u = (u_1, ..., u_m)$ – множители Лагранжа для ограничений (4.45), а $R(x)$ – многогранное выпуклое множество значений переменных x, удовлетворяющих ограничениям (4.46), (4.47).

Тогда решение задачи (4.44)–(4.47) сведется к решению задачи (4.57). На $(t+1)$-м шаге алгоритма обобщенного градиентного метода решения задачи (4.57) необходимо выполнить три этапа схемы декомпозиции по ограничениям.

1. Решить задачу
$$\min_{x \in R(x)} L(x,u) \tag{4.58}$$
при фиксированных значениях $u = u^t$. Для решения задачи (4.58) применим параметрический метод. Рассмотрим следующую блочную задачу параметрического линейного программирования:
$$\sum_{j=1}^{p} (c_j + u^t A_j - \lambda d_j) x_j \to \min,$$
$$B_j x_j = b_j \quad (j = \overline{1,p}),$$
$$x_j \geq 0 \quad (j = \overline{1,p}).$$
Тогда при фиксированном значении параметра $\lambda = \lambda^t$ решаем p задач линейного программирования:

$$q_j x_j \to \min, \tag{4.59}$$

$$B_j x_j = b_j, \tag{4.60}$$

$$x_j \geq 0, \tag{4.61}$$

где $q_j = c_j + u^t A_j - \lambda^t d_j \quad (j = \overline{1,p})$; $\lambda^t = L(x^*(u^{t-1}), u^t)$. Значение параметра λ определяется с учетом полученного на предыдущем шаге субградиентного метода решения задачи (4.58) $x^*(u^t) = \{x_j^*(u^t)\}$.

2. Определить значения обобщенного градиента функции $L^*(u)$ в точке u^t по формуле

$$G(u^t) = \left\{ \Big(\sum_{j=1}^{p} A_j x_j^*(u^t) - b \Big) \Big/ D(x^*(u^t)) \right\},$$

где $x^*(u^t)$ – оптимальное решение задачи (4.58).

3. Найти новые значения

$$u^{t+1} = \max\{0, u^t + h_{t+1} G(u^t)\},$$

где h_{t+1} – величина шага.

Использование такого алгоритма при решении блочных задач дробно-линейного программирования достаточно эффективно, когда каждый блок ограничений (4.46) имеет специальную структуру, например транспортный вид. Тогда для решения задач (4.59)–(4.61) можно использовать специальные алгоритмы.

4.3.4. Схема декомпозиции по переменным

Рассмотрим задачу дробно-линейного программирования, которая имеет блочную структуру со связующими переменными

$$F(x,y) = \frac{C(x,y)}{D(x,y)} = \frac{\sum\limits_{j=1}^{p} c_j x_j + \alpha y}{\sum\limits_{j=1}^{p} d_j x_j + \beta y} \to \min, \tag{4.62}$$

$$B_j x_j + F_j y = b_j \quad (j = \overline{1,p}), \tag{4.63}$$

$$x_j \geq 0 \quad (j = \overline{1,p}), \tag{4.64}$$

$$y \geq 0, \tag{4.65}$$

где c_j, d_j, x_j – векторы размерности n_j; α, β, y – векторы размерности k; b_j – векторы размерности m_j; B_j – матрицы размерности $m_j \times n_j$; F_j – матрицы размерности $m_j \times k$.

Для решения задачи (4.62)–(4.65) применяем схему декомпозиции по переменным. Для этого при фиксированных значениях переменных $y = \overline{y}$ рассматриваем задачу:

$$\min_{\overline{y} \geq 0} \Phi(\overline{y}), \tag{4.66}$$

где

$$\Phi(\overline{y}) = \min_{x \in R(\overline{y})} F(x, \overline{y}), \tag{4.67}$$

а $R(\overline{y})$ – множество допустимых значений ограничений (4.63)–(4.65) при фиксированных $y = \overline{y}$.

Таким образом задача (4.62)–(4.65) сводится к минимизации квазивыпуклой функции $\Phi(\overline{y})$. Так как $\Phi(\overline{y})$ является недифференцируемой, то для решения задачи (4.66) используем обобщенный градиентный метод.

Пусть на $(t+1)$-м шаге обобщенного градиентного метода имеем значения переменных $\overline{y}^t$ и решение $x^*(y^{t-1})$ задачи (4.67), найденное при фиксированных значениях переменных $\overline{y} = \overline{y}^{t-1}$. Тогда выполняем три этапа схемы декомпозиции по переменным.

1. Решаем задачу (4.67) при фиксированных значениях переменных $\overline{y} = \overline{y}^t$. Так как задача (4.67) имеет дробно-линейный функционал и блочно-диагональные ограничения, то для ее решения параметрическим методом рассматриваем задачу

$$\sum_{j=1}^{p} (c_j - \lambda d_j) x_j \to \min, \tag{4.68}$$

$$B_j x_j = q_j(\overline{y}^t) \quad (j = \overline{1,p}), \tag{4.69}$$

$$x_j \geq 0 \quad (j = \overline{1,p}), \tag{4.70}$$

где $q_j(\overline{y}^t) = b_j - F_j \overline{y}^t \quad (j = \overline{1,p})$.

Если фиксировать значения параметра $\lambda = \lambda^t = F(x^*(\overline{y}^{t-1}), \overline{y}^t)$, то задача (4.68)–(4.70) сводится к решению p задач линейного программирования

$$(c_j - \lambda^t d_j) x_j \to \min, \tag{4.71}$$

$$B_j x_j = q_j(\overline{y}^t), \tag{4.72}$$

$$x_j \geq 0. \tag{4.73}$$

2. Пусть $x_j^*(\overline{y}^t)$ – решение задачи (4.71)–(4.73), а $\overline{u}_j^*(\overline{y}^t)$ – значение множителей Лагранжа, соответствующие ограничениям (4.72), для функции:

$$L(x,u) = \frac{C(x,\overline{y})}{D(x,\overline{y})} + \sum_{j=1}^{p} u_j(B_j x_j + F_j \overline{y} - b_j).$$

Тогда значения обобщенного градиента функции $\Phi(\overline{y})$ в точке $\overline{y} = \overline{y}^t$ определяем по формуле:

$$g(\overline{y}^t) = \frac{\alpha D(x^*(\overline{y}^t), \overline{y}^t) - \beta C(x^*(\overline{y}^t), \overline{y}^t)}{[D(x^*(\overline{y}^t), \overline{y}^t)]^2} - \sum_{j=1}^{p} u_j^*(\overline{y}^t) \cdot F_j.$$

3. Находим новое значение переменных $\overline{y}$ по формуле

$$\overline{y}^{t+1} = \max\{0, \overline{y}^t - h_{t+1} g(\overline{y}^t)\},$$

где h_{t+1} – величина шага в обобщенном градиентном методе.

4.3.5. Схема декомпозиции по ресурсам

Для решения задачи дробно-линейного программирования (4.44)–(4.47) применим схему декомпозиции по ресурсам. Для этого представим вектор ресурсов в виде суммы объемов их распределении по блокам:

$$b = z_1 + z_2 + \cdots + z_p, \ b = \sum_{j=1}^{p} z_j,$$

где z_j – вектор распределенных объемов ресурсов j-му блоку.

Тогда задачу (4.44)–(4.47) можно представить в виде двух задач дробного программирования. Первая координирующая задача, имеет вид:

$$\Phi(z) = \frac{\alpha(z)}{\beta(z)} = \frac{\sum\limits_{j=1}^{p} \alpha_j(z_j)}{\sum\limits_{j=1}^{p} \beta_j(z_j)} \to \min, \tag{4.74}$$

$$\sum_{j=1}^{p} z_j \leq b, \tag{4.75}$$

в которой значения функций $\alpha_j(z_j)$ и $\beta_j(z_j)$ находятся алгоритмическим путем решения второй задачи дробно-линейного программирования (локальной задачи):

$$F(x) = \frac{\sum_{j=1}^{p} c_j x_j}{\sum_{j=1}^{p} d_j x_j} \to \min, \tag{4.76}$$

$$A_j x_j \leq z_j \quad (j = \overline{1,p}), \tag{4.77}$$

$$B_j x_j \leq b_j \quad (j = \overline{1,p}), \tag{4.78}$$

$$x_j \geq 0,\ z_j \geq 0 \quad (j = \overline{1,p}). \tag{4.79}$$

Так как функционал $\Phi(z)$ является недифференцируемым, то для решения задачи (4.74), (4.75) применим субградиентный метод с проектированием субградиента на линейном многообразии.

Тогда на $(t+1)$-м шаге субградиентного метода необходимо выполнить три этапа схемы декомпозиции по ресурсам.

1. Решить задачу (4.76)–(4.79) при фиксированных значениях z_j^t. Для ее решения используем параметрический метод и с учетом полученного решения на предыдущем шаге субградиентного метода $x_j^*(z_j^{t-1})$ фиксируем значения параметра λ по формуле

$$\lambda_t = F(x^*(z_j^{t-1})) = \frac{\sum_{j=1}^{p} c_j x_j^*(z_j^{t-1})}{\sum_{j=1}^{p} d_j x_j^*(z_j^{t-1})}.$$

Тогда задача (4.76)–(4.79) сводится к решению p задач линейного программирования для каждого блока в отдельности:

$$(c_j - \lambda_t d_j) x_j \to \min, \tag{4.80}$$

$$A_j x_j \leq z_j^t, \tag{4.81}$$

$$B_j x_j \leq b, \tag{4.82}$$

$$x_j \geq 0. \tag{4.83}$$

Находим оптимальное решение $x_j^*(z_j^t)$ задачи (4.80)–(4.83) и значения двойственных оценок $\lambda_j(z_j^t)$ ограничений (4.81), которые соответствуют множителям Лагранжа функции

$$L(x, z, \lambda) = \frac{\sum\limits_{j=1}^{p} c_j x_j + \sum\limits_{j=1}^{p} \lambda_j (A_j x_j - z_j)}{\sum\limits_{j=1}^{p} d_j x_j}.$$

2. Находим значения обобщенного градиента функции $\Phi(z)$ в точке z_j^t по формуле
$$g_\Phi(z) = \{-\lambda_j(z_j^t)/\beta(z)\},$$
где $\beta(z) = \sum\limits_{j=1}^{p} d_j x_j^*(z_j^t)$.

3. Находим новые значения переменных z_j^{t+1} по формулам одного из субградиентных методов с проектированием субградиента на линейном многообразии.

Глава 5

Задачи дробно-выпуклого программирования

В данной главе рассматриваются различные задачи дробно-выпуклого программирования, для решения которых приводятся метод преобразования переменных, параметрический метод и методы нелинейной оптимизации квазивыпуклой функции.

Субградиентные методы используются для нахождения седловой точки дробно-выпуклой функции, а метод эллипсоидов в сочетании с параметрическим методом применяется для решения общей задачи дробно-выпуклого программирования. Также приводится негладкие штрафные функции для задач дробно-выпуклого программирования, которые сводят их к решению задач безусловной недифференцируемой оптимизации субградиентными методами.

Для решения общей задачи дробно-выпуклого программирования предложены три группы схем декомпозиции (по ограничениям, переменным и ресурсам), которые разработаны на основе функции Лагранжа и методов недифференцируемой оптимизации.

Для сепарабельных задач с дробно-выпуклыми функционалами и с блочной структурой предлагаются использовать схемы декомпозиции по ограничениям, переменным или по ресурсам, в результате применения которых для каждого блока в отдельности решаются задачи выпуклого программирования.

5.1. Методы решения задач дробно-выпуклого программирования

Рассмотрим задачу дробно-выпуклого программирования:

$$f_0(x) = \frac{\varphi(x)}{\psi(x)} \ \to \ \min, \tag{5.1}$$

$$f_i(x) \leq 0 \quad (i = \overline{1, m}), \tag{5.2}$$

где $x \in R^n$, а функции

$$\varphi(x),\ \psi(x),\ f_i(x) \quad (i = \overline{1, m})$$

– непрерывны и дифференцируемы. Пусть $S(x)$ – множество значений переменных x, удовлетворяющих ограничениям (5.2), и предположим, что

$$\varphi(x),\ -\psi(x),\ f_i(x) \quad (i = \overline{1, m})$$

являются выпуклыми на множестве $S(x)$.

Не ограничивая общности, предположим, что $\varphi(x) \geq 0$, а $\psi(x) > 0$ для любого $x \in S(x)$.

Тогда функция $f_0(x)$ является квазивыпуклой, и задача (5.1), (5.2) заключается в нахождении некоторой точки $x \in S(x)$, дающей минимальное значение квазивыпуклого функционала $f_0(x)$.

Из сделанных предположений, следует, что если $S(x)$ не пустое множество, то функция $f_0(x)$ имеет ограниченное минимальное значение.

Методы решения задачи (5.1), (5.2) могут быть разбиты на следующие классы:

а) прямые методы решения задачи дробно-выпуклого программирования [37, 47, 48, 50, 151, 258, 266, 352, 439, 652], т.е. задача (5.1), (5.2) может быть рассмотрена как задача нелинейного программирования, и для ее решения используются общие методы нелинейной оптимизации без учета структуры функционала $f_0(x)$, или же как задача квазивыпуклого программирования [30, 94, 106, 181, 330, 406, 493, 494, 589, 629, 645, 695];

б) методы преобразования переменных [383], когда задача (5.1), (5.2) путем некоторой замены переменных сводится к задаче выпуклого программирования и для решения последней применяются общеизвестные методы;

в) параметрические методы [63, 172, 191, 260, 261, 362, 380, 436, 478], когда исходная задача сводится к решению и анализу некоторой задачи параметрического нелинейного программирования;

г) декомпозиционные методы и другие методы [107, 486, 496, 633, 667, 733], учитывающие специальную структуру системы ограничений, задающие множество $S(x)$.

Прямые методы. Для решения задачи дробно-выпуклого программирования (5.1), (5.2) можно использовать методы нелинейного программирования [34], [204]–[206], [272, 300, 367, 375, 390, 419, 633]. Как правило, это методы возможных направлений, градиентные и субградиентные методы, а также методы внутренних точек. В общем случае эти методы неэффективны. Что касается частных случаях задачи (5.1), (5.2), то для них предложены различные модификации общеизвестных методов. Как отмечено в предыдущей главе, для задачи дробно-линейного программирования предложены модификации симплекс-метода, метода разложения Данцига-Вулфа и другие. Для задач транспортного типа с дробно-линейным функционалом модифицирован метод потенциалов, как в матричной, так и в сетевой постановке [733].

Метод преобразования переменных. Приведем метод преобразования переменных [383] для решения задач дробного программирования.

Пусть $\alpha(t)$: $R^n \to R$ – монотонная, строго возрастающая функция, которая при $t > 0$ принимает положительные значения, т.е. $\forall\ t > 0$, $\alpha(t) > 0$.

Для решения задачи (5.1), (5.2) используется следующая процедура преобразования переменных

$$y = tx, \tag{5.3}$$

где $t \geq 0$ – некоторый параметр, выбранный таким образом, чтобы для заданной функции $\alpha(t)$ и некоторого $\gamma \in R$, $\gamma > 0$ выполнялось равенство

$$\psi(y/t)\alpha(t) = \gamma.$$

Пусть $\beta_i(t)$ $(i = \overline{1,m})$: $R^n \to R$ – m неотрицательных функций, т.е. для $t \geq 0$ имеем $\beta_i(t) \geq 0$ $(i = \overline{1,m})$.

Тогда, используя замену переменных (5.3), задача (5.1), (5.2) сводится к решению задачи нелинейного программирования:

$$F(y,t) = \varphi(y/t) \cdot \alpha(t) \to \min, \tag{5.4}$$

$$G(y,t) = \psi(y/t) \cdot \alpha(t) = \gamma, \tag{5.5}$$

$$H_i(y,t) = f_i(y/t) \cdot \beta_i(t) \leq 0 \quad (i = \overline{1,m}). \tag{5.6}$$

Задача (5.4)–(5.6) в общем случае не является задачей выпуклого программирования, так как имеется равенство (5.5) и множество допустимых значений не является выпуклым. Поэтому, вместо задачи (5.4)–(5.6) рассматривается следующая задача выпуклого программирования относительно переменных y и t, а именно

$$F(y,t) \to \min, \tag{5.7}$$

$$G(y,t) \le \gamma, \tag{5.8}$$

$$H_i(y,t) \le 0 \quad (i = \overline{1,m}). \tag{5.9}$$

Имеет место

Теорема 5.1. *Если* (y^*, t^*) *является оптимальным решением задачи* (5.7)–(5.9), *то* $x^* = y^*/t^*$ *является оптимальным решением задачи* (5.1), (5.2).

Аналогичный метод был предложен в работе [133] для решения задачи дробно-линейного программирования. Для данной задачи берутся $\alpha(t) = t$ и $\beta_i(t) = t \quad (i = \overline{1,m})$, $\gamma = 1$ и делается замена переменных $y_j = x_j \cdot t$, где

$$t = \frac{1}{\sum_{j=1}^{n} d_j x_j + d_0}.$$

Тогда получим следующую задачу линейного программирования:

$$\sum_{j=1}^{n} c_j y_j + c_0 t \ \to \min,$$

$$\sum_{j=1}^{n} d_j y_j + d_0 t = 1,$$

$$\sum_{j=1}^{n} a_{ij} y_j - b_i t \le 0 \quad (i = \overline{1,m}),$$

$$y_j \ge 0 \quad (j = \overline{1,n}), \ \ t > 0.$$

В результате ее решения симплекс-методом находим $x_j^* = y_j^*/t^*$ – оптимальное решение задачи дробно-линейного программирования.

Недостатком данного метода является введение дополнительного ограничения и дополнительной переменной, что отрицательно сказывается, когда первоначальная задача имеет специальную структуру системы ограничений.

Параметрические методы. Параметрические методы решения задач дробного программирования заключаются в том, что вместо исходной задачи решается последовательность задач выпуклого или линейного программирования. Итак, для решения задачи (5.1), (5.2) рассмотрим следующую задачу параметрического программирования:

$$Z(x, \lambda) = \varphi(x) - \lambda\psi(x) \rightarrow \min, \tag{5.10}$$

$$f_i(x) \leq 0 \quad (i = \overline{1, m}). \tag{5.11}$$

Если обозначить через

$$F(\lambda) = \min_{x \in S(x)} Z(x, \lambda),$$

тогда решение задачи (5.1), (5.2) сводится к решению уравнения $F(\lambda) = 0$.

Теорема 5.2. *Пусть для $\lambda = \lambda^*$ имеем $F(\lambda^*) = 0$, а x^* – оптимальное решение задачи* (5.10), (5.11) *при $\lambda = \lambda^*$, тогда $\lambda^* = \dfrac{\varphi(x^*)}{\psi(x^*)}$, а x^* является оптимальным решением задачи* (5.1), (5.2).

Для нахождения корня уравнения $F(\lambda) = 0$ используются различные общеизвестные методы. В работах [63, 172, 191, 260, 261, 362, 380, 436, 478] приводятся такие методы и делается анализ их эффективности.

В общем случае параметрический метод решения задачи (5.1), (5.2) заключается в следующем:

Шаг 0. Берем некоторое значение параметра $\lambda = \lambda_0$, чтобы $F(\lambda_0) \geq 0$.

Шаг k. Пусть λ_{k-1} – значение параметра λ, найденное на предыдущем шаге. Тогда:

- находим решение x_k^* задачи выпуклого программирования (5.10), (5.11) при $\lambda = \lambda_{k-1}$;
- полагаем $\lambda_k = \dfrac{\varphi(x_k^*)}{\psi(x_k^*)}$;
- если $|F(\lambda_k)| \leq \delta$, где δ – достаточно малое положительное число, то x_k^* является оптимальным решением задачи (5.1), (5.2).

Другие модификации данного метода получаются при различных подходах выбора значений параметра λ, т.е. методов решения уравнения $F(\lambda) = 0$.

Преимущество такого метода состоит в том, что система ограничений исходной задачи остается без изменений и поэтому при решении задачи (5.10), (5.11) можно использовать ее специфику.

В случае задачи дробно-линейного программирования на каждом шаге параметрического метода решается задача линейного программирования, а для задач транспортного типа с дробно-линейным или дробно-нелинейным функционалом параметрический метод не выводит нас из данного класса задач [664, 667, 733].

Недостаток данного метода состоит в том, что вместо одной исходной задачи необходимо решать целую последовательность задач того же класса, что сказывается на быстродействии ее решения.

Отметим преимущество параметрического метода для решения следующей задачи блочной структуры с дробным функционалом:

$$f(x) = \frac{\sum\limits_{j=1}^{p} \varphi_j(x_j)}{\sum\limits_{j=1}^{p} \psi_j(x_j)} \to \min,$$

$$h_j^l(x_j) \le b_j^l \quad (l = \overline{1, m_j};\ j = \overline{1, p}),$$

где $x_j \in R^{n_j}$, а $x \in R^n$.

В параметрическом методе на k-ом шаге, при фиксированном $\lambda = \lambda_k$, решаются p задач выпуклого программирования для каждого блока в отдельности, а именно

$$z_j(x_j, \lambda_k) = \varphi_j(x_j) - \lambda_k \psi_j(x_j) \ \to \min;$$

$$h_j^l(x_j) \le b_j^l \quad (l = \overline{1, m_j}).$$

Замечание. *В некоторых итерационных схемах блочную задачу дробного программирования необходимо повторно решать с незначительными изменениями функций $\varphi_j(x_j)$ числителя. В таких случаях, для нахождения приближенного решения блочной задачи дробного программирования, в параметрическом методе достаточно выполнить один шаг, в котором значение параметра λ вычисляется с учетом найденного решения на предыдущей итерации схемы, т.е., если на t-й* *итерации найдено решение x^t блочной задачи дробного программирования, то на $(t+1)$-й итерации берется $\lambda = \lambda^t = f(x^t)$. Таким образом, на каждой итерации вычислительной схемы решаются p задач выпуклого программирования:*

$$z_j(x_j, \lambda^t) \to \min,$$
$$h_j^l(x_j) \le b_j^l \quad (l = \overline{1, m_j}).$$

Декомпозиционные методы. Для решения задач дробно-выпуклого программирования блочной структуры в данной работе приводятся декомпозиционные методы [496, 667, 733], основанные на:

- схеме декомпозиции по ограничениям;
- схеме декомпозиции по переменным;
- схеме декомпозиции по ресурсам.

Данные методы разработаны на основе функции Лагранжа в сочетании с методами недифференцируемой оптимизации.

5.2. Задача нахождения седловой точки в дробно-выпуклом программировании

Рассмотрим задачу дробно-выпуклого программирования, которая имеет вид:

$$f_0(x) = \frac{\varphi(x)}{\psi(x)} \to \min, \tag{5.12}$$

$$f_i(x) \le 0 \quad (i = \overline{1, m}), \tag{5.13}$$

$$x \in X. \tag{5.14}$$

Решение данной задачи может быть сведено к нахождению седловой точки функции Лагранжа. Как отмечено ранее, для задачи дробно-выпуклого программирования можно построить два типа функций Лагранжа:

$$L_1(x, u) = \frac{\varphi(x)}{\psi(x)} + \sum_{i=1}^{m} u_i f_i(x),$$

$$L_2(x, u) = \frac{\varphi(x) + \sum_{i=1}^{m} u_i f_i(x)}{\psi(x)},$$

где $u = \{u_i\}$ – вектор множителей Лагранжа для ограничений (5.13). Функции $L_1(x, u)$ и $L_2(x, u)$ эквивалентны (теорема 1.21). Поэтому можно рассматривать только один тип функций Лагранжа для задачи (5.12)–(5.14). Тогда решение задачи (5.12)–(5.14) может быть сведено к решению одной из следующих двух задач нахождения седловой точки:

$$\max_{u\geq 0}\min_{x\in X} L_1(x,u),$$

$$\max_{u\geq 0}\min_{x\in X} L_2(x,u).$$

Согласно теоремы 1.22 в результате решения одной из этих задач можно найти и седловую точку другой задачи. Поэтому достаточно решить только одну задачу нахождения седловой точки для задачи (5.12)–(5.14).

Рассмотрим следующую задачу нахождения седловой точки:

$$\max_{u\geq 0}\min_{x\in X} L(x,u), \tag{5.15}$$

в которой $x \in R^n,\ u \in R^m$, а функция $L(x,u)$ определяется по формуле

$$L(x,u) = \frac{\varphi(x) + \sum\limits_{i=1}^{m} u_i f_i(x)}{\psi(x)}.$$

Предположим, что функции $\varphi(x), -\psi(x), f_i(x)\quad (i=\overline{1,m})$ являются выпуклыми и удовлетворяют условиям

$$\varphi(x) \geq 0, \quad \psi(x) > 0, \quad f_i(x) \leq 0 \quad (i=\overline{1,m})$$

для любого $x \in X$.

Тогда функция $L(x,u)$ является квазивыпуклой по переменным x на множестве X при фиксированных значениях $u \geq 0$ и линейной по переменным u при фиксированных значениях переменных $x \in X$.

Задача (5.15) является задачей нахождения седловой точки функции Лагранжа $L(x,u)$, т.е. найти такую пару (x^*,u^*) для которых выполняются двухсторонние неравенства

$$L(x^*,u) \leq L(x^*,u^*) \leq L(x,u^*)$$

для любых $u \geq 0$ и $x \in X$.

Задачу (5.15) можно разделить на две задачи:

$$\max_{u\geq 0} L(u) \tag{5.16}$$

и

$$L(u) = \min_{x\in X} L(x,u). \tag{5.17}$$

Функция $L(u)$ является кусочно-линейной и недифференцируемой, а ее значения в заданной точке $\overline{u}$ вычисляются алгоритмически, т.е. необходимо решить задачу (5.17) при фиксированных $\overline{u}$, а полученное решение $x^*(\overline{u})$ использовать для вычисления значений функции $L(u)$ в точки $\overline{u}$ по формуле

$$L(\overline{u}) = L(x^*(\overline{u}), \overline{u}) = \frac{\varphi(x^*(\overline{u})) + \sum_{i=1}^{m} \overline{u}_i f_i(x^*(\overline{u}))}{\psi(x^*(\overline{u}))}.$$

Пусть для решения задачи (5.16) применяется некоторая итерационная процедура недифференцируемой оптимизации, т.е. некоторый субградиентный метод, в результате которого вырабатывается некоторая последовательность значений $u^0, u^1, u^2, \ldots, u^t, \ldots$, сходящиеся к оптимальной точке u^*. Следует отметить, что в оптимальной точке u^* выполняется условие

$$u_i^* \frac{\partial L(u^*)}{\partial u_i} = 0 \quad (i = \overline{1,m}), \quad \text{т.е.} \quad u_i^* f_i(x^*) = 0 \quad (i = \overline{1,m}).$$

Для вычисления значений обобщенного градиента функции $L(u)$ в некоторой точке u^t необходимо решать задачу (5.17). Задача (5.17) является задачей дробно-выпуклого программирования и для ее решения можно использовать любой из методов приведенные в п. 5.1. Одним из таких методов является параметрический.

В соответствие с параметрическим методом задача дробно-выпуклого программирования (5.17) сводится к решению задачи параметрического программирования

$$\min_{x \in X} \{Z(x, u, \lambda) = \varphi(x) + \sum_{i=1}^{m} u_i f_i(x) - \lambda \psi(x)\}. \tag{5.18}$$

Задача (5.18) при фиксированных значениях параметра λ является задачей выпуклого программирования. Поэтому решение задачи (5.17) сводится к решению последовательности задач (5.18) при фиксированных значениях параметра λ.

Учитывая тот факт, что субградиентные методы являются устойчивыми относительно точности вычисления значений обобщенного градиента, в качестве оптимального решения задачи (5.17) может быть взято некоторое приближенное решение $\overline{x}(u) \in X$. Такое решение может быть получено, если решить только одну задачу выпуклого программирования (5.18) при некотором фиксированном значении параметра λ.

Также следует отметить, что если для решения общей задаче выпуклого программирования (5.18) используется некоторый итерационный метод, то в качестве первоначальной точке может быть взято предыдущее решение $\overline{x}(u) \in X$.

Открытым остается только вопрос, каким образом фиксировать значения параметра λ, чтобы в процессе решения задачи (5.15) получить седловую точку функции $L(x, u)$.

Для этих целях предлагается использовать следующую процедуру выбора значений параметра λ.

Итак, пусть для решения задачи (5.16) используется некоторый субградиентный метод с первоначальными заданными значениями u^0 и λ^0, а также соответствующие им решение задачи (5.18), т.е. имеем первоначальное решение $\overline{x}(u^0, \lambda^0)$.

Перед выполнением очередного $(t+1)$-го $(t = 0, 1, 2, \dots)$ шага субградиентного метода имеем:

u^t – значение переменных u полученных на предыдущем шаге субградиентного метода;

λ^t – значение параметра λ вычисленное на предыдущем шаге субградиентного метода;

$\overline{x}(u^t, \lambda^t)$ – решение задачи (5.17) полученное при фиксированном $u = u^t$ и одновременно, оптимальное решение задачи выпуклого программирования (5.18), полученное при фиксированном $\lambda = \lambda^t$.

Тогда на $(t+1)$-м шаге субградиентного метода в сочетанием с параметрическим методом выполняем четыре его этапа.

1. Находим значения обобщенного градиента функции $L(u)$ в точке $u = u^t$, вычисленное с учетом оптимального решения $\overline{x}(u^t, \lambda^t)$ задачи (5.18),
$$g_i(u^t) = f_i\big(\overline{x}(u^t, \lambda^t)\big)/\psi\big(\overline{x}(u^t, \lambda^t)\big) \quad (i = \overline{1, m}).$$
2. Находим новое значение параметра λ по формуле
$$\lambda = \lambda^{t+1} = L(\overline{x}(u^t, \lambda^t), u^t).$$
3. Находим новые значения переменных u^{t+1} по формуле, соответствующей выбранному субградиентному методу, например,
$$u_i^{t+1} = \max\{0, u_i^t + h_{t+1} g_i(u^t)\} \quad (i = \overline{1, m}),$$
где h_{t+1} величина шага в субградиентном методе.
4. Решаем задачу выпуклого программирования (5.18) при фиксированном $\lambda = \lambda^{t+1}$ и находим оптимальное решение $\overline{x}(u^{t+1}, \lambda^{t+1})$.

В качестве начальной точки в методе выпуклого программирования для решения задачи (5.18) берется точка $\overline{x}(u^t, \lambda^t)$.

Таким образом, для вычисления значений обобщенного градиента функции $L(u)$ в точки $u = u^t$ решаем задачу дробно-выпуклого программирования (5.17) или одну задачу выпуклого программирования (5.18) при фиксированном значения параметра λ по формуле

$$\lambda = \lambda^{t+1} = L(\overline{x}(u^t, \lambda^t), u^t).$$

Теорема 5.3. *Если u^* является оптимальным решением задачи* (5.16)*, то решение $x^*(u^*, \lambda^*)$ является оптимальным для задачи* (5.17)*, а пара (u^*, x^*) является седловой точкой функции $L(x, u)$, т.е. решение задачи* (5.15)*.*

5.3. Субградиентный метод решения задачи дробно-выпуклого программирования

Рассмотрим задачу дробно-выпуклого программирования (5.1), (5.2) вида

$$f_0(x) = \frac{\varphi(x)}{\psi(x)} \to \min,$$

$$f_i(x) \le 0 \quad (i = \overline{1, m}),$$

для решения которой непосредственно применяются методы недифференцируемой оптимизации в сочетании с параметрическим методом.

Обозначим через $S(x)$ множество значений переменных x, удовлетворяющих ограничениям (5.2) и предположим, что $\varphi(x) \ge 0$, а $\psi(x) > 0$ для $\forall x \in S(x)$.

Задачу (5.1), (5.2) представим в эквивалентном параметрическом виде

$$z(x, \lambda) = \varphi(x) - \lambda\psi(x) \to \min,$$

$$f_i(x) \le 0 \quad (i = \overline{1, m}).$$

Тогда при фиксированном значении параметра $\lambda = \overline{\lambda} \ge 0$, данная задача является задачей выпуклого программирования. Применим для решения задачи (5.1), (5.2) субградиентные методы, типа метода эллипсоидов, на каждой итерации которого будем гарантировать неотрицательное значение параметра λ. Пусть выполнены k итераций некоторого субградиентного метода, в результате которого для задачи (5.1), (5.2) найдено некоторое решение x^k. Тогда на $(k+1)$-й итерации выполняем четыре этапа.

1. Находим значение параметра λ_k по формуле

$$\lambda_k = \begin{cases} f_0(x^k), \text{ если } \varphi(x^k) \geq 0 \text{ и } \psi(x^k) > 0, \\ 0, \text{ в противном случае.} \end{cases}$$

2. Находим $f_{i^*}(x^k) = \max\limits_{1 \leq i \leq m} f_i(x^k)$, где i^* значение индекса i функции $f_i(x^k)$ с максимальным значением в точке x^k.

3. Вычисляем значения обобщенного градиента в точке x^k по формуле

$$g(x^k) = \begin{cases} g_z(x^k) = g_\varphi(x^k) - \lambda_k g_\psi(x^k), \text{ если } f_{i^*}(x^k) \leq 0, \\ g_{i^*}(x^k) = g_{f_{i^*}}(x^k), \text{ если } f_{i^*}(x^k) > 0, \end{cases}$$

где $g_\varphi(x^k)$ и $g_\psi(x^k)$ соответственно субградиенты функций $\varphi(x)$ и $\psi(x)$ в точке x^k, а $g_{f_{i^*}}(x^k)$ – субградиент функции $f_{i^*}(x)$ в точке x^k.

4. Находим новую точку по формуле

$$x^{k+1} = x^k - h_k g(x^k),$$

где h_k – величина шага в субградиентном методе.

Остальные вычисления соответствуют выбранному субградиентному методу.

При решения задачи (5.1), (5.2) субградиентным методом учитываются две особенности:

1) если точка $x^k \in S(x)$, т.е. она является допустимой для задачи (5.1), (5.2) и $\lambda_k = f_0(x^k) \geq 0$, то значение субградиента дробно-выпуклой функции $f_0(x)$ определяется относительно выпуклой функции $z(x, \lambda_k)$ и имеет вид

$$g_z(x^k, \lambda_k) = g_\varphi(x^k) - \lambda_k g_\psi(x^k);$$

2) если точка $x^k \notin S(x)$, т.е. она является недопустимой для задачи (5.1), (5.2), то значение субградиента определяется относительно выпуклой функции $f_{i^*}(x)$, для которой $f_{i^*}(x_k) > 0$ и она является наиболее удаленной от точки x^k.

Следует также отметить, что формула

$$\lambda_k = \begin{cases} f_0(x^k), \text{ если } \psi(x^k) > 0, \\ 0, \text{ в противном случае} \end{cases}$$

имеет чисто описательный смысл, так как, если $x^k \in S(x)$, то $\psi(x^k) > 0$ и $\lambda_k = f_0(x^k)$, значение которого используется при вычислении субградиента для функции $z(x, \lambda_k)$. Если же $x^k \notin S(x)$, то $\lambda_k = f_0(x^k)$ для $\psi(x^k) > 0$, и $\lambda_k = 0$ в противном случае. Однако, независимо от варианта вычисления значения параметра λ_k, при $x^k \notin S(x)$ он не используется при расчетах значений обобщенного градиента.

Теорема 5.4 *Пусть x^* является точка, к которой сходится выработанная последовательность $x^0, x^1, x^2, \ldots$ субградиентным методом. Тогда точка x^* является оптимальным решением задачи дробно-выпуклого программирования* (5.1), (5.2).

По аналогии с задачей дробно-выпуклого программирования (5.1), (5.2) рассмотрим задачу минимизации суммы выпуклых и дробно-выпуклых функций, т.е. задачу следующего вида:

$$f(x) = \frac{\varphi(x)}{\psi(x)} + h(x) \to \min, \tag{5.19}$$

$$f_i(x) \leq 0 \quad (i = \overline{1, m}). \tag{5.20}$$

Пусть на множестве допустимых решений $S(x) = \{x : f_i(x) \leq 0, i = \overline{1, m}\}$ функции $\varphi(x)$ и $h(x)$ – выпуклы, а $\psi(x)$ – вогнута, притом $\varphi(x) \geq 0$, $\psi(x) > 0$, а $h(x) \leq 0$, для любого $x \in S(x)$. Тогда функция $f(x)$ является квазивогнутой и имеет единственный локальный минимум, и тем самым задача (5.19), (5.20) является задачей квазивогнутой оптимизации.

Для решения задачи (5.19), (5.20) можно применить параметрический метод, а также один из субградиентных методов недифференцируемой оптимизации.

Для этого рассмотрим параметрический аналог задачи (5.19), (5.20) в следующем виде

$$F(\lambda) = \min_{x \in S(x)} \{Z(x, \lambda) = \varphi(x) - \lambda\psi(x) + h(x)\}. \tag{5.21}$$

Если предположить, что $\lambda \geq 0$, то функция $Z(x, \lambda)$ является выпуклой по x, так как представляет собой сумму трех выпуклых функций $\varphi(x)$, $-\psi(x)$ и $h(x)$. Тогда задача (5.21) при фиксированном $\lambda \geq 0$ является задачей выпуклого программирования и решение исходной задачи (5.19), (5.20) сводится к нахождению корня уравнения $F(\lambda) = 0$.

Тогда параметрический метод решения задачи (5.19), (5.20) заключается в следующем:

Шаг 0. Выбираем некоторое значение $\lambda = \lambda_0$, чтобы $F(\lambda_0) \geq 0$.

Шаг k. Пусть λ_{k-1} – значение параметра λ, найденное на предыдущем шаге. Тогда:

- находим решение x_k^* задачи выпуклого программирования (5.21) при $\lambda = \lambda_{k-1}$;

- полагаем $\lambda_k = \varphi(x_k^*)/\psi(x_k^*)$;

- если $|F(\lambda_k)| \leq \delta$, где δ достаточно малое положительное число, то x_k^* является оптимальным решением задачи (5.19), (5.20).

Применим теперь один из субградиентных методов (например, метод эллипсоидов) в сочетание с параметрическим методом для решения задачи (5.19), (5.20), т.е. при фиксированном значении параметра $\lambda = \overline{\lambda} \geq 0$, будем рассматривать задачи минимизации выпуклой функции $Z(x, \overline{\lambda})$ при ограничениях (5.20).

Пусть выполнены k итерации некоторого субградиентного метода решения исходной задаче (5.19), (5.20) и получена точка x^k. Тогда на $(k+1)$-й итерации выполняем следующие этапы.

1. Находим значение параметра λ_k по формуле

$$\lambda_k = \begin{cases} \varphi(x^k)/\psi(x^k), \text{ если } \varphi(x^k) \geq 0 \text{ и } \psi(x^k) > 0; \\ 0, \text{ в противном случае.} \end{cases}$$

2. Находим $f_{i^*}(x^k) = \max\limits_{1 \leq i \leq m} f_i(x^k)$, где i^* значение индекса i функции $f_i(x^k)$ с максимальным значением в точке x^k.

3. Вычисляем значение обобщенного градиента в точке x^k по формуле

$$g(x^k) = \begin{cases} g_Z(x^k) = g_\varphi(x^k) - \lambda_k g_\psi(x^k) + g_h(x^k), \text{ если } f_{i^*}(x^k) \leq 0; \\ g_{i^*}(x^k) = g_{f_{i^*}}(x^k), \text{ если } f_{i^*}(x^k) > 0, \end{cases}$$

где $g_\varphi(x^k)$, $g_\psi(x^k)$ и $g_h(x^k)$ – соответственно субградиенты функций $\varphi(x)$, $\psi(x)$ и $h(x)$ в точке x^k.

4. Находим новую точку x^{k+1} по формуле

$$x^{k+1} = x^k - \gamma_k(x^k),$$

где γ_k – величина шага в субградиентном методе.

Остальные вычисления соответствуют выбранному субградиентному методу.

5.4. Метод негладких штрафных функций для решения задач дробно-выпуклого программирования

Рассмотрим задачу дробно-выпуклого программирования (5.12)–(5.14), для решения которой применяется метод негладких штрафных функций. Пусть $R(x)$ – множество допустимых решений задачи (5.12)–(5.14).

Построим штрафные функции для задачи (5.12)–(5.14) по одной из формул

$$P_t(x) = \frac{\varphi(x) + S_t(x)}{\psi(x)} \quad (t = \overline{1,4})$$

или

$$Q_t(x) = \frac{\varphi(x)}{\psi(x)} + S_t(x) \quad (t = \overline{1,4})$$

и рассмотрим задачи

$$\min_{x \in X} P_t(x) \quad (t = \overline{1,4}) \tag{5.22}$$

или

$$\min_{x \in X} Q_t(x) \quad (t = \overline{1,4}), \tag{5.23}$$

в которых $S_t(x)$ определяются по одной из формул:

$$S_1(x) = \sum_{i=1}^{m} s_i \ \max\{0, f_i(x)\}, \tag{5.24}$$

$$S_2(x) = p \max\{0, \max_{1 \le i \le m} f_i(x)\}, \tag{5.25}$$

$$S_3(x) = \sum_{i=1}^{m} \alpha_i(f_i(x)), \tag{5.26}$$

$$S_4(x) = \beta(\max_{1 \le i \le m} f_i(x)). \tag{5.27}$$

Параметры p, $s_i \quad (i = \overline{1,m})$ и функции β и $\alpha_i \quad (i = \overline{1,m})$ определяются в соответствии с требованиями, приведенные в параграфах 1.10 и 1.11 данной работы.

Так как функции $P_t(x)$ и $Q_t(x)$, являются недифференцируемыми, то для решения любой из задач безусловной оптимизации типа (5.22) и (5.23) используется субградиентный метод в сочетанием с параметрическим методом для параметрических штрафных функциях

$$z_t(x, \lambda) = \varphi(x) - \lambda\psi(x) + S_t(x) \quad (t = \overline{1,4}),$$

которые являются выпуклыми и одинаковы для функциях $P_t(x)$ и $Q_t(x)$.

Пусть на k-м шаге субградиентного метода минимизации функций $P_t(x)$ или $Q_t(x)$ найдена точка x^k. Тогда на очередном $(k+1)$-м шаге субградиентного метода необходимо выполнить основные четыре этапа.

1. Находим значение параметра λ_k по формуле

$$\lambda_k = \begin{cases} f_0(x^k), \text{ если } \psi(x) > 0; \\ 0, \text{ если } \psi(x) \leq 0. \end{cases}$$

2. Вычислим значения штрафной функции $S_t(x)$ в точке x^k по одной из формул (5.24)–(5.27).

3. Вычислим значения обобщенного градиента функции $P_t(x)$ или функции $Q_t(x)$ в точке x^k с учетом функции $z_t(x, \lambda)$ по одной и той же формуле

$$g_z(x^k) = \begin{cases} g_\varphi(x^k) - \lambda_k g_\psi(x^k), \text{ если } x^k \in R(x); \\ g_\varphi(x^k) - \lambda_k g_\psi(x^k) + g_{S_t}(x^k), \ x^k \notin R(x), \end{cases}$$

где g_φ и g_ψ соответственно субградиенту функций $\varphi(x)$ и $\psi(x)$ вычисленных в точке x^k, а g_{S_t} – соответственно субградиент одной из функций $S_t(x) \ \ (t = \overline{1,4})$, вычисленных в точке x^k.

Отметим некоторые особенности вычисления субградиентов функций $S_t(x) \ \ (t = \overline{1,4})$ в точке x^k. Пусть $x^k \notin R(x)$, а I^+ – дискретное множество индексов i, для которых $f_i(x^k) > 0$. Тогда для каждой функции в отдельности имеем:

1) $g_{S_1}(x^k) = \sum\limits_{i \in I^+} s_i \cdot g_{f_i}(x^k)$,

где $g_{f_i}(x^k)$ значения субградиентов функций $f_i(x), \ i \in I^+$, вычисленных в точке x^k.

2) $g_{S_2}(x^k) = p \cdot g_{f_{i^*}}(x^k)$,

где $g_{f_{i^*}}(x^k)$ значение субградиента функции $f_{i^*}(x)$, вычисленной в точке x^k, а i^* значение индекса i, для которого

$$f_{i^*}(x^k) = \max_{i \in I^+} f_i(x^k).$$

3) $g_{S_3}(x^k) = \sum\limits_{i \in I^+} g_{\alpha_i} \cdot g_{f_i}(x^k)$,

где g_{α_i} – субградиенты функций α_i, а $g_{f_i}(x^k)$ – субградиенты функций $f_i(x)$, $i \in I^+$, вычисленных в точке x^k.

4) $g_{S_4}(x^k) = g_\beta \cdot g_{f_{i^*}}(x^k)$,

где g_β – субградиент функции β, $g_{f_{i^*}}(x^k)$ – субградиент функции $f_{i^*}(x)$, а i^* –значение индекса i, для которого $f_{i^*}(x^k) = \max\limits_{i \in I^+} f_i(x^k)$.

4. Вычислим значения новой точки x^{k+1} с учетом ее проектирования на множестве X. Для этого:

- находим

$$x^{k+1} = x^k - h_{k+1}\, g_z(x^k),$$

где h_{k+1} – величина шага в субградиентном методе;

- выполняем операцию проектирования точки x^{k+1} на множестве X, т.е. находим новые значения точки ΠR_X

$$x^{k+1} = \Pi R_X[x^{k+1}],$$

где ΠR_X – оператор проектирования, который зависит от структуры множества X.

5.5. Схемы декомпозиции по ограничениям в дробно-выпуклом программировании

5.5.1. Задача дробно-выпуклого программирования

Рассмотрим следующую задачу дробно-выпуклого программирования, в которой система ограничений разбита на две группы:

$$f_0(x) = \frac{\varphi(x)}{\psi(x)} \to \min, \tag{5.28}$$

$$f_i(x) \le a_i \quad (i = \overline{1,m}), \tag{5.29}$$

$$h_k(x) \le b_k \quad (k = \overline{1,p}). \tag{5.30}$$

Пусть $x = (x_1, x_2, \ldots, x_n) \in R^n$, φ, ψ, $f_i \quad (i = \overline{1,m})$ и $h_k \quad (k = \overline{1,p})$: $R^n \to R$ – непрерывные и дифференцируемые функции в R^n, а $a_i \quad (i = \overline{1,m})$ и $b_k \quad (k = \overline{1,p})$ – заданные числа. Если предположить, что функции φ, $-\psi$, $f_i \quad (i = \overline{1,m})$ и $h_k \quad (k = \overline{1,p})$ выпуклы, то задача (5.28)–(5.30) будет задачей дробно-выпуклого программирования.

Обозначим через $S(x)$ множество значений переменных x, удовлетворяющих ограничениям (5.30). Предположим, что множество ограничено и для любого $x \in S(x)$ имеем $\psi(x) > 0$. Построим функцию Лагранжа задачи (5.28)–(5.30) на множестве ограничений (5.29)

$$L(x,u) = \Big[\varphi(x) + \sum_{i=1}^{m} u_i \cdot (f_i(x) - a_i)\Big] / \psi(x),$$

где $u = (u_1, u_2, \ldots, u_m)$ – множители Лагранжа ограничений (5.29), и рассмотрим следующие две взаимосвязанные задачи:

$$\max_{u \geq 0} L^*(u); \tag{5.31}$$

$$L^*(u) = \min_{x \in S(x)} L(x,u). \tag{5.32}$$

Тогда решение задачи (5.28)–(5.30) сведется к задаче максимизации функционала $L^*(u)$ при ограничениях $u \geq 0$. Так как $S(x)$ – ограниченное множество, то функция $L^*(u)$ определена для любого u, является вогнутой и кусочно-линейной. Для решения задачи (5.31) используем субградиентные методы недифференцируемой оптимизации. Из выражения для $L^*(u)$ задачи (5.32)

$$L^*(u) = \Big[\varphi(x^*(u)) + \sum_{i=1}^{m} u_i \cdot \big(f_i(x^*(u)) - a_i\big)\Big] / \psi(x^*(u)),$$

где $x^*(u)$ – оптимальный вектор задачи (5.32), вытекает, что обобщенный градиент функции $L^*(u)$ в точке u вычисляется по формуле

$$g_i(u) = [f_i(x^*(u)) - a_i] / \psi(x^*(u)) \quad (i = \overline{1,m}). \tag{5.33}$$

Тогда итерационный процесс решения задачи (5.28)–(5.30) может быть построен с применением одного из алгоритмов метода обобщенного градиента, на $(t+1)$-м шаге которого необходимо выполнить три его этапа.

1. Решить задачу (5.32) при фиксированных $u = u^t$ (при решении задачи (5.32) можно использовать один из методов, приведенных в предыдущем параграфе).
2. Определить значения обобщенного градиента функции $L^*(u)$ в точке u^t по формуле (5.33).

3. Найти новые значения

$$u_i^{t+1} = \max\{0, u_i^t + v_{t+1}\, g_i(u^t)\} \quad (i = \overline{1,m}),$$

где v_{t+1} – величина шага.

Замечание. *На каждом шаге обобщенного градиентного метода необходимо решить задачу* (5.32) *при фиксированных значениях переменных* $u = u^t$. *Задача* (5.32) *является задачей дробно-выпуклого программирования и для ее решения применяем параметрический метод. Для этого строим задачу*

$$\min_{x \in S(x)} \left\{ Z(x, u^t, \lambda) = \varphi(x) - \lambda\psi(x) + \sum_{i=1}^{m} u_i^t \cdot (f_i(x) - a_i) \right\}, \tag{5.34}$$

которая при фиксированном значении параметра λ *является задачей выпуклого программирования.*

В соответствии с параметрическим методом решается некоторая последовательность таких задач, каждый раз фиксируя новые значения параметра λ до тех пор пока не будет найдено оптимальное решение задачи (5.32). Однако, учитывая тот факт, что метод обобщенного градиента является устойчивым, относительно погрешности вычисления значений обобщенного градиента, то задача (5.32) может быть решена и приближенно. Для этого применим следующую процедуру ее решения.

Пусть для решения задачи (5.31) применяется обобщенный градиентный метод с первоначальными данными u^0, λ^0 и полученным решением задачи (5.34) $x^*(u^0, \lambda^0)$.

Допустим, что перед $(t+1)$-м шагом $(t = 0, 1, 2, \dots)$ имеем следующие значения: u^t, λ^t и $x^*(u^t, \lambda^t)$. Тогда с учетом параметрического метода выполняем четыре этапа.

1. Находим значение обобщенного градиента функции $L^*(u)$ в точке u^t с учетом решения $x^*(u^t, \lambda^t)$ по формуле

$$g_i(u^t) = [f_i\left(x^*(u^t, \lambda^t)\right) - a_i]/\psi\left(x^*(u^t, \lambda^t)\right).$$

2. Определяем значение параметра λ по формуле

$$\lambda^{t+1} = L(x^*(u^t), u^t).$$

3. Находим новую точку

$$u_i^{t+1} = \max\left\{0, u_i^t + v_{t+1}\, g_i(u_i^t)\right\} \quad (i = \overline{1,m}).$$

4. Решаем задачу выпуклого программирования (5.34) при фиксированных параметрах $\lambda = \lambda^{t+1}$ и $u = u^{t+1}$ и находим решение $x^*(u^{t+1}, \lambda^{t+1})$, которое используется для вычисления обобщенного градиента на следующем шаге. В качестве исходной точке в методе решения задачи (5.34) берется предыдущее решение $x^*(u^t, \lambda^t)$.

Таким образом, на каждом шаге метода обобщенного градиента решается только одна задача выпуклого программирования, или точнее говоря, продолжается решение предыдущей задачи выпуклого программирования с точке $x^*(u^t, \lambda^t)$ но с новыми значения параметров λ и u.

Данный метод решения задачи дробно-выпуклого программирования может быть успешно использован в случаях, когда система ограничений (5.29), (5.30) имеет специальную структуру.

5.5.2. Блочная задача дробно-выпуклого программирования

Пусть имеем p групп переменных $x_j = (x_j^1, x_j^2, \ldots, x_j^{n_j}) \in R^{n_j}$, образующих вектор $x = (x_1, x_2, \ldots, x_p) \in R^n$, где $n = \sum\limits_{j=1}^{p} n_j$. На множестве каждой группы переменных x_j определим следующие непрерывные функции: φ_j, ψ_j, f_j^i $(i = \overline{1,m})$, h_j^l $(l = \overline{1,m_j}) : R^{n_j} \to R$.

Рассмотрим следующую задачу дробно-выпуклого программирования, в которой система ограничений имеет блочно-диагональную структуру:

$$F(x) = \frac{\varphi(x)}{\psi(x)} = \frac{\sum\limits_{j=1}^{p} \varphi_j(x_j)}{\sum\limits_{j=1}^{p} \psi_j(x_j)} \to \min, \tag{5.35}$$

$$\sum_{j=1}^{p} f_j^i(x_j) \le a_i \quad (i = \overline{1,m}), \tag{5.36}$$

$$h_j^l(x_j) \le b_j^l \quad (l = \overline{1,m_j};\ j = \overline{1,p}), \tag{5.37}$$

в которой $a = (a_1, a_2, \ldots, a_m) \in R^m$; $b_j = (b_j^1, b_j^2, \ldots, b_j^{m_j}) \in R^{m_j}$, $j = \overline{1,p}$ – постоянные числа, а функции φ_j, $-\psi_j$, f_j^i $(i = \overline{1,m})$, h_j^l $(l = \overline{1,m_j})$ – выпуклы.

В отличие от задачи (5.28)–(5.30) вторая группа ограничений данной задачи имеет блочно-диагональную структуру. Для ее решения используем приведенную выше схему разложения по ограничениям.

Для задачи (5.35), (5.36) построим функцию Лагранжа

$$L(x,u) = \Big[\varphi(x) + \sum_{i=1}^{m} u_i \cdot \Big(\sum_{j=1}^{p} f_j^i(x_j) - a_i\Big)\Big] / \psi(x), \qquad (5.38)$$

и рассмотрим задачи

$$\max_{u \geq 0} L^*(u), \qquad (5.39)$$

$$L^*(u) = \min_{x \in S(x)} L(x,u), \qquad (5.40)$$

где $u = (u_1, u_2, \ldots, u_m)$ – множители Лагранжа для ограничений (5.36), а $S(x)$ – выпуклое множество значений переменных x, удовлетворяющих ограничениям (5.37).

Тогда решение задачи (5.35)–(5.37) сводится к задаче максимизации функционала $L^*(u)$ при ограничениях $u \geq 0$. Для решения задачи (5.39) используем субградиентные методы недифференцируемой оптимизации.

Из выражения (5.38) вытекает, что обобщенный градиент функции $L^*(u)$ в точке u вычисляется по формуле

$$g_i(u) = \Big[\sum_{j=1}^{p} f_j^i(x_j^*(u)) - a_i\Big] \Big/ \sum_{j=1}^{p} \psi_j(x_j^*(u)) \quad (i = \overline{1,m}), \qquad (5.41)$$

где $x^*(u) = \{x_j^*(u)\}$ – оптимальный вектор задачи (5.40) при фиксированном значении параметра u.

Тогда итерационный процесс решения задачи (5.39) может быть построен с применением одного из алгоритмов метода обобщенного градиента, на $(t+1)$-м шаге которого необходимо выполнить три его этапа.

1. Решить задачу дробно-выпуклого программирования (5.40) при фиксированных $u = u^t$.
2. Определить значения обобщенного градиента функции $L^*(u)$ в точке u^t по формуле (5.41).

3. Вычислить новые значения
$$u_i^{t+1} = \max\{0,\ u_i^t + v_{t+1}\, g_i(u_i^t)\} \quad (i = \overline{1,m}),$$
где v_{t+1} – величина шага.

На каждом шаге субградиентного метода необходимо решить задачу (5.40) при фиксированных значениях переменных $u_i = u_i^t$ $(i = \overline{1,m})$, которая имеет блочные ограничения и несепарабельный дробно-выпуклый функционал. Для ее решения используется параметрический метод. Тогда (5.40) сводится к решению p задач выпуклого программирования следующего вида:
$$\varphi_j(x_j) - \lambda\psi_j(x_j) + \sum_{i=1}^{m} u_i^t\, f_j^i(x_j) \to \min,$$
$$h_j^l(x_j) \le b_j^l \quad (l = \overline{1,m_j})$$
и при фиксированном значении параметра λ.

Замечание. *Как и при решении задачи* (5.32) *значения параметра* λ *можно фиксировать с учетом полученного решения* $x_j^*(u^t,\lambda^t)$, *т.е. задача* (5.40) *решается при фиксированными* $u = u^{t+1}$ *и* $\lambda = \lambda^{t+1} =$
$$= L\big(x^*(u^t,\lambda^t), u^t\big) = \frac{\sum\limits_{j=1}^{p}\varphi_j(x_j^*(u^t,\lambda^t)) + \sum\limits_{i=1}^{m} u_i^t\Big(\sum\limits_{j=1}^{p} f_j^i\big(x_j^*(u^t,\lambda^t)\big) - a_i\Big)}{\sum\limits_{j=1}^{p}\psi_j(x_j^*(u^t,\lambda^t))}.$$

Тогда, в конечном итоге в параметрическом методе делается только одна итерация и решаются p задач выпуклого программирования для каждого блока в отдельности, каждую из них продолжая решать с достигнутой точки $x_j^*(u^t,\lambda^t)$.

В предыдущей главе схема декомпозиции по ограничениям была использована для решения задачи дробно-линейного программирования, в результате которой исходная задача дробно-линейного программирования блочно-диагональной структуры со связующими ограничениями сводится к безусловной максимизации недифференцируемого функционала $L^*(u)$ и линейных задач для каждого блока. Аналогичные схемы были использованы для решения задач дробно-линейного программирования транспортного типа, в которых $m < n$, и в результате построены достаточно эффективные алгоритмы (см. главу 8 данной работы).

5.6. Схемы декомпозиции по переменным в дробно-выпуклом программировании

5.6.1. Задача дробно-выпуклого программирования

Рассмотрим задачу дробно-выпуклого программирования, в которой множество переменных условно разбито на две группы:

$$F(x,y) = \frac{\varphi(x,y)}{\psi(x,y)} \quad \to \min, \tag{5.42}$$

$$f_i(x,y) \le 0 \quad (i = \overline{1,m}), \tag{5.43}$$

где $x = (x_1, x_2, \ldots, x_n) \in R^n,\ y = (y_1, y_2, \ldots, y_k) \in R^k$.

Обозначим через $M(x,y)$ множество значений переменных (x,y), удовлетворяющих ограничениям (5.43), и предположим, что $\varphi(x,y) \ge 0$ и $\psi(x,y) > 0$ для всех $(x,y) \in M(x,y)$, а функции $\varphi(x,y)$, $-\psi(x,y)$, $f_i(x,y) \quad (i = \overline{1,m})$ - выпуклые и необязательно дифференцируемые по совокупности переменных x и y. Тогда функция $F(x,y)$ является квазивыпуклой функцией на выпуклом множестве $M(x,y)$.

Построим функцию Лагранжа (5.42), (5.43) по формуле

$$L(x,y,u) = \left[\varphi(x,y) + \sum_{i=1}^{m} u_i f_i(x,y)\right] \Big/ \psi(x,y), \tag{5.44}$$

где $u = (u_1, u_2, \ldots, u_m)$ – множители Лагранжа ограничений (5.43).

Предположим, что для задачи (5.42), (5.43) выполняются необходимые и достаточные условия Куна-Таккера и существует пара $\{(x^*, y^*), u^*\}$, которая является седловой точкой функции $L(x,y,u)$.

Рассмотрим общую схему декомпозиции по переменным для задачи (5.42), (5.43). Зафиксируем некоторую группу переменных, например $x = \overline{x}$, и рассмотрим задачу:

$$F(\overline{x},y) = \frac{\varphi(\overline{x},y)}{\psi(\overline{x},y)} \ \to \min; \tag{5.45}$$

$$f_i(\overline{x},y) \le 0 \quad (i = \overline{1,m}). \tag{5.46}$$

Для тех значений $\overline{x}$, для которых разрешима задача (5.45), (5.46), определим функцию $\Phi(\overline{x})$ по формуле

$$\Phi(\overline{x}) = \min_{y \in S(\overline{x})} F(\overline{x},y), \tag{5.47}$$

где $S(\overline{x})$ – множество значений переменных y, удовлетворяющих ограничениям (5.46) при фиксированных значениях переменных $x = \overline{x}$.

Теорема 5.5. *Если функции $\varphi(x,y)$, $-\psi(x,y)$ и $f_i(x,y)$ $(i=\overline{1,m})$ – выпуклые по совокупности переменных x и y, то функция $\Phi(x)$, определенная по формуле* (5.47)*, является квазивыпуклой на некотором подмножестве $W \subset R_x^n$. Если для некоторого $\overline{x} \in W$ выполнено условие Слейтера, то обобщенный градиент функции $\Phi(x)$ в точке $x = \overline{x}$ вычисляется по формуле*

$$g_\Phi(\overline{x}) = g_L^x(\overline{x}, y(\overline{x})), \tag{5.48}$$

где L – функция Лагранжа задачи (5.42), (5.43)*, которая определяется по формуле* (5.44)*; $y(\overline{x})$ – одно из оптимальных значений в задаче* (5.45), (5.46)*; $u = \{u_i\}$ – множители Лагранжа задачи* (5.45), (5.46)*, полученные в соответствии с теоремой Куна-Таккера; $g_L^x(\overline{x}, y(\overline{x}))$ – проекция такого обобщенного градиента функции $L(z,u) = L(x,y,u)$ на подпространство R_x^n, у которого проекция на подпространство R_y^k равна нулю* (*обобщенный градиент берется в точке $z = (\overline{x}, y(\overline{x}))$*)*.*

В случае, когда функции $\varphi(x,y)$, $\psi(x,y)$ и $f_i(x,y)$ непрерывно дифференцируемы по x, обобщенный градиент функции $\Phi(x)$ в точке $x = \overline{x}$ определяется по формуле

$$\begin{gathered} g_\Phi(\overline{x}) = \Big[\sum_{i=1}^{m} u_i(\overline{x})\, g_{f_i}^x(\overline{x}, y(\overline{x})) + g_\varphi^x(\overline{x}, y(\overline{x})) - \\ - g_\psi^x(\overline{x}, y(\overline{x}))\, L(\overline{x}, y(\overline{x}), u(\overline{x}))\Big] / \psi(\overline{x}, y(\overline{x})), \end{gathered} \tag{5.49}$$

где $g_{f_i}^x(\overline{x}, y(\overline{x}))$, $g_\varphi^x(\overline{x}, y(\overline{x}))$ и $g_\psi^x(\overline{x}, y(\overline{x}))$ – соответственно проекции обобщенных градиентов функций $f_i(x,y)$, $\varphi(x,y)$ и $\psi(x,y)$ в точке $\overline{z} = (\overline{x}, y(\overline{x}))$ на R_x^n.

Следует отметить, что не при всех $\overline{x}$ задача (5.45), (5.46) будет разрешима. Выход из таких ситуаций заключается в использовании метода "штрафных функций", при котором задача (5.45), (5.46) заменяется следующей задачей:

$$\frac{\varphi(\overline{x}, y) + M\sum_{i=1}^{m} v_i}{\psi(\overline{x}, y)} \to \min,$$

$$f_i(\overline{x}, y) - v_i \le 0 \quad (i = \overline{1,m}),$$

$$v_i \ge 0 \quad (i = \overline{1,m}),$$

где M – достаточно большое положительное число.

Легко видеть, что эта задача имеет допустимое решение при любом $\overline{x}$ с выполнением условия Слейтера.

Таким образом, если предположить, что задача (5.42), (5.43) сведена к такому виду, что функция $\Phi(x)$ определена для любого $x \in W$ и имеется возможность вычисления обобщенного градиента $g_\Phi(\overline{x})$ в произвольной точке $\overline{x}$, то получим итеративный алгоритм решения задачи (5.42), (5.43), на $(t+1)$-м шаге которого необходимо выполнить следующие три этапа.

1. Решить задачу (5.45), (5.46) при $\overline{x} = \overline{x}^t$ и определить вектор множителей Лагранжа $u(\overline{x}^t) = \{u_i(\overline{x}^t)\}$.

2. Вычислить обобщенный градиент $g_\Phi(\overline{x})$ по формуле (5.48) или (5.49).

3. Определить новые значения

$$\overline{x}^{t+1} = \overline{x}^t - h_{t+1} g_\Phi(\overline{x}^t),$$

где h_{t+1} – величина шага.

Замечание. *Для нахождения значений обобщенного градиента необходимо найти решение задачи* (5.45), (5.46) *при фиксированном* $\overline{x} = \overline{x}^t$. *Применяя параметрический метод для ее решения фиксируем значения параметра* λ *по формуле*

$$\lambda = \lambda^{t+1} = F\big(\overline{x}^t, y(\overline{x}^t, \lambda^t)\big),$$

и решаем одну задачу выпуклого программирования

$$\min_{y \in S(\overline{x}^{t+1})} \{\varphi(\overline{x}^{t+1}, y) - \lambda^{t+1} \psi(\overline{x}^{t+1}, y)\},$$

начиная с точке $y(x^t, \lambda^t)$ *предыдущей итерации субградиентного метода. Полученное решение* $y(\overline{x}^{t+1}, \lambda^{t+1})$ *и соответствующие значения множителей Лагранжа* $u(\overline{x}^{t+1}, \lambda^{t+1})$ *используются для вычисления значений обобщенного градиента функции* $\Phi(x)$.

5.6.2. Блочная задача дробно-выпуклого программирования

Рассмотрим следующую задачу блочно-диагональной структуры со связующими переменными:

$$\frac{\varphi(x,y)}{\psi(x,y)} = \frac{\sum\limits_{j=1}^{p} \varphi_j(x_j) + \alpha(y)}{\sum\limits_{j=1}^{p} \psi_j(x_j) + \beta(y)} \to \min, \tag{5.50}$$

$$h_j^l(x_j) + f_j^l(y) \le a_j^l \quad (l = \overline{1,m_j};\ j = \overline{1,p}), \tag{5.51}$$

в которой имеем p групп переменных $x_j = (x_j^1, x_j^2, \ldots, x_j^{n_j}) \in R^{n_j}$, образующих вектор $x = (x_1, x_2, \ldots, x_p) \in R^n$, где $n = \sum\limits_{j=1}^{p} n_j$, а также переменные $y = (y_1, y_2, \ldots, y_k) \in R^k$. На множестве каждой группы переменных x_j определены непрерывные функции

$$\varphi_j,\ \psi_j,\ h_j^l \quad (l = \overline{1,m_j}):\ R^{n_j} \to R,$$

а на множестве переменных y определены непрерывные функции

$$\alpha,\ \beta,\ f_j^l \quad (l = \overline{1,m_j}):\ R^k \to R.$$

Если предположить, что функции $\varphi_j(x_j)$ и $-\psi_j(x_j)$ выпуклы в R^{n_j}, а $\alpha(y)$ и $-\beta(y)$ выпуклы в R^k, то функции $\varphi(x,y)$ и $-\psi(x,y)$ также будут выпуклыми. Предположим также, что функции $h_j^l(x_j)$ и $f_j^l(y)$ выпуклы на всем множестве определения. Тогда задача (5.50), (5.51) будет задачей дробно-выпуклого программирования.

Обозначим через $M(x,y)$ множества значений переменных (x,y), удовлетворяющих ограничениям (5.51), и предположим, что $\psi(x,y) > 0$, а $\varphi(x,y) \ge 0$ для всех $(x,y) \in M(x,y)$. Тогда функция

$$F(x,y) = \frac{\varphi(x,y)}{\psi(x,y)}$$

является квазивыпуклой на множестве $M(x,y)$.

Применив схему декомпозиции по переменным для решения задачи (5.50), (5.51) и фиксируя переменные $y = \overline{y}$, получим задачу:

$$F(x,\overline{y}) = \frac{\varphi(x,\overline{y})}{\psi(x,\overline{y})} = \frac{\sum\limits_{j=1}^{p} \varphi_j(x_j) + \alpha(\overline{y})}{\sum\limits_{j=1}^{p} \psi_j(x_j) + \beta(\overline{y})} \to \min, \tag{5.52}$$

$$h_j^l(x_j) \le b_j^l(\overline{y}) \quad (l = \overline{1,m_j};\ j = \overline{1,p}), \tag{5.53}$$

в которой $b_j^l(\overline{y}) = a_j^l - f_j^l(\overline{y})$ $(l = \overline{1,m_j};\ j = \overline{1,p})$ – постоянные числа.

Обозначим через $S(\overline{y})$ множество значений переменных x, удовлетворяющих ограничениям (5.53), и определим функцию $\Phi(y)$ по формуле

$$\Phi(\overline{y}) = \min_{x \in S(\overline{y})} \frac{\varphi(x,\overline{y})}{\psi(x,\overline{y})}.$$

Построим функцию Лагранжа задачи (5.52), (5.53):

$$L(x,v) = \frac{\varphi(x,\overline{y})}{\psi(x,\overline{y})} + \sum_{j=1}^{p} \sum_{l=1}^{m_j} v_j^l \left[h_j^l(x_j) - b_j^l(\overline{y})\right].$$

Здесь v_j^l – множители Лагранжа ограничений (5.53). Тогда значения обобщенного градиента функции $\Phi(y)$ в точке $y = \overline{y}$ определяются по формуле

$$G_\Phi(\overline{y}) = G_L^y(x(\overline{y}),\overline{y}),$$

которая для данной задачи имеет вид

$$\begin{aligned} G_\Phi(\overline{y}) = \frac{\psi(x(\overline{y}),\overline{y})\, G_\alpha^y(x(\overline{y}),\overline{y}) - \varphi(x(\overline{y}),\overline{y})\, G_\beta^y(x(\overline{y}),\overline{y})}{[\psi(x(\overline{y}),\overline{y})]^2} + \\ + \sum_{j=1}^{p} \sum_{l=1}^{m_j} v_j^l(\overline{y})\, G_{g_j^l}^y(x(\overline{y}),\overline{y}), \end{aligned} \tag{5.54}$$

где $G_\alpha^y(x(\overline{y}),\overline{y})$, $G_\beta^y(x(\overline{y}),\overline{y})$ и $G_{g_j^l}^y(x(\overline{y}),\overline{y})$ – соответствующие проекции обобщенных градиентов функций $\alpha(y)$, $\beta(y)$ и $f_j^l(y)$ в точке $\overline{z} = (x(\overline{y}),\overline{y})$ на R_y^k, а $\overline{z} = (x(\overline{y}),\overline{y})$ – оптимальное решение задачи (5.52), (5.53) и $v_j^l(\overline{y})$ – соответствующие множители Лагранжа при $y = \overline{y}$.

Таким образом, решение задачи (5.50), (5.51) сводится к минимизации строго квазивыпуклой функции $\Phi(y)$. Тогда на $(t+1)$-м шаге алгоритма обобщенного градиентного метода (решения задачи минимизации функции $\Phi(y)$) необходимо провести следующие основные три его этапа.

1. Решить задачу (5.52), (5.53) при фиксированных $\overline{y} = \overline{y}^t$. Для определения значений обобщенного градиента функции $\Phi(y)$ по формуле (5.54) необходимо знать как решение $x(\overline{y})$ задачи (5.52), (5.53), так и множители Лагранжа $v(\overline{y})$, соответствующие ограничениям (5.53).

Задача (5.52), (5.53) при фиксированных $\overline{y} = \overline{y}^t$ имеет дробный (несепарабельный) функционал и блочно-диагональные ограничения. Для ее решения используем параметрический метод. Для этого рассмотрим следующую задачу параметрического выпуклого программирования:

$$\sum_{j=1}^{p}(\varphi_j(x_j) - \lambda\psi_j(x_j)) + \alpha(\overline{y}) - \lambda\beta(\overline{y}) \to \min,$$

$$h_j^l(x_j) \le b_j^l(\overline{y}) \quad (l = \overline{1, m_j};\ j = \overline{1, p}).$$

В результате применения параметрического метода, при фиксированном значении параметра λ, задача (5.52), (5.53) сводится к решению p задач выпуклого программирования вида

$$\varphi_j(x_j) - \lambda\psi_j(x_j) \to \min,$$

$$h_j^l(x_j) \le b_j^l(\overline{y}) \quad (l = \overline{1, m_j}),$$

для каждой группы переменных x_j в отдельности.

Замечание. *Как и в предыдущих случаях, в параметрическом методе решения задачи* (5.52), (5.53) *достаточно выполнить один шаг с фиксированном значении параметра*

$$\lambda = \lambda^{t+1} = F\big(x(\overline{y}^t, \lambda^t), \overline{y}^t\big),$$

где $x(\overline{y}^t, \lambda^t)$ найденное решение задачи (5.52), (5.53) *на предыдущем шаге обобщенного градиентного метода, а задача выпуклого программирования для каждого j продолжает свое решение с достигнутой точке $x_j(g^t, \lambda^t)$.*

Пусть $x(\overline{y})$ – оптимальное (или некоторое допустимое) решение задачи (5.52), (5.53) найденное по алгоритму параметрического метода, а $v(\overline{y})$ – множители Лагранжа ограничения (5.53).

2. Определить значения обобщенного градиента функции $\Phi(y)$ в точке $y = \overline{y}^t$ по формуле (5.54).

3. Вычислить новые значения

$$\overline{y}^{t+1} = \overline{y}^{t} - \gamma_{t+1}\, G_{\Phi}(\overline{y}^{t}),$$

где γ_{t+1} – величина шага.

В предыдущей главе схема декомпозиции по переменным была использована для решения различных задач дробно-линейного программирования блочной структуры. Также она была опробована и на задачах дробно-линейного программирования транспортного типа (см. главу 8).

5.7. Схема декомпозиции по ресурсам в дробно-выпуклом программировании

Рассмотрим схему декомпозиции по ресурсам для решения задач дробно-выпуклого программирования, в которой перераспределение ресурсов осуществляется путем решения задачи недифференцируемой оптимизации субградиентными методами.

5.7.1. Схема декомпозиции по ресурсам для задач дробно-выпуклого программирования

Пусть $x_1, x_2, \ldots, x_p$ ($x_j \in R^{n_j},\ j = \overline{1,p}$) – группы переменных, образующих вектор $x = (x_1, x_2, \ldots, x_n) \in R^n$, где $n = \sum\limits_{j=1}^{p} n_j$.

Рассмотрим теперь задачу дробно-выпуклого программирования с аддитивно-сепарабельными функциями $\varphi(x)$ и $\psi(x)$, а именно

$$F(x) = \frac{\varphi(x)}{\psi(x)} = \frac{\sum\limits_{j=1}^{p} \varphi_j(x_j)}{\sum\limits_{j=1}^{p} \psi_j(x_j)} \;\to \min, \tag{5.55}$$

$$\sum_{j=1}^{p} F_j^i(x_j) \le b_i \quad (i = \overline{1,m}) \tag{5.56}$$

$$W_j(x_j) \le \overline{b}_j \quad (j = \overline{1,p}), \tag{5.57}$$

где функции Φ_j, F_j^i – выпуклые функции; W_j – выпуклые векторные функции от соответствующих переменных; $b = (b_1, b_2, \ldots, b_m)$ – фиксированный вектор глобальных ресурсов; $\overline{b}_j$ – вектор констант соответствующего размера.

Представим вектор ресурсов b в виде суммы

$$b = z_1 + z_2 + \cdots + z_p, \quad z_j \geq 0 \ (j = \overline{1,p}), \tag{5.58}$$

а ограничения (5.56) заменим на

$$F_j^i(x_j) \leq z_j^i \quad (i = \overline{1,m};\ j = \overline{1,p}), \tag{5.59}$$

где $z_j = (z_j^1, z_j^2, \ldots, z_j^m)$ – вектор, показывающий распределение ресурса.

Получим задачу дробно-выпуклого программирования с блочной структурой системы ограничений:

$$F(x) = \frac{\sum\limits_{j=1}^{p} \varphi_j(x_j)}{\sum\limits_{j=1}^{p} \psi_j(x_j)} \ \rightarrow \min, \tag{5.60}$$

$$F_j^i(x_j) \leq z_j^i \quad (i = \overline{1,m};\ j = \overline{1,p}), \tag{5.61}$$

$$W_j(x_j) \leq \overline{b}_j \quad (j = \overline{1,p}). \tag{5.62}$$

Для решения задачи (5.60)–(5.62) применим параметрический метод, и как показано ранее, она распадается на p задач отдельно для каждого блока.

Таким образом, при $v = v_k$ на k-м шаге параметрического метода решаются p подзадач вида:

$$F_j(x_j, v_k) = \varphi_j(x_j) - v_k \psi_j(x_j) \ \rightarrow \min, \tag{5.63}$$

$$F_j^i(x_j) \leq z_j^i \quad (i = \overline{1,m}), \tag{5.64}$$

$$W_j(x_j) \leq \overline{b}_j. \tag{5.65}$$

Пусть $x_j^*(z_j)$ – оптимальное решение задачи (5.63)–(5.65) при $v = v^*$, для которого достигается оптимальное решение задачи (5.60)–(5.62), найденное параметрическим методом при фиксированном z_j.

Построим функцию

$$\Phi(z) = \frac{\sum\limits_{j=1}^{p} \varphi_j(x_j^*(z_j))}{\sum\limits_{j=1}^{p} \psi_j(x_j^*(z_j))}$$

относительно переменных z_j и рассмотрим задачу:

$$\Phi(z) = \frac{\sum\limits_{j=1}^{p} \alpha_j(z_j)}{\sum\limits_{j=1}^{p} \beta_j(z_j)} \to \min, \tag{5.66}$$

$$\sum_{j=1}^{p} z_j \le b, \quad z_j \ge 0 \ (j = \overline{1,p}), \tag{5.67}$$

в которой значения функций α_j и β_j относительно переменных z_j находятся алгоритмически, путем решения задачи (5.60)–(5.62) при фиксированных значениях z_j. Можно показать, что функция

$$\Phi(z) = \frac{\alpha(z)}{\beta(z)} = \frac{\sum\limits_{j=1}^{p} \alpha_j(z_j)}{\sum\limits_{j=1}^{p} \beta_j(z_j)}$$

является квазивыпуклой функцией относительно переменных z_j, а ее субградиент определяется по формуле

$$g_\Phi(z) = \left\{-\lambda_j^i(z_j)/\beta(z) \quad (j = \overline{1,p}; i = \overline{1,m}\,)\right\} \tag{5.68}$$

где $\lambda_j^i(z_j)$ – оптимальные значения множителей Лагранжа, соответствующие ограничениям (5.61), для функции Лагранжа вида

$$L(x,z,\lambda) = \frac{\sum\limits_{j=1}^{p} \varphi_j(x_j) + \sum\limits_{j=1}^{p}\sum\limits_{i=1}^{m} \lambda_j^i(F_j^i(x_j) - z_j^i)}{\sum\limits_{j=1}^{p} \psi_j(x_j)}.$$

Так как функционал $\Phi(z)$ недифференцируем, то для решения задачи (5.66), (5.67) применяются субградиентные методы.

Таким образом, схема декомпозиции по ресурсам для задачи (5.55)–(5.57) заключается в следующем.

Решается задача (5.66), (5.67) субградиентным методом, на $(t+1)$-м шаге которого необходимо выполнить три его этапа.

1. Решить задачу (5.60)–(5.62) при фиксированных $z_j^i(t)$ параметрическим методом, которая распадается на p задач типа (5.63)–(5.65) и найти оптимальные значения множителей Лагранжа $\lambda_j^i(z_j^i)$, соответствующих ограничениям (5.61).
2. Найти значения обобщенного градиента функции $\Phi(z)$ в точке z с координатами $z_j^i = z_j^i(t)$ по формуле (5.68), в которой значение функции $\beta(z)$ в точке $z = z(t)$ определяется с учетом найденного оптимального плана $x_j^*(z_j^i(t))$ задачи (5.60)–(5.62).
3. Вычислить новые значения параметров z_j^i по формулам одного из субградиентных методов с проектированием на линейное многообразие.

Из предложенного алгоритма следует, что последовательно решаются следующие две группы задач:

1) задача (5.66), (5.67) (координирующая задача) решается на верхнем уровне, в результате чего происходит перераспределение глобальных ресурсов b;

2) задача (5.63)–(5.65) (локальные задачи) решаются на втором уровне, в результате этого с помощью множителей Лагранжа z_j^i оцениваются выделенные объемы глобальных ресурсов λ_j^i.

Замечание. *При решении задачи* (5.60)–(5.62) *параметрическим методом выполняется только одна итерация при фиксированном значении параметра*

$$v = v^{t+1} = \frac{\sum\limits_{j=1}^{p} \varphi_j\big(x_j^*(z(t), v^t)\big)}{\sum\limits_{j=1}^{p} \psi_j\big(x_j^*(z(t), v^t)\big)},$$

где $x_j^*(z(t), v^t)$ – *оптимальные решения задач* (5.63)–(5.65), *найденные на предыдущем шаге обобщенного градиентного метода.*

Сложность данного метода обусловлена процедурой перераспределения глобальных ресурсов. В литературе известны и другие процедуры перераспределения.

5.7.2. Схема декомпозиции по ресурсам для решения сепарабельных задач дробно-выпуклого программирования

Рассмотрим сначала следующий класс сепарабельных задач дробно-выпуклого программирования блочно-диагональной структуры со связующими ограничениями:

$$F_1(x) = \max_j \{\Phi_j(x_j) = \frac{\varphi_j(x_j)}{\psi_j(x_j)}\} \to \min, \tag{5.69}$$

$$\sum_{j=1}^{p} F_j^i(x_j) \le b_i \quad (i = \overline{1,m}), \tag{5.70}$$

$$W_j(x_j) \le \overline{b}_j \quad (j = \overline{1,p}). \tag{5.71}$$

Здесь φ_j, $-\psi_j$, и F_j^i – выпуклые функции; W_j – выпуклые вектор-функции от соответствующих переменных; $b = (b_1, b_2, \ldots, b_m)$ – вектор заданных ресурсов; $\overline{b}_j$ – вектор констант соответствующей размерности.

Одновременно с задачей (5.69)–(5.71), рассмотрим и задачу минимизации функционала

$$F_2(x) = \sum_{j=1}^{p} \Phi_j(x_j) \tag{5.72}$$

при ограничениях (5.70), (5.71).

Для решения задач (5.69)–(5.72) может быть использована схема декомпозиции по ограничениям. Как было показано выше, это приводит к задачам минимизации одного из функционалов:

$$L_1(x,u) = \max_j \Phi_j(x_j) + \sum_{i=1}^{m} u_i \Big[\sum_{j=1}^{p} F_j^i(x_j) - b_i\Big],$$

$$L_2(x,u) = \sum_{j=1}^{p} \Phi_j(x_j) + \sum_{i=1}^{m} u_i \Big[\sum_{j=1}^{p} F_j^i(x_j) - b_i\Big],$$

при ограничениях (5.71) и фиксированном $u = \overline{u}$.

Поскольку функционал $L_1(x,u)$ не является сепарабельной функцией, то такую задачу нельзя решать для каждого блока в отдельности. Для этого предлагается применять некоторый последовательный алгоритм поочередного решения дробно-выпуклых и выпуклых задач.

Функционал $L_2(x, u)$ является сепарабельным на каждой группе переменных x_j, однако при этом необходимо решать обобщенную задачу выпуклого и дробно-выпуклого программирования для каждого блока в отдельности, что не всегда может обеспечить глобальный минимум. Иное получается при применении схемы декомпозиции по ресурсам с последующим использованием субградиентных методов.

Представим вектор ресурсов в виде равенств (3.57). Тогда задачи (5.69)–(5.71) и (5.70)–(5.72) разлагаются на p подзадач:

$$\Phi_j(x_j) \to \min, \tag{5.73}$$

$$F_j^i(x_j) \le z_j^i \quad (i = \overline{1, m}), \tag{5.74}$$

$$W_j(x_j) \le \bar{b}_j. \tag{5.75}$$

Пусть $x_j^*(z_j)$ – оптимальное решение задачи (5.73)–(5.75) при заданном z_j, а $\{u^*(z_j)\}$ – оптимальные значения множителей Лагранжа, соответствующие ограничениям (5.74), где функция Лагранжа определяется по формуле

$$L_j(x_j, u_j, v_j) = \frac{\varphi_j(x_j) + \sum\limits_{i=1}^{m} u_j^i \left[F_j^i(x_j) - z_j^i\right]}{\psi_j(x_j)}.$$

Определим функции

$$\Phi_1(z) = \max_j \Phi_j(x_j^*(z_j))$$

и

$$\Phi_2(z) = \sum_{j=1}^{p} \Phi_j(x_j^*(z_j)).$$

Так как функции $\Phi_j(x_j^*(z_j))$ являются квазивыпуклыми, то функция $\Phi_1(z)$ – также квазивыпуклая, а ее субградиент имеет вид

$$G_\Phi(z) = \left\{ g_j^i(z_j) = \frac{-u_j^{i*}(z_j)}{\psi_j(x_j^*(z_j))} \right\}. \tag{5.76}$$

Тогда задача (5.69)–(5.71) сводится к решению задачи минимизации квазивыпуклого функционала $\Phi_1(z)$ при ограничениях (3.57), что позволяет найти глобальный минимум. Для решения этой задачи может быть использован субградиентный метод с проектированием на линейные многообразия. На $(k+1)$-м шаге таких методов необходимо выполнить три этапа.

1. При фиксированных $z_j = z_j^k$ решить задачи (5.73)–(5.75) и найти оптимальные решения $x_j^*(z_j^k)$ и соответствующие множители Лагранжа $u_j^*(z_j^k)$ ограничений (5.74).
2. Определить значения обобщенного градиента функции $\Phi_1(z)$ в точке $z_j = z_j^k$ по формуле (5.76).
3. Вычислить новые значения z_j^{k+1} по одному из субградиентных методов с проектированием на линейном многообразии.

Что касается функции $F_2(z)$, то она не всегда является квазивыпуклой. Поэтому применение субградиентного метода для ее минимизации при линейных ограничениях (3.57) не всегда позволяет найти глобальный минимум.

Замечание. *Для вычисления значений обобщенного градиента задачи* (5.73)–(5.75) *решаются по следующей схеме*:
- *находим* $v_j^{k+1} = \varphi_j\big(x_j^*(z_j^k, v_j^k)\big)/\psi_j\big(x_j^*(z_j^k, v_j^k)\big)$,
- *решаем задачу выпуклого программирования*

$$\varphi_j(x_j) - v_j^{k+1}\psi_j(x_j) \to \min,$$

$$F_j^i(x_j) \le z_j^{ik+1} \quad (i = \overline{1, m}),$$

$$w_j(x_j) \le \overline{b}_j$$

и находим оптимальное решение $x_j^*(z_j^{k+1}, v_j^{k+1})$ *и значения множителей Лагранжа* $u_j^*(z_j^{k+1}, v_j^{k+1})$.

Глава 6

Обобщенные задачи дробного программирования

В данной главе рассматриваются некоторые обобщения задач дробного программирования, в которых используются операции взятия максимума, суммирования или мультипликатирования на множестве линейных, дробно-линейных, выпуклых или дробно-выпуклых функций. Для их решения предлагаются параметрические методы, прямые методы недифференцируемой оптимизации и схемы декомпозиции по ограничениям с применением субградиентных методов. Рассматриваются задачи многокритериальной оптимизации, которые сводятся к решению одной из обобщенных задач дробного программирования.

Для различных сепарабельных задач обобщенного дробно-выпуклого программирования блочной структуры предлагаются схемы декомпозиции по ограничениям, переменным и ресурсам, в сочетание с методами недифференцируемой оптимизации.

Рассматриваются также задачи обобщенного дробно-выпуклого программирования с целевыми функциями в виде суммы двух функционалов от различных групп переменных, когда хотя бы один из них является дробным. Ограничения для таких задач имеют блочную структуру по одной группе переменных с наличием связующих блоков по другой группе переменных. Для их решения предлагаются одноуровневые или двухуровневые алгоритмы, для каждого из которых используются схемы декомпозиции и субградиентные методы.

6.1. Задачи дискретного минимакса и обобщенного дробного программирования

1. Задачи дискретного минимакса. Одной из известных оптимизационных задач является задача дискретного минимакса, которая состоит в минимизации максимума конечного числа функций. Пусть заданы функции $\varphi_i(x),\ i \in I$, где $I = \{1, \ldots, m\}$. Тогда задача дискретного минимакса состоит в минимизации функции

$$f_0(x) = \max_{i \in I} \varphi_i(x),$$

когда x меняется во всем пространстве R^n или когда $x \in S$, где S – некоторый компакт из R^n.

Задача безусловной минимизации функции $f_0(x)$, которая определяется в результате взятия операции максимума конечного числа функций, называется задачей дискретного минимакса и исследована в работах [149, 160, 161, 175, 580, 606, 608, 620, 651, 657, 701, 736]. Для ее решения приводятся различные модификации алгоритмов решения задач математического и выпуклого программирования, в частности метод наискорейшего спуска, метод последовательных приближений, субградиентные методы, ε-квазиградиентные методы, методы двойственных направлений и чебышевских центров. Особенно следует отметить семейство методов линеаризации и их модификаций, на каждом шаге которых решаются вспомогательные задачи линейного или квадратичного программирования.

A. *Безусловная задача дискретного минимакса.* Если предположить, что функции $\varphi_i(x),\ i \in I$ непрерывно дифференцируемые, то рассматривая эквивалентную задачу

$$\min_{x,t}\{t\colon\ \varphi_i(x) \le t, \quad i \in I\}$$

получаем следующее утверждение: для того чтобы точка x^* была точкой минимума функции $t = f_0(x^*) = \max\limits_{i \in I} \varphi_i(x)$, необходимо, чтобы нашлись такие числа $u_i,\ i \in I$, что

$$\sum_{i \in I} u_i \frac{\partial \varphi_i(x^*)}{\partial x_j} = 0 \quad (j = \overline{1, n});$$

$$u_i\left(\varphi_i(x^*) - f_0(x^*)\right) = 0 \quad (i \in I);$$

$$\sum_{i\in I} u_i = 1, \quad u_i \geq 0, \quad i \in I.$$

Данные условия получаются, если для эквивалентной задачи дискретного минимакса построить функцию Лагранжа в виде

$$L(x,t,u) = t + \sum_{i\in I} u_i(\varphi_i(x) - t) = t\Big(1 - \sum_{i\in I} u_i\Big) + \sum_{i\in I} u_i\varphi_i(x)$$

и рассмотреть необходимые условия экстремума Куна-Таккера

$$\frac{\partial L(x,t,u)}{\partial x_j} = \sum_{i\in I} u_i \frac{\partial \varphi_i(x)}{\partial x_j} = 0 \qquad (j = \overline{1,n}),$$

$$\frac{\partial L(x,t,u)}{\partial t} = 1 - \sum_{i\in I} u_i = 0,$$

$$\frac{\partial L(x,t,u)}{\partial u_i} = \varphi_i(x) - t \leq 0, \quad i \in I.$$

Также, в соответствии с общей теорией построения двойственных лагранжевых задач, имеем функцию

$$L^*(u) = \min_{t,x} L(x,t,u) = \begin{cases} -\infty, & \text{если} \quad \sum\limits_{i\in I} u_i \neq 1, \\ \sum\limits_{i\in I} u_i\varphi_i(x), & \text{если} \quad \sum\limits_{i\in I} u_i = 1. \end{cases}$$

Тогда двойственная задача дискретного минимакса может быть записана в виде:

$$\max_u L^*(u),$$

$$\sum_{i\in I} u_i = 1,$$

$$u_i \geq 0, \quad i \in I,$$

в которой функция $L^*(u)$ при данных ограничениях имеет вид

$$L^*(u) = \min_x \sum_{i\in I} u_i\varphi_i(x).$$

Функция $L^*(u)$ является кусочно-линейной, вогнутой и недифференцируемой. Тогда для ее максимизации могут быть использованы субградиентные методы. Пусть на r-м шаге такого метода имеем точку u^r, для которой $u_i^r \geq 0$, $i \in I$. Тогда на $(r+1)$-м шаге выполняем следующие основные этапы субградиентного метода.

1. При фиксированном $u = u^r$ решаем безусловную задачу выпуклой оптимизации

$$\min_x \sum_{i\in I} u_i^r \varphi_i(x),$$

и находим ее оптимальное решение $x^*(u^r)$.

2. Находим значение обобщенного градиента функции $L^*(u)$ в точке u^r по формуле

$$g_{L^*}(u^r) = \{\varphi_i(x^*(u^r)), \quad i \in I\}.$$

3. Находим новую точку u^{r+1} по формуле

$$\widetilde{u}_i^{r+1} = \max\{0,\, u_i^r + \gamma_{r+1}\varphi_i(x^*(u^r))\}, \quad i \in I,$$

где γ_{r+1} – величина шага субградиентного метода.

4. Проводим операцию проектирования точки $\widetilde{u}^{r+1}$ на линейное многообразие $U = \left\{u_i\colon \sum\limits_{i\in I} u_i = 1, \quad u_i \geq 0, \quad i \in I\right\}$, т.е. находим $u^{r+1} = \Pi_U(\widetilde{u}^{r+1})$, где $\Pi_U(\widetilde{u}^{r+1})$ – оператор проектирования точки $\widetilde{u}^{r+1}$ на множество U.

Б. *Условная задача дискретного минимакса.* Известно [657], что в случае задачи дискретного минимакса выпуклых функций при выпуклых ограничениях

$$f_0(x) = \max_{i\in I} \varphi_i(x) \to \min,$$

$$h_k(x) \leq 0 \quad (k = \overline{1,p}),$$

в которой функции $\varphi_i(x)$, $i \in I$ и $h_k(x)$ $(k = \overline{1,p})$ непрерывны и выпуклы, для ее решения рассматривается эквивалентная задача выпуклого программирования

$$t \to \min;$$

$$\varphi_i(x) \leq t, \quad i \in I;$$

$$h_k(x) \leq 0 \quad (k = \overline{1,p}).$$

Полученное оптимальное решение (x^*, t^*) будет оптимальным и для исходной задачи.

Для данной эквивалентной задачи дискретного минимакса с ограничениями построим функцию Лагранжа

$$L(x,t,u,z) = t\left(1 - \sum_{i\in I} u_i\right) + \sum_{i\in I} u_i\varphi_i(x) + \sum_{k=1}^{p} z_k h_k(x),$$

где $z_k \geq 0$ $(k = \overline{1,p})$ – множители Лагранжа соответствующие ограничениям $h_k(x) \leq 0$ $(k = \overline{1,p})$.

Тогда двойственная задача, построенная на основе функции Лагранжа, имеет вид: найти

$$\max_{u,z} L^*(u,z),$$

при ограничениях

$$\sum_{i\in I} u_i = 1,$$

$$u_i \geq 0, \quad i\in I, \quad z_k \geq 0 \quad (k=\overline{1,p}),$$

где

$$L^*(u,z) = \min_{x,t} L(x,t,u,z).$$

Задача $\min\limits_{x,t} L(x,t,u,z)$ при фиксированных множителях Лагранжа u и z является задачей выпуклого программирования. Функция $L^*(u,z)$ также является кусочно-линейной, вогнутой и недифференцируемой, для максимизации которой могут быть использованы субградиентные методы. Схема применения таких методов аналогична той, которая применена для безусловной задачи дискретного минимакса, только максимизация проводится по переменным u и z.

Пусть на r-ом шаге субградиентного метода имеем точку (u^r, z^r), для которой $u_i^r \geq 0$, $i\in I$ и $z_k^r \geq 0$ $(r=\overline{1,p})$. Тогда на $(k+1)$-м шаге субградиентного метода выполняем следующие этапы.

1. При фиксированных $u=u^r$, $z=z^r$ и $t=0$ решаем безусловную задачу выпуклой оптимизации

$$\min_x \Big[\sum_{i\in I} u_i^r \varphi_i(x) + \sum_{k=1}^{p} z_k^r h_k(x)\Big]$$

и находим ее оптимальное решение $x^*(u^r,z^r)$.

2. Вычисляем значения обобщенных градиентов функции $L^*(u,z)$ в точке (u^r,z^r) по формулам

$$g_{L^*}^u(u^r,z^r) = \big\{\varphi_i\left(x^*(u^r,z^r)\right),\ i\in I\big\};$$

$$g_{L^*}^z(u^r,z^r) = \big\{h_k\left(x^*(u^r,z^r)\right), \quad k=\overline{1,p}\big\}.$$

3. Находим новую точку (u^{r+1}, z^{r+1}) по формулам

$$\widetilde{u}_i^{r+1} = \max\big\{0,\ u_i^r + \gamma_{r+1}\varphi_i\left(x^*(u^r,z^r)\right)\big\},\ i\in I;$$

$$z_k^{r+1} = \max\big\{0,\ z_k^r + \gamma_{r+1}h_k\left(x^*(u^r,z^r)\right)\big\} \quad (k=\overline{1,p}).$$

4. Проводим операцию проектирования точки $\widetilde{u}^{r+1}$ на линейное многообразии $U = \big\{u_i : \sum\limits_{i\in I} u_i = 1,\ u_i \geq 0,\ i\in I\big\}$, т.е. находим

$u^{r+1} = \Pi_U(\widetilde{u}^{r+1})$, где $\Pi_U(\widetilde{u}^{r+1})$ – оператор проектирования точки $\widetilde{u}^{r+1}$ на множество U.

Другой подход применения субградиентных методов может быть рассмотрен в случае недифференцируемой минимизации выпуклой функции $f_0(x)$.

Как известно, если функции $\varphi_i(x)$ являются выпуклыми и необязательно непрерывными и дифференцированными на всем пространстве определения R^n, то функция $f_0(x)$ является выпуклой и не везде дифференцируемой. В точках, где максимум достигается одновременно для нескольких функций $\varphi_i(x)$, имеет место разрыв обычного градиента, и в таких точках определяются значения субградиентов (обобщенных градиентов). Для решения задач дискретного минимакса выпуклых функций $\varphi_i(x)$ в работе [736] предложены два алгоритма метода обобщенного градиентного спуска с растяжением пространства, которые показали свою эффективность при решения задач с небольшим числом переменных, но при большом числе функций, участвующих в операции взятия максимума. В этой же работе приводится информация о решении задачи дискретного минимакса с числом переменных равным 1200 и числом функций $\varphi_i(x)$ равным 1536. Отметим, что для эффективного решения данных задач может быть использован и алгоритм метода эллипсоидов.

B. *Субградиентный метод решения безусловной задачи дискретного минимакса.* Применение субградиентных методов при решения задачи дискретного минимакса $f_0(x) = \max\limits_{i \in I} \varphi_i(x)$ состоит в следующем. Пусть имеем некоторую точку x_r, полученную на предыдущем шаге. Тогда выполняем следующие три этапа:

1) находим максимальное значение

$$f_0(x_r) = \max_{i \in I} \varphi_i(x_r) = \varphi_{i_r}(x_r),$$

где i_r – индекс функции $\varphi_i(x)$, для которой достигается максимум в точке x_r;

2) находим значение обобщенного градиента функции $f_0(x)$ в точке x_r по формуле

$$g_{f_0}(x_r) = g_{\varphi_{i_r}}(x_r) \text{ (так как } \; f_0(x_r) = \varphi_{i_r}(x_r)),$$

где $g_{\varphi_{i_r}}(x_r)$ – значение субградиента функции $\varphi_{i_r}(x)$ в точке x_r;

3) находим новую точку x_{r+1} по одной из формул перехода к очередной точке в субградиентном методе

$$x_{r+1} = x_r - \gamma_{r+1} g_{f_0}(x_r),$$

где γ_{r+1} – величина шага.

Решение x^*, к которому стремится последовательность $\{x_r\}$ при $r \to \infty$ и будет оптимальным для исходной задачи.

Г. *Метод эллипсоидов решения условной задачи дискретного минимакса.* В случае, когда необходимо решить задачу дискретного минимакса на множестве S, которое образовано ограничениями $h_k(x) \leq 0 \quad (k = \overline{1,p})$, т.е. $S = \{x \in R^n : h_k(x) \leq 0, \quad k = \overline{1,p}\}$, где $h_k(x)$ – выпуклые функции, то предлагается применить метод эллипсоидов (см. п. 2.5.). В данном случае функция $f_0(x)$ из задачи (2.29), (2.30) определяется с помощью взятия операции максимума от конечного числа функций $\varphi_i(x)$, т.е. $f_0(x) = \max\limits_{i \in I} \varphi_i(x)$. Тогда значения обобщенного градиента в задаче $\min\limits_{x \in S} f_0(x)$ в точке x_r определяется по формуле:

$$g(x_r) = \begin{cases} g_{f_0}(x_r) = g_{\varphi_{i_r}}(x_r), & \text{если} \quad \max\limits_{1 \leq k \leq p} h_k(x_r) = h_{k_r}(x_r) \leq 0; \\ g_{h_{k_r}}(x_r), \quad \text{если} \quad \max\limits_{1 \leq k \leq p} h_k(x_r) = h_{k_r}(x_r) > 0, & \end{cases}$$

где k_r – индекс для которого достигается поточечный максимум по функциям ограничений задачи, т.е. $h_{k_r}(x_r) = \max\limits_{1 \leq k \leq p} h_k(x_r)$, а i_r – индекс для которого достигается поточечный максимум по функциям целевого функционала, т.е. $\varphi_{i_r}(x_r) = \max\limits_{i \in I} \varphi_i(x_r)$, $g_{f_0}(x_r)$ – субградиент функции $f_0(x)$ в точке x_r, $g_{\varphi_{i_r}}(x_r)$ – субградиент функции $\varphi_{i_r}(x)$ в точке x_r, $g_{h_{k_r}}(x_r)$ – субградиент функции $h_{k_r}(x)$ в точке x_r.

Остальные вычисления проводятся в соответствие с методом эллипсоидов.

Таким образом, для решения задачи дискретного минимакса от конечного числа выпуклых функций, от которых не требуется непрерывности и дифференцируемости могут быть использованы субградиентные методы, которые с достаточной эффективностью обеспечат получение оптимального решения.

2. Задачи обобщенного дробного программирования. Одновременно с задачами дробного программирования сформулированы и другие задачи оптимизации, в которых функция цели содержит различные сочетания дробных функций и для которых применяются различные математические операции. К таким операциям относятся:

- операция взятия минимума или максимума от конечного числа функций или от конечного числа отношений двух функций;

- операция суммирования конечного числа функций или конечного числа отношений двух функций;

- операция мультипликатирования (умножения) конечного числа функций или конечного числа отношений двух функций;

- операция перечисления конечного числа функций или конечного числа отношений двух функций;

- операция отношения двух функций, каждая из которых получена в результате применения приведенных выше операций (минимума, максимума, суммирования, мультипликатирования или перечисления).

В результате применения таких операций получаем более сложные задачи оптимизации, как правило глобальной, для решения которых можно использовать точные или приближенные методы, различные эвристические процедуры или методы случайного поиска, а также другие методы и алгоритмы, позволяющие с достаточной точностью найти решение данных задач.

Приведем ниже математические постановки некоторых классов обобщенных дробных задач, которые будут далее рассмотрены в данной работе. Для этого предположим, что имеются два класса функций $\{\varphi_i(x)\}$ и $\{\psi_i(x), \quad i \in I\}$, которые участвуют в формировании функции цели оптимизационных задач, и класс функций $\{h_k(x), \quad k = \overline{1,p}\}$, определенных в пространстве R^n, свойства которых будут уточнены по необходимости. Оптимизация в таких задачах будет проводиться либо на всем пространстве R^n либо на заданном множестве S.

Задача дробного дискретного минимакса. Наряду с функцией $f_0(x)$ может быть определена и другая функция, как максимум отношений двух функций $\{\varphi_i(x), \ i \in I\}$ и $\{\psi_i(x), \ i \in I\}$. Тогда получим дробную задачу дискретного минимакса

$$\min_{x \in S} \left[f_1(x) = \max_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} \right].$$

Данную задачу также называют обобщенной дробной задачей и она была одной из первых обобщений задач дробного программирования. Такого рода задачи рассмотрены в работах [30], [155]–[157], [189, 268, 444], в которых приведены необходимые и достаточные условия оптимальности [13, 153, 328, 334, 339, 344, 345, 351, 372, 473, 476, 553, 563, 566, 568], разработаны различные двойственные задачи [12, 13, 41, 52, 53, 58, 59, 113, 118, 147, 157, 236, 268, 297, 329, 330, 333, 342, 451, 454, 476, 541, 550, 552, 554, 567, 568, 576] и для их решения предложены параметрические методы [52, 53, 90], [153]–[156], [633], методы частичной линеаризации [62, 63, 153], субградиентные методы [484, 496, 633], а также другие методы их решения [40, 62, 75, 191, 231, 252, 455, 484, 487, 489, 499, 505, 528, 567, 568, 633].

Задача суммирования дробных функций. Другое обобщение дробных задач оптимизации может быть сформулировано в виде суммы конечного числа дробных функций, а именно, найти

$$\min_{x\in S}\left[f_2(x)=\sum_{i\in I}\frac{\varphi_i(x)}{\psi_i(x)}\right].$$

Такая задача рассмотрена в различных работах, для решения которой предложены параметрические методы [15, 101, 176, 186, 445], методы внутренних точек [208], методы ветвей и границ [64], [67]–[70], [323, 325, 461, 533, 555] и другие методы глобальной оптимизации [70, 71, 159, 462, 463]. Также предложены методы решения дробных задач с функционалами в виде суммы линейных и дробно-линейных функций [9, 10, 109, 248, 249, 355, 477, 700], суммы двух [102, 631] или многих дробно-линейных функций [94, 135, 136, 305, 306, 313, 323, 325, 533, 555]. В данной работе мы предлагаем применить для ее решения субградиентные методы и схемы декомпозиции по ограничениям.

Задача мультипликативного дробного программирования. Если применить операцию произведения отношений линейных или нелинейных функций, то получим задачу мультипликативного дробного-программирования вида:

$$\min_{x\in S}\left[f_3(x)=\prod_{i\in I}\frac{\varphi_i(x)}{\psi_i(x)}\right],$$

которая рассмотрена в работах [169, 248, 273, 308, 312, 447], в которых по аналогии с выпуклыми мультипликативными задачами [65, 72, 216, 309, 310, 324], предложено использовать для ее решения общие методы глобальной оптимизации.

Мы предлагаем использовать и методы недифференцируемой оптимизации для решения задач с целевой функцией в виде произведения дробно-выпуклых или дробно-линейных функций.

Задача отношения максимума и минимума конечного числа функций. Если в качестве функции цели взять отношения двух противоположных операций максимума и минимума от различных функций, то получаем следующую обобщенную задачу дробного минимакса:

$$\min_{x\in S}\left[f_4(x)=\frac{\max\limits_{i\in I}\varphi_i(x)}{\min\limits_{i\in I}\psi_i(x)}\right],$$

для решения которой в данной работе предлагаются методы недифференцируемой оптимизации, двойственные параметрические методы, а также схемы декомпозиции для выпуклых и линейных функций.

Задача многокритериальной дробной оптимизации. В случае, когда функция цели задана в виде векторной функции, компоненты которой представляют собой дробные функции, то получаем многокритериальную задачу дробной оптимизации, а именно:

$$\min_{x\in S}\left[f(x)=\left(\frac{\varphi_1(x)}{\psi_1(x)},\frac{\varphi_2(x)}{\psi_2(x)},\ldots,\frac{\varphi_m(x)}{\psi_m(x)}\right)\right],$$

в которой порядок перечисления дробных функций может иметь определенное значение при ее решении, или же порядок перечисления функций является произвольным.

Смешанные обобщенные задачи дробной оптимизации. Другие классы обобщенных задач дробной оптимизации могут быть получены, когда в оптимизационных задачах переменные разделены на две группы x и y, на множестве которых в функционалах могут быть использованы различные операции, над входящими в них функциях. Пусть переменные задачи разделены на две группы $x \in R^{n_1}$ и $y \in R^{n_2}$, а $S(x,y)$ – множество допустимых решений для обеих групп переменных. Тогда могут быть рассмотрены следующие смешанные задачи дробной оптимизации: найти минимум одной из функций

$$F_1(x,y)=\frac{\sum\limits_{i\in I}\varphi_i(x)}{\sum\limits_{i\in I}\psi_i(x)}+\frac{\alpha(y)}{\beta(y)},$$

$$F_2(x,y)=\max_{i\in I}\frac{\varphi_i(x)}{\psi_i(x)}+\frac{\alpha(y)}{\beta(y)},$$

$$F_3(x,y)=\sum_{i\in I}\frac{\varphi_i(x)}{\psi_i(x)}+\frac{\alpha(y)}{\beta(y)},$$

$$F_4(x,y)=\prod_{i\in I}\frac{\varphi_i(x)}{\psi_i(x)}+\frac{\alpha(y)}{\beta(y)},$$

$$F_5(x,y)=\frac{\max\limits_{i\in I}\varphi_i(x)}{\min\limits_{i\in I}\psi_i(x)}+\frac{\alpha(y)}{\beta(y)}$$

при ограничения $(x,y)\in S(x,y)$.

Эти задачи могут быть записаны в общем виде:

$$\min_{x,y\,\in\,S(x,y)} F_r(x,y)\quad (r=\overline{1,5}).$$

В частности могут быть рассмотрены варианты, когда в дробных функционалах, функции знаменателя тождественно равны единицы, т.е. имеем либо $\{\psi_i(x) \equiv 1, \quad i \in I\}$ либо $\beta(y) \equiv 1$. В таких случаях имеем сочетание из суммы линейных и дробно-линейных, или выпуклых и дробно-выпуклых функций.

Для решения приведенных выше обобщенных задач дробного программирования разработаны различные алгоритмы и методы, основная часть из которых построены по следующей универсальной схеме.

1) Обобщенная задача дробной оптимизации сводится к некоторой эквивалентной задаче, в которой с помощью некоторых преобразований дробные функции исключаются из функционала, но добавляются в ограничениях. Полученная задача, как правило, имеет расширенные ограничения, за счет добавления дополнительных ограничений и дополнительных переменных. Впервые, аналогичный прием был использован для решения дробно-линейной задачи в работе [133], где предложена схема сведения ее к линейной задаче путем введения дополнительной связующей переменной и дополнительного ограничения. Однако такой прием неэффективен при решении специальных задач, например, транспортных задач с дробно-линейным функционалом. Другой прием был использован при сведение задачи дробно-линейного программирования к решению параметрической задачи линейного программирования [171, 172].

2) Для решения полученной эквивалентной задачи, в общем случае, которая является задачей нелинейного программирования, и, как правило, глобальной оптимизации, применяются общие методы ее решения. Одними из таких методов глобальной оптимизации являются метод ветвей и границ и методы типа внутренних точек, предназначенные для решения общих задач глобальной оптимизации.

Особенностью применения таких методов является итеративное решение задачи оптимизации с измененными целевыми функциями на каждой итерации, а также при наличии дополнительных ограничений и переменных. Такое преобразование исходных ограничений не позволяет применять эффективные алгоритмы и методы оптимизации для решения задач со специальной структурой ограничений, обладающие специфическими свойствами, достаточно хорошо изученные теоретически и имеющие широкое практическое применение, для решения которых разработано эффективное программное обеспечение. Это в первую очередь относится к задачам транспортного типа, задачам блочной структуры, задачам с линейными ограничениями, задачам о назначениях, распределительным задачам и тому подобным.

Для учета специфических свойств ограничений исходной задачи в данной работе предлагается использовать другие алгоритмы и методы, основанные на функции Лагранжа, двойственных задачах и схемах декомпозиции по ограничениям и переменным с применением субградиентных методов. Общая схема решения обобщенных дробных задач с использованием таких методов заключается в следующем:

1) для исходной обобщенной задачи дробного программирования рассматривается ее двойственная задача или строится эквивалентная задача нелинейного программирования с дополнительными ограничениями и переменными;

2) для полученной эквивалентной задачи строится функция Лагранжа относительно дополнительных ограничений, для которых вводятся соответствующие множители Лагранжа, с помощью которых введенные дополнительные ограничения и переменные возвращаются в функционал оптимизации;

3) рассматривается задача нахождения седловой точки функции Лагранжа;

4) задача нахождения седловой точки разбивается на два этапа;

- на первом, внешнем этапе, на множестве множителей Лагранжа решается задача недифференцируемой оптимизации одним из субградиентных методов;
- на втором, внутреннем этапе, на каждом шаге получаем вспомогательную задачу, как правило, той же выпуклости как и первоначальная, на множестве тех же исходных переменных и ограничений, для решения которой применяются эффективные алгоритмы, учитывающие специфику таких ограничений.

Также следует отметить, что внутренняя задача решается с целью нахождения значений субградиента в текущей точке. Так как субградиентные методы устойчивы относительно точности определения субградиентов, то подзадача может быть решена приближенными или эвристическими методами с достаточно высокой точностью сходимости к ее оптимальному решению при приближении к седловой точке функции Лагранжа.

По данной схеме предлагается решить рассмотренные выше задачи обобщенного дробного программирования. Процесс анализа каждого класса обобщенных задач дробного программирования включает следующие этапы:

1) строится эквивалентная задача;
2) приводятся необходимые и достаточные условия оптимальности;
3) рассматриваются двойственные задачи;
4) строится параметрическая задача;
5) используются параметрические методы;

6) строятся функции Лагранжа;

7) используется схема декомпозиции по ограничениям или переменным, с применением субградиентных методов;

8) применяются эффективные алгоритмы и методы решения внутренних задач.

Далее приведенная выше схема решения обобщенных дробных задач конкретизируется для задач дробно-линейной оптимизации, задач блочной структуры с сепарабельными обобщенными дробными функционалами, а также для обобщенных дробных задач транспортного типа.

Для каждой обобщенной дробной нелинейной задачи рассматриваются соответствующие задачи обобщенного дробно-линейного программирования, для решения которых используются метод частичной линеаризации, параметрический метод и схемы декомпозиции по ограничениям, с применением методов недифференцируемой оптимизации. На каждом шаге субградиентного метода решаются задачи линейного программирования на множестве исходных ограничений.

6.2. Задача обобщенного дробно-выпуклого программирования

Рассмотрим следующую дробно-выпуклую задачу дискретного минимакса, или так называемую обобщенную задачу дробно-выпуклого программирования в смысле минимизации функционала, полученного в результате применения операции взятия максимума дробных функций:

$$f(x) = \max_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} \to \min, \tag{6.1}$$

$$h_k(x) \le 0 \quad (k = \overline{1,p}), \tag{6.2}$$

в которой функции $\{\varphi_i(x),\, i \in I\}$, $\{-\psi_i(x),\, i \in I\}$ и $\{h_k(x),\ k = \overline{1,p}\}$ – непрерывны, выпуклы и дифференцируемы на R^n. Предполагается также что функции $\varphi_i(x) \ge 0, \quad i \in I$, а $\psi_i(x) > 0, \quad i \in I$ для всех x, удовлетворяющие ограничениям (6.2).

Задача (6.1), (6.2) в литературе встречается как задача обобщенного дробно-выпуклого программирования [155]–[157], [189, 268, 444]. Большинство алгоритмов решения задач обобщенного дробно-выпуклого программирования основаны на идеях параметрического метода [171, 172, 260, 261], который был использован для этих целей в работах [52, 53, 62, 63, 90, 153, 156, 189, 192, 633]. Различные теоретические вопросы и свойства задач обобщенного дробно-выпуклого программирования рассмотрены в работах [12, 13, 27, 41, 52, 53, 58, 153, 189, 328, 344, 345, 351, 372, 473, 476, 563, 565, 566, 568], в которых кроме параметрического метода предложены и другие алгоритмы. В работах [38, 472] предложен метод секущих плоскостей, который основан на решение вспомогательных задач нелинейной и негладкой оптимизации. Теоретические свойства задач дробного программирования, изученные в работах [32, 36, 148, 359, 438, 493, 494] легли в основу разработки новых точных и приближенных алгоритмов решения обобщенных задач дробно-выпуклого программирования [40, 58, 62, 75, 185, 252, 258, 274, 351, 376, 395, 423, 455, 505].

В данной монографии для решения обобщенной задачи дробно-выпуклого программирования приводятся прямые субградиентные методы, нелинейный параметрический метод, установлены необходимые и достаточные условия оптимальности, а также рассматриваются двойственные к ней задачи и функция Лагранжа. Приводятся схемы декомпозиции по ограничениям для решения обобщенных задач дробно-выпуклого и дробно-линейного программирования [633, 484, 496].

1. Построение эквивалентной задачи. Для задачи (6.1), (6.2) также может быть рассмотрена эквивалентная ей задача нелинейного программирования

$$t \to \min;$$

$$\varphi_i(x) - t\,\psi_i(x) \le 0, \quad i \in I;$$

$$h_k(x) \le 0 \quad (k = \overline{1,p});$$

$$t \ge 0.$$

В результате решения данной задачи, которая в общем случае может быть и задачей глобальной оптимизации, получаем и оптимальное решение исходной задачи. В зависимости от класса линейных или нелинейных функций $\varphi_i(x), \psi_i(x), \quad i \in I$ и $h_k(x) \quad (k = \overline{1,p})$ для решения эквивалентной задачи применяются различные алгоритмы и методы соответствующей эффективности.

Однако такой подход не является самым эффективным в случае задач со специализированными ограничениями (блочные, транспортного типа и др.). Также, наличие в ней произведений типа $t{\cdot}\psi_i(x)$ выводит исходную задачу из класса задач выпуклого программирования. Поэтому для решения обобщенной задачи дробно-выпуклого программирования необходимо использовать другие подходы и методы, в которых достигается желаемая эффективность.

2. Необходимые и достаточные условия. Введем обозначение

$$S = \{x\colon h_k(x) \le 0 \quad (k = \overline{1,p})\}$$

и предположим, что $\varphi_i(x) \ge 0, \quad i \in I$ и $\psi_i(x) > 0, \quad i \in I$ для всех $x \in S$. Тогда функция $f(x)$ является квазивыпуклой на S и не везде дифференцируемой на S.

Рассмотрим задачу обобщенного дробно-выпуклого программирования: найти

$$v^* = f(x^*) = \min_{x\in S}\max_{i\in I}\frac{\varphi_i(x)}{\psi_i(x)}. \tag{6.3}$$

Для анализа задачи (6.3) рассматривается эквивалентная задача параметрического программирования, которая является более удобной для анализа и решения, чем первоначальная дробная задача. Такой подход был использован в работах [52, 53, 62, 63, 90, 153, 156]. В алгоритме нелинейного параметрического метода решение задачи (6.3) сводится к решению параметрической задачи

$$F(v) = \min_{x\in S}\max_{i\in I}\big(\varphi_i(x) - v\psi_i(x)\big). \tag{6.4}$$

Для установления необходимых и достаточных условий оптимальности задачи (6.3) используется эквивалентная задача (6.4). Для этих целей рассмотрим следующую задачу, которая эквивалентна задаче (6.4) и соответственно задаче (6.3):

$$t \to \min, \tag{6.5}$$

$$\varphi_i(x) - v\psi_i(x) \le t \quad (i = \overline{1,m}), \tag{6.6}$$

$$h_k(x) \le 0 \quad (k = \overline{1,p}). \tag{6.7}$$

Имеют место следующие утверждения [153], характеризующие связь между задачами (6.3), (6.4) и (6.5)–(6.7).

Лемма 6.1. *Если задача* (6.3) *имеет оптимальное решение* x^* *с оптимальным значением целевой функции* v^*, *тогда* $F(v^*) = 0$, *и обратное, если* $F(v^*) = 0$, *тогда решение* x^* *задачи* (6.4) *при* $v = v^*$ *является оптимальным решением задачи* (6.3).

Лемма 6.2. *Если* (x, v, t) *является допустимым решением задачи* (6.5)–(6.7), *то* x *является допустимым решением задачи* (6.4). *Если* x *является допустимым решением задачи* (6.3), *то существуют такие* v *и* t, *что* (x, v, t) *является допустимым решением задачи* (6.5)–(6.7).

Лемма 6.3. *Точка x^* является оптимальным решением задачи (6.3) с оптимальным значением целевой функции v^* тогда и только тогда, когда (x^*, v^*, t^*) является оптимальным решением задачи (6.5)–(6.7) со значением целевой функции $t^* = 0$.*

Теорема 6.1. (Необходимые условия оптимальности). *Пусть x^* является оптимальным решением задачи (6.3) с оптимальным значением целевой функции v^*. Тогда существуют $t^* \in R$, $u^* \in R^m$, $z^* \in R^p$, такие что (x^*, v^*, u^*, z^*) удовлетворяют условиям:*

$$\nabla\left[\sum_{i=1}^{m} u_i^*(\varphi_i(x^*) - v^*\psi_i(x^*)) + \sum_{k=1}^{p} z_k^* h_k(x^*)\right] = 0, \tag{6.8}$$

$$u_i^*(\varphi_i(x^*) - v^*\psi_i(x^*)) = 0 \quad (i = \overline{1, m}), \tag{6.9}$$

$$z_k^* h_k(x^*) = 0 \quad (k = \overline{1, p}), \tag{6.10}$$

$$\varphi_i(x^*) - v^*\psi_i(x^*) \le 0 \quad (i = \overline{1, m}), \tag{6.11}$$

$$h_k(x^*) \le 0 \quad (k = \overline{1, p}), \tag{6.12}$$

$$\sum_{i=1}^{m} u_i^* = 1, \tag{6.13}$$

$$t^* = 0, \tag{6.14}$$

$$t^* \in R, \quad u^* \in R^m, \quad z^* \in R^p, \quad u^* \ge 0, \quad z^* \ge 0. \tag{6.15}$$

Теорема 6.2. (Достаточные условия). *Пусть (x^*, v^*, u^*, z^*) удовлетворяют условиям (6.8)–(6.15) и для переменных x функция*

$$A(x) = \sum_{i=1}^{m} u_i^*(\varphi_i(x) - v^*\psi_i(x)) + \sum_{k=1}^{p} z_k^* h_k(x)$$

является псевдовыпуклой для всех x, которые являются допустимыми решениями задачи (6.5)–(6.7). Тогда x^* является оптимальным решением задачи (6.3) с оптимальным значением целевой функции v^*.

Теорема 6.3. (Достаточные условия). *Пусть* (x^*, v^*, u^*, z^*) *удовлетворяют условиям* (6.8)–(6.15) *и для переменных* x *функция*

$$B(x) = \sum_{i=1}^{m} u_i^*(\varphi_i(x) - v^*\psi_i(x))$$

является псевдовыпуклой, а функция

$$C(x) = \sum_{i=1}^{m} z_k^* h_k(x)$$

является квазивыпуклой для всех x, *которые являются допустимыми решениями задачи* (6.5)–(6.7). *Тогда* x^* *является оптимальным решением задачи* (6.3) *с оптимальным значением целевой функции* v^*.

Приведенные выше утверждения позволяют с точностью до эквивалентности рассматривать для решения не исходную задачу обобщенного дробно-выпуклого программирования (6.1), (6.2), а ей эквивалентную задачу (6.5)–(6.7), которая считается более удобной и прозрачной для разработки эффективных алгоритмов. Данные утверждения характеризуют необходимые и достаточные условия оптимальности обобщенной задачи дробно-выпуклого программирования в классе выпуклой, псевдовыпуклой и квазивыпуклой оптимизации.

3. Параметрические методы. В работах [52, 53, 90, 153, 156] для решения задачи (6.1), (6.2) применяются различные варианты параметрического метода [171, 172, 260, 261] дробного программирования, который сводит исходную задачу к решению параметрической задачи (6.4).

А. *Обычный параметрический метод.* В соответствие с обычным параметрическим методом имеем следующий алгоритм решения задачи (6.1), (6.2).

Шаг 0. Полагаем $v = v_0$, чтобы $F(v_0) \geq 0$.

Шаг r. Пусть v_{r-1} – значение параметра v найденное на предыдущем шаге. Тогда:

- Находим решение x_r^* задачи выпуклого дискретного минимакса (6.4) при $v = v_{r-1}$. Для этого используются соответствующие алгоритмы и методы, приведенные в п. 6.1 для выпуклого случая. Одним из методов решения задачи (6.4) при фиксированном $v = v_{r-1}$ является ее сведение к эквивалентной задаче (6.5)–(6.7), т.е. к задаче

$$\min t,$$

$$\varphi_i(x) - v_{r-1}\psi_i(x) \leq t \quad (i = \overline{1,m}),$$

$$h_k(x) \leq 0 \quad (k = \overline{1,p}),$$

которая является задачей выпуклого программирования. Пусть (x_r^*, t_r^*) – оптимальное решение данной задачи.

- Находим $v_r = \max\limits_{i \in I} \dfrac{\varphi_i(x_r^*)}{\psi_i(x_r^*)} = \dfrac{\varphi_{i_r}(x^*)}{\psi_{i_r}(x^*)}$, где i_r – индекс для которого достигается максимальное отношение соответствующих значений функций.

- Если $t_r^* = 0$, то v_r является корнем уравнения $F(v) = 0$, т.е. если $|F(v_r)| \leq \delta$, где δ – достаточно малое положительное число, то x_r^* является оптимальным решением задачи (6.1), (6.2). Останов.

В результате применения параметрического метода исходная задача дискретного дробно-выпуклого минимакса сводится к решению последовательности выпуклых задач дискретного минимакса, т.е. к решению эквивалентных выпуклых задач с дополнительными ограничениями и дополнительной переменной.

Б. *Нормированный параметрический метод.* В работах [153, 156] предложено параметрическую задачу (6.4) рассматривать в эквивалентной (нормированной) форме

$$F_z(v) = \min_{x \in S} \max_{i \in I} \frac{\varphi_i(x) - v\psi_i(x)}{\psi_i(z)},$$

для любого $z \in S$.

В данном случае, нахождение корня уравнения $F_z(v) = 0$ связано с решением эквивалентной задачи (6.5)–(6.7) и получаем следующий алгоритм параметрического метода.

Шаг 0. Пусть имеем $x_0 \in S$ и находим $v_0 = \max\limits_{i \in I} \dfrac{\varphi_i(x^0)}{\psi_i(x^0)}$.

Шаг r. Пусть $x_{r-1} \in S$ – допустимое решение найденное на предыдущем шаге. Тогда:

- находим решение (x_r^*, t_r^*) модифицированной задачи (6.5)–(6.7) следующего вида

$$\min \ t,$$

$$\psi_i(x) - v_{r-1}\psi_i(x) - t\psi_i(x_{r-1}^*) \leq 0 \quad (i = \overline{1, m}),$$

$$h_k(x) \leq 0 \quad (k = \overline{1, p});$$

- полагаем $v_r = \max\limits_{i \in I} \dfrac{\varphi_i(x_r^*)}{\psi_i(x_r^*))}$;

- если $t_r^* = 0$, то v_r является корнем уравнения $F_z(v) = 0$, т.е. если $|F(v_r)| \leq \delta$, где δ – достаточно малое положительное число, то x_r^*

является оптимальным решением задачи (6.5)–(6.7) и соответственно исходной задачи (6.1), (6.2). Останов.

B. *Параметрический метод частичной линеаризации.* Другой алгоритм параметрического метода [62, 63] связан с понятием частичной линеаризации билинейно-выпуклой функцией

$$H_i(x,t) = \varphi_i(x) - t\psi_i(x) \quad (i = \overline{1,m}).$$

Пусть имеем некоторую допустимую точку (x_r, t_r) эквивалентной задачи (6.1), (6.2). Тогда рассмотрим частичную линеаризацию в точке (x_r, t_r) по переменной t в виде

$$H_i^r(x,t) = H_i(x,t_r) + (t - t_r)\nabla_t H_i(x_r,t_r) =$$
$$= \varphi_i(x) - t_r\psi_i(x) - (t - t_r)\psi_i(x_r).$$

Тогда на каждом шаге алгоритма параметрического метода решается задача

$$\min\ t,$$
$$\varphi_i(x) - t_r\psi_i(x) - (t - t_r)\psi_i(x_r) \le 0 \quad (i = \overline{1,m}),$$
$$h_k(x) \le 0 \quad (k = \overline{1,p}),$$

которая является задачей выпуклого программирования.

Пусть вместо исходной задачи (6.1), (6.2) имеем эквивалентную ей задачу

$$\min\ t,$$
$$\varphi_i(x) - t\psi_i(x) \le 0 \quad (i = \overline{1,m}),$$
$$h_k(x) \le 0 \quad (k = \overline{1,p}).$$

Для решения данной задачи используем параметрический метод и частичную линеаризацию.

Шаг 0. Пусть имеем $x_0 \in S$; полагаем $t_0 = \max\limits_{i\in I} \dfrac{\varphi_i(x_0)}{\psi_i(x_0)} =$ $= \min\{t:\ H_i^0(x_0,t) \le 0,\ \ i \in I\}$.

Шаг r. Пусть $x_{r-1} \in S$ – допустимое решение, найденное на предыдущем шаге и $t_{r-1} = \max\limits_{i\in I} \dfrac{\varphi_i(x_{r-1})}{\psi_i(x_{r-1})} = \min\{t:\ H_i^{r-1}(x_{r-1},t) \le 0,\ i \in I\}$.

Тогда:

- находим решение (x_r^*, t_r^*) задачи выпуклого программирования

$$\min\ t,$$
$$\varphi_i(x) - t_{r-1}\psi_i(x) - t\psi_i(x_{r-1}) \le -t_{r-1}\psi_i(x_{r-1}) \quad (i = \overline{1,m}),$$
$$h_k(x) \le 0 \quad (k = \overline{1,p});$$

- если $t_{r-1} = t_r^*$, тогда x_r^* является оптимальным решением задачи (6.1), (6.2); Останов, в противном случае

- полагаем $t_r = \max\limits_{i \in I} \dfrac{\varphi_i(x_r^*)}{\psi_i(x_r^*)}$ и выполняем очередной $(r+1)$-й шаг алгоритма.

Данный алгоритм, также как и параметрический метод, генерирует аналогичную последовательность точек (x_r, t_r), которая сходится к оптимальному решению задачи (6.5)–(6.7), а следовательно и к оптимальному решению исходной задачи (6.1), (6.2).

Таким образом, для решения задачи обобщенного дробно-выпуклого программирования можно использовать параметрические методы, которые сводят исходную задачу к решению последовательности таких же минимаксных выпуклых задач при исходных ограничениях, или же к последовательности задач выпуклого программирования, но с дополнительными ограничениями и дополнительным параметром (переменной).

4. Двойственные задачи. В обобщенном дробно-выпуклом программировании двойственная задача строится не для исходной задачи (6.3), а для эквивалентной ей задачи (6.5)–(6.7). Тогда для (6.5)–(6.7) имеем следующую двойственную задачу [46, 157]:

$$\sum_{i=1}^{m} u_i(\varphi_i(x) - v\psi_i(x)) + \sum_{k=1}^{p} z_k h_k(x) \to \max, \tag{6.16}$$

$$\nabla\left[\sum_{i=1}^{m} u_i(\varphi_i(x) - v\psi_i(x)) + \sum_{k=1}^{p} z_k h_k(x)\right] = 0, \tag{6.17}$$

$$\sum_{i=1}^{m} u_i = 1, \tag{6.18}$$

$$x \in R^n, \quad u \in R^m, \quad z \in R^p, \quad u \geq 0, \quad z \geq 0. \tag{6.19}$$

В ограничениях (6.17) вектор-градиент ∇ по переменным x содержит частные производные дифференцируемых функций $\varphi_i(x), \psi_i(x)$ и $h_k(x)$. Тогда ограничения (6.17) могут быть записаны в следующем виде

$$\sum_{i=1}^{m} u_i\left(\frac{\partial\varphi_i(x)}{\partial x_j} - v\frac{\partial\psi_i(x)}{\partial x_j}\right) + \sum_{k=1}^{p} z_k\frac{\partial h_k(x)}{\partial x_j} = 0 \quad (j = \overline{1,n}).$$

Другая формулировка двойственной задачи состоит в следующем:

$$\sum_{i=1}^{m} u_i(\varphi_i(x) - v\psi_i(x)) \to \max, \tag{6.20}$$

$$\nabla\left[\sum_{i=1}^{m} u_i(\varphi_i(x) - v\psi_i(x)) + \sum_{k=1}^{p} z_k h_k(x)\right] = 0, \tag{6.21}$$

$$\sum_{k=1}^{p} z_k h_k(x) \geq 0, \tag{6.22}$$

$$\sum_{i=1}^{m} u_i = 1, \tag{6.23}$$

$$x \in R^n, \quad u \in R^m, \quad z \in R^p, \quad u \geq 0, \quad z \geq 0. \tag{6.24}$$

Для задач (6.3) и (6.16)–(6.19) или (6.20)–(6.24) имеет место общая теория двойственности. Поэтому вместо исходной задачи (6.3) можно решать одну из задач (6.16)–(6.19) или (6.20)–(6.24), в которых также присутствуют дополнительные ограничения и дополнительные переменные. Двойственные задачи часто используются для анализа исходной задачи на оптимальность и разработки эффективных методов ее решения.

Для исходной задачи дискретного дробно-выпуклого программирования (6.3) может быть рассмотрена другая эквивалентная ей задача дробно-выпуклой (квазивыпуклой) оптимизации [40, 156, 367, 565]: найти

$$v^* = f(x^*) = \min_{x \in S} \max_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} = \min_{x \in S} \max_{u \in U} \frac{\varphi(x,u)}{\psi(x,u)} = \min_{x \in S} \max_{u \in U} \frac{\sum\limits_{i=1}^{m} u_i \varphi_i(x)}{\sum\limits_{i=1}^{m} u_i \psi_i(x)},$$

в которой

$$S = \{x\colon \ h_k(x) \leq 0, \quad k = \overline{1,p}\},$$

$$U = \Big\{u\colon \ \sum_{i=1}^{m} u_i = 1, \quad u_i \geq 0, \quad i = \overline{1,m}\Big\}.$$

Так как имеет место свойство минимакса

$$\min_{x\in S}\max_{u\in U}\frac{\sum\limits_{i=1}^{m}u_i\varphi_i(x)}{\sum\limits_{i=1}^{m}u_i\psi_i(x)}=\max_{u\in U}\min_{x\in S}\frac{\sum\limits_{i=1}^{m}u_i\varphi_i(x)}{\sum\limits_{i=1}^{m}u_i\psi_i(x)},$$

то определяем функцию

$$\Theta(u)=\min_{x\in S}\frac{\varphi(x,u)}{\psi(x,u)}=\min_{x\in S}\frac{\sum\limits_{i=1}^{m}u_i\varphi_i(x)}{\sum\limits_{i=1}^{m}u_i\psi_i(x)},$$

которая является квазивогнутой и недифференцируемой.

Рассмотрим задачу квазивогнутой оптимизации

$$\max_{u\in U}\ \Theta(u),$$

которую можно отнести к двойственной задаче обобщенного дробно-выпуклого программирования (6.3). Если u^* решение двойственной задачи, а x^* решение задачи (6.3), то имеет место равенство

$$v^*=f(x^*)=\Theta(u^*).$$

Если рассмотреть параметрическую задачу

$$F(u,v)=\min_{x\in S}\sum_{i=1}^{m}u_i(\varphi_i(x)-v\psi_i(x)),$$

то в соответствии с параметрическим методом для оптимальных решений x^* и u^* имеем $v^*=f(x^*)$, $F(u^*,v^*)=0$, или же $F(u^*,\Theta(u^*))=0$. Тогда вместо исходной задачи (6.3), решаем двойственную задачу $\max\limits_{u\in U}\Theta(u)$. Для ее решения применяем два метода: двойственный параметрический метод и субградиентный метод.

A. *Двойственный параметрический метод.* Для решения задачи (6.3) используем двойственный параметрический метод [40], который содержит следующие шаги.

Шаг 0. Пусть $u^0\in U$. Находим решение $x^*(u^0)$ задачи дробно-выпуклого программирования

$$v_0=\Theta(u^0)=\min_{x\in S}\frac{\sum\limits_{i=1}^{m}u_i^0\varphi_i(x)}{\sum\limits_{i=1}^{m}u_i^0\psi_i(x)},$$

для решения которой можно применить любые методы и алгоритмы дробно-выпуклого программирования, в частности параметрический метод для решения задачи выпуклого параметрического программирования вида

$$\min_{x\in S}\sum_{i=1}^{m}u_i^0\big(\varphi_i(x)-v\psi_i(x)\big).$$

Шаг r. Пусть на предыдущем, $(r-1)$-м шаге, получены следующие значения: u^{r-1} – значения переменных u, $x^*(u^{r-1})$ – значения переменных x и $v_{r-1}=\Theta(u^{r-1})$ – значения параметра v. Тогда на r-ом шаге:

- находим оптимальное решение u^r задачи

$$\max_{u\in U}\ F(u,\Theta(u^{k-1})),$$

т.е. решаем задачу линейного программирования

$$\sum_{i=1}^{m}b_iu_i\to\max,$$

$$\sum_{i=1}^{m}u_i=1,$$

$$u_i\geq 0\quad(i=\overline{1,m}),$$

в которой $b_i=\varphi_i(x^*(u^{r-1}))-\Theta(u^{r-1})\psi_i(x^*(u^{r-1}))$, $\quad i=\overline{1,m}$; решение данной задачи определяется по формуле

$$u_i^r=\begin{cases}1, & \text{для}\ \ i=i_r,\\ 0, & \text{для всех}\ \ i=\overline{1,m},\ i\neq i_r,\end{cases}$$

где i_r – значение индекса i для которого $b_{i_r}=\max\limits_{i=\overline{1,m}}b_i$;

- если $F(u^r,\Theta(u^{r-1}))=0$, то u^{r-1} является оптимальным решением двойственной задачи, а $x^*(u^{r-1})$ – оптимальное решение исходной задачи (6.3). Останов, в противном случае

- находим новое значение параметра v путем решения задачи дробно-выпуклого программирования

$$v_r=\Theta(u^r)=\min_{x\in S}\frac{\sum\limits_{i=1}^{m}u_i^r\varphi_i(x)}{\sum\limits_{i=1}^{m}u_i^r\psi_i(x)}=\min_{x\in S}\frac{\varphi_{i_r}(x)}{\psi_{i_r}(x)},$$

т.е. параметрическим методом при $v_{r-1} = \Theta(u^{r-1})$ находим решение $x^*(u^r)$ задачи выпуклого программирования

$$\min_{x\in S}\sum_{i=1}^{m} u_i^r\left(\varphi_i(x) - \Theta(u^{r-1})\psi_i(x)\right) = \min_{x\in S}\left(\varphi_{i_r}(x) - \Theta(u^{r-1})\psi_{i_r}(x)\right).$$

Так как $u_i^r = 1$ для $i = i_r$, для которого имеем $b_{i_r} = \max\limits_{i\in I} b_i$, а остальные $u_i^r = 0$ для $i = \overline{1,m}$, $\ i \neq i_r$, то данная задача выпуклого программирования решается только относительно функций $\varphi_{i_r}(x)$ и $\psi_{i_r}(x)$, т.е. получаем задачу дробно-выпуклого программирования

$$\Theta(u^r) = \min_{x\in S}\frac{\varphi_{i_r}(x)}{\psi_{i_r}(x)},$$

или задачу параметрического программирования

$$\min_{x\in S}\left[\varphi_{i_r}(x) - \Theta(u^{r-1})\psi_{i_r}(x)\right],$$

для которой находим решение $x^*(u^r)$, и естественно, находим

$$v_r = \Theta(u^r) = \frac{\varphi_{i_r}(x^*(u^r))}{\psi_{i_r}(x^*(u^r))}.$$

Если $v_{r-1} = v_r$, то имеет место $F(u^r, v_r) = 0$, т.е. v_r является корнем данного уравнения, а u^r – оптимальное решение двойственной задачи $\max\limits_{u\in U}\Theta(u)$.

В результате применения двойственного параметрического метода получаем итеративный процесс относительно параметра v_r. Полученная последовательность $\{v_r\}$ стремится к корню v^* уравнения $F(u^*, v^*) =$ $= 0$, т.е. имеем равенство

$$\sum_{i=1}^{m} u_i^*(\varphi_i(x^*) - v^*\psi_i(x^*)) = 0,$$

где u^* – оптимальное решение соответствующей линейной задачи, в которой $u_{i^*}^* = 1$, а $u_i^* = 0$ для всех $i = \overline{1,m}$, $\ i \neq i^*$, а i^* – значение индекса i, для которого имеем $b_{i^*} = \max\limits_{i\in I}\ b_i$;

x^* – оптимальное решение задачи выпуклого программирования

$$\min_{x\in S}\left(\varphi_{i^*}(x) - v^*\psi_{i^*}(x)\right).$$

Таким образом для решения u^* и v^* имеем равенства $v^* = f(x^*) =$
$= \dfrac{\varphi_{i^*}(x^*)}{\psi_{i^*}(x^*)} = \max\limits_{i\in I} \dfrac{\varphi_i(x^*)}{\psi_i(x^*)} = \dfrac{\sum\limits_{i=1}^{m} u_i^*\varphi_i(x^*)}{\sum\limits_{i=1}^{m} u_i^*\psi_i(x^*)} = \min\limits_{x\in S}\max\limits_{i\in I} \dfrac{\varphi_i(x)}{\psi_i(x)}$, т.е. x^* является оптимальным решением исходной задачи (6.3).

Б. *Субградиентный метод решения двойственной задачи.* Для решения двойственной задачи $\max\limits_{u\in U}\Theta(u)$ применяем субградиентный метод недифференцируемой оптимизации. Пусть имеем точку u^r, тогда на $(r+1)$-м шаге субградиентного метода необходимо выполнить следующие этапы.

1. При фиксированном $v_r = \Theta(u^r)$ решаем задачу выпуклого программирования

$$\min_{x\in S}\sum_{i=1}^{m} u_i^r(\varphi_i(x) - v_r\psi_i(x))$$

и находим ее оптимальное решение $x^*(u^r)$.

2. Находим значения субградиента параметрической функции $F(u, \Theta(u^r))$ в точке u^r по формуле

$$g_{F}(u^r) = \left\{g_{F}^i(u^r) = \bigl(\varphi_i(x^*(u^r)) - v_r\psi_i(x^*(u^r))\bigr),\quad i=\overline{1,m}\right\},$$

или же нормированного значения субградиента функции $\Theta(u)$ в той же точке u^r по формуле

$$g_{\Theta}(u^r) = \left\{g_{\Theta}^i(u^r) = \frac{\varphi_i(x^*(u^r)) - v_r\psi_i(x^*(u^r))}{\psi(x^*(u^r))},\quad i=\overline{1,m}\right\},$$

откуда имеем $g_{F}^i(u^r) = g_{\Theta}^i(u^r)\cdot\psi(x^*(u^r))$.

3. Находим новую точку u^{r+1} по формуле

$$u^{r+1} = \Pi_{U}(\widetilde{u}^{r+1}),\quad \text{где}\ \ \widetilde{u}_i^{r+1} = \max\left\{0,\ u_i^r + \gamma_{r+1}g_{F}^i(u^r)\right\},\quad i=\overline{1,m},$$

а $\Pi_{U}(\widetilde{u}^{r+1})$ – оператор проектирования точки $\widetilde{u}^{r+1}$ на линейное многообразие $U = \left\{u_i : \sum\limits_{i=1}^{m} u_i = 1,\quad u_i \geq 0,\quad i\in I\right\}$.

Пусть u^* – оптимальное решение задачи недифференцируемой оптимизации $\max\limits_{u\in U}\Theta(u)$, а $v^* = \Theta(u^*)$, тогда полученное решение x^* задачи выпуклого программирования

$$\min_{x\in S}\sum_{i=1}^{m} u_i^*(\varphi_i(x) - v^*\psi_i(x))$$

является оптимальным решением исходной задачи (6.3), притом

$$v^* = \Theta(u^*) = f(x^*), \quad \text{а} \quad F^*(u^*, v^*) = 0.$$

Другим, двойственным подходом, является построение функции Лагранжа и рассмотрение двойственных лагранжевых задач с учетом параметрической задачи. В дальнейшем применим такой подход для разработке методов решения задачи (6.5)–(6.7), и соответственно исходной задачи (6.1), (6.2), основанные на функциях Лагранжа и недифференцируемой оптимизации, параметрического метода и схемах декомпозиции по ограничениям [484, 633].

5. Субградиентные методы. Рассмотрим сначала аналогичные алгоритмы субградиентных методов для решения задач дискретного минимакса дробных функционалов. Пусть имеем дробную задачу минимакса (6.1), (6.2), в которой функции $\varphi_i(x)$, $\psi_i(x)$, $i \in I$ и $h_k(x)$ $(k = \overline{1,p})$, необязательно являются гладкими, т.е. непрерывными и везде дифференцируемыми. Тогда для решения задачи $\min\limits_{x \in S} f(x)$, в которой S может совпадать со всем пространством R^n или быть определенной ограничениями (6.2) могут быть использованы прямые субградиентные методы. Предложим следующие два способа решения обобщенной задачи дробного программирования (дробного минимакса), основанные на субградиентных методах и которые соответствуют предложенным методам решения общей задачи дробно-выпуклого программирования (см. п. 5.3.).

А. *Безусловная задача дробно-выпуклого минимакса.* Рассмотрим для начало задачу дробного минимакса на всем пространстве R^n, т.е. задачу

$$\min_{x \in R^n} \left[f(x) = \max_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} \right],$$

для которой $\varphi_i(x) \geq 0$, а $\psi_i(x) > 0$, $i \in I$ для любого $x \in R^n$. Для ее решения применяем субградиентный метод, с учетом параметрической задачи типа (6.4), которая имеет вид

$$F(v) = \min_{x \in R^n} \left[Z(x, v) = \max_{i \in I} \big(\varphi_i(x) - v\psi_i(x) \big) \right],$$

т.е. имеем параметрическую задачу дискретного минимакса.

Пусть имеем некоторую точку x_r, полученную на предыдущем шаге субградиентного метода. Тогда для соответствующей параметрической задачи дробного минимакса выполняем основные четыре этапа вычислений субградиентного метода (см. п. 5.3.).

1. Находим значение функции $f(x)$ в точке x_r по формуле

$$f(x_r) = \frac{\varphi_{i_r}(x_r)}{\psi_{i_r}(x_r)} = \max_{i \in I} \frac{\varphi_i(x_r)}{\psi_i(x_r)},$$

где i_r – индекс функций $\varphi_i(x)$ и $\psi_i(x)$ для которых достигается максимум их отношения в точке x_r.

2. Находим значение параметра v_r по формуле

$$v_r = f(x_r).$$

3. Находим значение субградиента функции $Z(x, v)$ в точке x_r, т.е. с учетом параметризации функции $f(x)$ по формуле

$$g_Z(x_r) = g_{\varphi_{i_r}}(x_r) - v_r g_{\psi_{i_r}}(x_r),$$

где $g_{\varphi_{i_r}}(x_r)$, $g_{\psi_{i_r}}(x_r)$ – значения субградиентов соответственно для функций $\varphi_{i_r}(x)$ и $\psi_{i_r}(x)$ в точке x_r.

4. Находим новую точку x_{r+1} по одной из формул перехода к новой точке в субградиентном методе

$$x_{r+1} = x_r - \gamma_{r+1} g_Z(x_r),$$

где γ_{r+1} – величина шага.

Пусть x^* является решение, полученное в результате вычислений по субградиентному методу. Тогда в этой точке множество субградиентов функции $f(x)$ содержит нулевой вектор, т.е. $0 \in g_Z(x^*)$. Откуда получаем, что

$$v^* = \frac{g_{\varphi_{i^*}}(x^*)}{g_{\psi_{i^*}}(x^*)} = f(x^*) = \frac{\varphi_{i^*}(x^*)}{\psi_{i^*}(x^*)}.$$

Тогда имеем $F(v^*) = \min\limits_{x \in R^n} \max\limits_{i \in I} \big(\varphi_i(x) - v^* \psi_i(x)\big) = \varphi_{i^*}(x^*) - v^* \psi_{i^*}(x^*) = = 0$, т.е. v^* является корнем уравнения $F(v) = 0$. Таким образом, решение x^*, найденное по субградиентному методу для задачи параметрического дискретного минимакса $\min\limits_{x \in R^n} Z(x, v)$, является оптимальным и для исходной задачи дробного дискретного минимакса $\min\limits_{x \in R^n} f(x)$.

Б. *Условная задача дробно-выпуклого минимакса.* Рассмотрим случай задачи (6.3), когда множество S определено ограничениями (6.2), т.е. $S = \{x \in R^n : h_k(x) \le 0, \ \ k = \overline{1,p}\}$, а $\varphi_i(x) \ge 0$, $\psi_i(x) > 0$, $\ i \in I$ для любого $x \in S$. Тогда для решения параметрической задачи обобщенного дробного программирования (6.4)

$$F(v) = \min_{x \in S} [Z(x, v) = \max_{i \in I} \big(\varphi_i(x) - v\psi_i(x)\big)],$$

применяем метод эллипсоидов (см. п. 5.3.), который в данном случае имеет следующие особенности при вычисления значений обобщенного градиента.

Пусть имеем точку x_r, найденную на r-м шаге метода эллипсоидов. Тогда значения обобщенного градиента функции $Z(x, v)$ в точке x_r определяются по формуле

$$g(x_r) = \begin{cases} g_Z(x_r) = g_{\varphi_{i_r}}(x_r) - v_r g_{\psi_{i_r}}(x_r), & \text{если } \ h_{k_r}(x_r) \leq 0; \\ g_h(x_r) = g_{h_{k_r}}(x_r), \ \ \text{если } \ h_{k_r}(x_r) > 0, \end{cases}$$

в которой:

k_r – индекс, для которого достигается поточечный максимум по функциям ограничений, а $h_{k_r}(x_r)$ – значение этого максимума в точке x_r, т.е. k_r – индекс, для которого $h_{k_r}(x_r) = \max\limits_{1 \leq k \leq p} h_k(x_r)$;

i_r – индекс, для которого достигается поточечный максимум по функциям целевого функционала, т.е. i_r – индекс, для которого

$$f(x_r) = \max_{i \in I} \frac{\varphi_i(x_r)}{\psi_i(x_r)} = \frac{\varphi_{i_r}(x_r)}{\psi_{i_r}(x_r)};$$

v_r – значение параметра v, который определяется в текущей точке x_r по формуле

$$v_r = \begin{cases} f(x_r) = \dfrac{\varphi_{i_r}(x_r)}{\psi_{i_r}(x_r)}, & \text{если } \ \varphi_{i_r}(x_r) \geq 0 \ \text{ и } \ \psi_{i_r}(x_r) > 0; \\ 0, & \text{в противном случае;} \end{cases}$$

$g_Z(x_r)$ – значение субградиента функции $Z(x, v)$ в точке x_r при фиксированном значении параметра $v = v_r$;

$g_h(x_r)$ – значение субградиента, определенное для одной из функций ограничений $h_k(x) \leq 0 \quad (k = \overline{1, p})$;

$g_{\varphi_{i_r}}(x_r)$, $g_{\psi_{i_r}}(x_r)$ и $g_{h_{k_r}}(x_r)$ – значения субградиентов в точке x_r соответственно для функций $\varphi_{i_r}(x)$, $\psi_{i_r}(x)$ и $h_{k_r}(x)$.

Полученное решение x^* по методу эллипсоидов для задачи $\min\limits_{x \in S} Z(x, v)$, при $v^* = f(x^*)$ будет оптимальным решением для исходной задачи (6.3). Действительно, так как x^* является оптимальным решением, то множество субградиентов функции $Z(x, v)$ содержит нулевой вектор, т.е. $0 \in g(x^*)$. Тогда, так как x^* – допустимое решение, то имеем что $0 \in g_Z(x_r)$, или же $g_{\varphi_{i^*}}(x^*) - v^* g_{\psi_{i^*}}(x^*) = 0$, где i^* – индекс, для которого достигается поточечный максимум отношения функций $\varphi_i(x)$ и $\psi_i(x)$ в точке x^* при $v = v^*$, т.е. для которого

$$f(x^*) = \frac{\varphi_{i^*}(x^*)}{\psi_{i^*}(x^*)} = \max_{i \in I} \frac{\varphi_i(x^*)}{\psi_i(x^*)}.$$

Откуда получим, что

$$v^* = \frac{g_{\varphi_{i^*}}(x^*)}{g_{\psi_{i^*}}(x^*)} = f(x^*) = \frac{\varphi_{i^*}(x^*)}{\psi_{i^*}(x^*)}.$$

Тогда имеем

$$F(v^*) = Z(x^*, v^*) = \varphi_{i^*}(x^*) - v^*\psi_{i^*}(x^*) = \varphi_i(x^*) - \frac{\varphi_{i^*}(x^*)}{\psi_{i^*}(x^*)}\psi_{i^*}(x^*) = 0,$$

т.е. v^* является корнем уравнения $F(v) = 0$, а x^* является оптимальным решением исходной задачи (6.3).

6. Функция Лагранжа и недифференцируемая оптимизация. Для задачи обобщенного дробно-выпуклого программирования функцию Лагранжа можно построить, как для исходной задачи (6.1), (6.2), так и для ее эквивалентных задач (6.5)–(6.7). Для каждой из них получаем задачу дробно-выпуклой или выпуклой безусловной оптимизации. Однако, в конечном итоге, получаем одинаковые схемы применения субградиентных методов решения исходной задачи (6.1), (6.2). Рассмотрим два варианта построения функции Лагранжа и применения методов недифференцируемой оптимизации.

A. *Функция Лагранжа задачи* (6.1), (6.2). Предлагается использовать для задачи обобщенного дробно-выпуклого программирования (6.1), (6.2) следующую дробную функцию Лагранжа

$$L(x, u, z) = \frac{\sum\limits_{i=1}^{m} u_i\varphi_i(x) + \sum\limits_{k=1}^{p} z_k h_k(x)}{\sum\limits_{i=1}^{m} u_i\psi_i(x)},$$

$$\sum_{i=1}^{m} u_i = 1,$$

$$u_i \geq 0 \quad (i = \overline{1, m}); \qquad z_k \geq 0 \quad (k = \overline{1, p}),$$

в которой $z = (z_1, z_2, \ldots, z_p)$ является вектором множителей Лагранжа (двойственные переменные) для ограничений (6.2), а $u = (u_1, u_2, \ldots$ $\ldots, u_m)$ представляет собой вектор параметров свертывания функций, входящие в целевой функционал (6.1).

Для получения безусловной функции Лагранжа приведем еще одну операцию релаксации для ограничения $\sum\limits_{i=1}^{m} u_i = 1$. Тогда, введя дополнительную двойственную переменную t, получим следующую функцию Лагранжа

$$L(x, u, z, t) = \frac{\sum\limits_{i=1}^{m} u_i \varphi_i(x) + \left(1 - \sum\limits_{i=1}^{m} u_i\right) \cdot t + \sum\limits_{k=1}^{p} z_k h_k(x)}{\sum\limits_{i=1}^{m} u_i \psi_i(x)}.$$

Решение задачи (6.1), (6.2) сводится к нахождению седловой точки (x^*, t^*, u^*, z^*) функции Лагранжа $L(x, t, u, z)$, которая соответствует задачам:

$$L(x^*, t^*, u^*, z^*) = \max_{u,z} \min_{x,t} L(x, t, u, z) = \min_{x,t} \max_{u,z} L(x, t, u, z).$$

С учетом их разделения на подзадачи по уровням анализа имеем следующие две задачи:

- задачи внутреннего уровня

$$L^*(u, z) = \min_{x,t} L(x, t, u, z),$$

которая представляет собой задачу дробно-выпуклого программирования по переменным x и t;

- задачи внешнего уровня

$$\max_{u,z} L^*(u, z),$$

которая является двойственной задачей для (6.1), (6.2).

Так как по определению, функция $L^*(u, z)$ является кусочно-линейной, вогнутой и не везде дифференцируемой по переменным u и z, то для ее максимизации используются методы недифференцируемой оптимизации. Если для этого применить один из субградиентных методов, то на каждом его шаге при фиксированных значениях переменных u и z необходимо решать задачу дробно-выпуклого программирования $\min\limits_{x,t} L(x, t, u, z)$.

Пусть для решения задачи $\max\limits_{u,z} L^*(u, z)$ используется один из субградиентных методов, на r-м шаге которого имеем текущую точку (u^r, z^r). Тогда на очередном $(r+1)$-м шаге необходимо выполнить основные три этапа вычислений.

1. Решить задачу $\min\limits_{x,t} L(x, t, u, z)$ при фиксированных $u_i = u_i^r$ $(i = \overline{1, m})$ и $z_k = z_k^r$ $(k = \overline{1, p})$, для которой функционал по переменным x и t является дробно-выпуклым и имеет вид:

$$F_r(x,t) = \frac{\sum\limits_{i=1}^{m} u_i^r \varphi_i(x) + \Bigl(1 - \sum\limits_{i=1}^{m} u_i^r\Bigr) t + \sum\limits_{k=1}^{p} z_k^r h_k(x)}{\sum\limits_{i=1}^{m} u_i^r \psi_i(x)}.$$

Эта задача является задачей безусловной минимизации дробно-выпуклого (квазивыпуклого или псевдовыпуклого) функционала $F(x,t)$, для чего можно использовать различные алгоритмы и методы, например, параметрический метод. Для этого рассматривается параметрическая задача $\lambda(v) = \min\limits_{x,t} Q_r(x,t,v)$, в которой

$$Q_r(x,t,v) = \sum_{i=1}^{m} u_i^r(\varphi_i(x) - v\psi_i(x)) + \sum_{k=1}^{p} z_k^r h_k(x) + \Bigl(1 - \sum_{i=1}^{m} u_i^r\Bigr) t$$

и находится корень $v(u^r, t^r)$ уравнения $\lambda(v) = 0$.

Таким образом, вместо задачи оптимизации дробно-выпуклой функции $\min\limits_{x,t} F_r(x,t)$ решаем задачу оптимизации выпуклой функции $\min\limits_{x,t} Q_r(x,t,v)$ при фиксированном значении параметра $v = v_{r-1}$.

Для этого определяем значение параметра

$$v_{r-1} = f(x_{r-1}^*) = \max_{i \in I} \frac{\varphi_i(x_{r-1}^*)}{\psi_i(x_{r-1}^*)},$$

где x_{r-1}^* – найденное решение задачи $\min\limits_{x,t} Q_r(x,t,v_{r-1})$ на предыдущем шаге субградиентного метода.

Находим решение задачи выпуклой оптимизации

$$\min_{x,t} Q_r(x,t,v_{r-1}).$$

Если фиксировать $t = 0$, то имеем задачу минимизации функции $A(x)$, которая по теореме 6.3 является псевдовыпуклой.

Пусть $x_r^* = x(u^r, z^r)$ является решение данной задачи.

2. Находим значения обобщенных градиентов функции $L^*(u,z)$ по переменным u и z в точке (u^r, z^r) с учетом полученного решения x_r^* и функции $Q_r(x,t,v_{r-1})$ по формулам

$$g_{L^*}^{u_i} = g_{Q_r}^{u_i} = \varphi_i(x_r^*) - v_r \psi_i(x_r^*) \quad (i = \overline{1,m}),$$

$$g_{L^*}^{z_k} = g_{Q_r} = h_k(x_r^*) \quad (k = \overline{1,p}).$$

3. Находим значения новой точки (u^{r+1}, z^{r+1}) по формулам

$$u_i^{r+1} = \max\{0,\ u_i^r + \gamma_{r+1} g_{Q_r}^{u_i}\} \quad (i = \overline{1,m}),$$

$$z_k^{r+1} = \max\{0,\ z_k^r + \gamma_{r+1} g_{Q_r}^{z_k}\} \quad (k = \overline{1,p}),$$

где γ_{r+1} – величина шага в субградиентном методе.

Пусть u^* и z^* – решение задачи $\max\limits_{u,z} L^*(u,z)$, а x^* и $t^* = 0$ – решение задач $\min\limits_{x,t} L(x,t,u^*,z^*)$, $\min\limits_{x,t} F(x,t)$ или же $\min\limits_{x,t} Q(x,t,v)$. Тогда $v^* = f(x^*) = \max\limits_{i\in I} \dfrac{\varphi_i(x)}{\psi_i(x)}$, т.е. x^* – оптимальное решение исходной задачи (6.3).

Б. *Функция Лагранжа для задачи* (6.5)–(6.7). Для задачи (6.5)–(6.7), и соответственно для исходной задачи (6.3), можно рассматривать функцию Лагранжа, которая в общем имеет вид:

$$L(x,v,t,u,z) = \sum_{i=1}^{m} u_i\big(\varphi_i(x) - v\,\psi_i(x)\big) + \left(1 - \sum_{i=1}^{m} u_i\right)t + \sum_{k=1}^{p} z_k h_k(x),$$

где $u = (u_1, u_2, \ldots, u_m) \in R^m$, $z = (z_1, z_2, \ldots, z_p) \in R^p$ являются множителями Лагранжа (двойственные переменные), соответствующие ограничениям (6.6) и (6.7). Тогда решение задачи (6.5)–(6.7) сводится к нахождению седловой точки $(x^*, v^*, t^*, u^*, z^*)$ функции Лагранжа $L(x,v,t,u,z)$, которая соответствует задачам:

$$L(x^*, v^*, t^*, u^*, z^*) = \max_{u,z} \min_{x,v,t} L(x,v,t,u,z) = \min_{x,v,t} \max_{u,z} L(x,v,t,u,z).$$

С учетом их разделения на подзадачи по уровням анализа имеем следующие две задачи:

- задачу внутреннего уровня

$$L^*(u,z) = \min_{x,v,t} L(x,v,t,u,z);$$

- задачу внешнего уровня

$$\max_{u,z} L^*(u,z).$$

Так как по определению функция $L^*(u,z)$ является кусочно-линейной, вогнутой и не везде дифференцируемой по переменных u и z, то для ее максимизации используются методы недифференцируемой оптимизации. Если для этого применить один из субградиентных методов, то на каждом его шаге, при фиксированных значениях переменных u и z необходимо решить задачу $\min\limits_{x,v,t} L(x,v,t,u,z)$ и найти значения обобщенных градиентов g_L^u и g_L^z по общим формулам, которые соответствуют выражениям левых частей для ограничений (6.6) для переменных u

$$g_{L^*}^{u_i} = \varphi_i(x) - v\,\psi_i(x) - t \quad (i = \overline{1,m})$$

и ограничений (6.7) для переменных z

$$g_{L^*}^{z_k} = h_k(x) \quad (k = \overline{1,p}).$$

Значения обобщенных градиентов вычисляются в текущей точке (u^r, z^r) r-го шага субградиентного метода с учетом полученного решения $\big(x(u^r,z^r),\, v(u^r,z^r),\, t(u^r,z^r)\big)$ задачи $\min\limits_{x,v,t} L(x,v,t,u^r,z^r)$, которая является задачей безусловной оптимизации выпуклого функционала при фиксированных параметров v и t.

Таким образом исходная задача обобщенного дробно-выпуклого программирования (6.1), (6.2) сводится к решению задачи безусловной недифференцируемой оптимизации $\max\limits_{u,z} L^*(u,z)$ субградиентным методом, на каждом шаге которого при фиксированных параметрах $\overline{v}$ и $\overline{t}$ решается задача безусловной минимизации функции $A(x) = \sum\limits_{i=1}^{m} u_i^r(\varphi_i(x) - \overline{v}\psi_i(x)) + \sum\limits_{k=1}^{p} z_k^r h_k(x)$, которая, в соответствии с теоремой 6.2 является псевдовыпуклой.

Замечание. *В предложенном методе нахождения седловой точки функции Лагранжа, при решении задачи* $\min\limits_{x,v,t} L(x,v,t,u^r,z^r)$ *значения параметров* v *и* t *интуитивно можно фиксировать следующим образом:* $\overline{t} = 0$, *а* $\overline{v} = f(x^*(u^{r-1}, z^{r-1}))$, *где* $x^*(u^{r-1}, z^{r-1})$ *полученное оптимальное решение задачи минимизации функции* $A(x)$ *на предыдущем шаге субградиентного метода. Такой прием фиксирования параметров* v *и* t *позволит через значения переменных* $x(u^r, z^r)$ *приблизиться к значениям функционалов* (6.3) *и* (6.5) *в оптимальной точке* x^* (*см. леммы* 6.1–6.3).

7. Параметрический метод и схема декомпозиции по ограничениям. В нелинейном параметрическом методе [52, 53, 90], [153]–[156], [189, 192] задача (6.1), (6.2) сводится к решению параметрической задачи выпуклого программирования (6.4).

Алгоритм нелинейного параметрического метода заключается в следующем.

Предварительный 0-й шаг. Берем некоторую точку $x^0 \in S$. Полагаем $v_1 = f(x^0)$ и $r = 1$.

Общий r-й шаг. Пусть $x^{r-1} \in S$ – решение задачи (6.4), найденное на предыдущем шаге при фиксированном значении параметра v. Тогда:

- полагаем $v_r = f(x^{r-1})$;
- находим оптимальное решение x^r задачи (6.4) при фиксированном значении параметра $v = v_r$, которая является задачей дискретного минимакса выпуклой функции при исходных выпуклых ограничениях;
- если $F(v_r) = 0$, то x^r – оптимальное решение задачи (6.1), (6.2), в противном случае заменяем $r - 1$ на r и повторяем общий шаг.

Для нахождения первоначальной допустимой точки $x^0 \in S$ достаточно решить задачу выпуклого программирования минимизации любой из функций $\varphi_i(x)$ на S.

Следует отметить, что на каждом шаге нелинейного параметрического метода необходимо решить аналогичную по сложности задачу выпуклого программирования (задача (6.4)), при дополнительных ограничениях и переменных. Как уже отмечалось выше, для решения задачи (6.4) в свою очередь применяются итеративные алгоритмы, на каждом шаге которых необходимо решить одну или две вспомогательные задачи оптимизации.

Одним из самых простых приемов решения задачи (6.4) является ее сведение к эквивалентной задаче:

$$t \to \min; \tag{6.25}$$

$$\varphi_i(x) - v_r\psi_i(x) \le t, \quad i \in I; \tag{6.26}$$

$$h_k(x) \le 0 \quad (k = \overline{1,p}); \tag{6.27}$$

$$t \ge 0. \tag{6.28}$$

При фиксированном значением параметра v_r задача (6.25)–(6.28) является задачей выпуклого программирования и она может быть решена общеизвестными методами.

Однако, если исходная задача (6.1), (6.2) имеет специальную структуру ограничений, то прибавление ограничений (6.26) не способствует применению специальных методов ее решения. В таких целях используют схему декомпозиции по ограничениям и субградиентные методы.

Тогда для задачи (6.25)–(6.28) предложенный подход состоит в следующем [484, 633].

На множестве ограничений (6.26) строится функция Лагранжа

$$L(x,t,u) = t + \sum_{i=1}^{m} u_i(\varphi_i(x) - v_r\psi_i(x) - t)$$

и рассматривается задача

$$\max_u L^*(u), \tag{6.29}$$

где

$$L^*(u) = \min_t \min_{x\in S} L(x,t,u). \tag{6.30}$$

Отличие от приведенной выше схемы решения задачи (6.5)–(6.7) заключается в том, что функция Лагранжа строится только для части ограничений. В данном подходе ограничения (6.26) учитываются в задаче внешнего уровня, а ограничения (6.27) – в задаче внутреннего уровня.

Для решения задачи (6.29) применяется субградиентный метод максимизации функции $L^*(u)$ по переменным u, а для нахождения обобщенного градиента функции $L^*(u)$ в некоторой фиксированной точке $u = \overline{u}$ решается задача (6.30), для которой система ограничений совпадает с исходными ограничениями задачи (6.1), (6.2) и имеет вид

$$\sum_{i=1}^{m} \overline{u}_i(\varphi_i(x) - v_r\psi_i(x)) + (1 - \sum_{i=1}^{m} \overline{u}_i)t \to \min,$$

$$h_k(x) \le 0 \quad (k = \overline{1,p}).$$

С учетом независимости значений переменной t от системы ограничений, окончательный вид подзадачи определения значений субградиента будет следующим:

$$\min_{x\in S} \sum_{i=1}^{m} \overline{u}_i(\varphi_i(x) - v_r\psi_i(x)). \tag{6.31}$$

Функция цели в задачи (6.31) соответствует функцией $B(x)$, которая по теореме 6.3 является псевдовыпуклой для всех $x \in S$. В итоге, при решении параметрическим методом задачи (6.1), (6.2) специальной структуры получим двухуровневый алгоритм, на каждом уровне которого соответствующие им задачи решаются итерационными алгоритмами.

Таким образом, при решении задачи (6.1), (6.2) параметрическим методом итерационный процесс организуется по переменной v и для каждого $v = v_r,\ r = 0, 1, 2, \ldots$ (внешний итерационный цикл) решается задача (6.29) субградиентным методом и по переменным u организуется внутренний итерационный цикл и находятся $u = u^l,\ l = 0, 1, 2, \ldots$. В результате применения предложенного подхода к решению обобщенной задачи дробно-выпуклого программирования приходим к двухуровневому процессу оптимизации, внутри которого решается задача выпуклого программирования (6.31) при исходных ограничениях (6.2). Таким образом задача (6.31) будет той же выпуклости и при тех же ограничениях, как и исходная задача (6.1), (6.2), т.е. выпуклого или линейного программирования, транспортного типа, или блочно-диагональной структуры.

8. Схема декомпозиции по ограничениям. В дальнейшем предложим алгоритм решения задачи (6.1), (6.2) (для случая, когда ограничения задачи имеют специальную структуру), который будет иметь только один итерационный процесс (относительно переменных u) [469, 470, 715].

Предложенный алгоритм основан на функции Лагранжа, схеме декомпозиции по ограничениям и субградиентных методах [484, 633].

Для задачи (6.1), (6.2) рассмотрим эквивалентную ей задачу:

$$t \to \min; \tag{6.32}$$

$$\varphi_i(x) - t\psi_i(x) \le 0, \quad i \in I; \tag{6.33}$$

$$h_k(x) \le 0 \quad (k = \overline{1, p}); \tag{6.34}$$

$$t \ge 0. \tag{6.35}$$

Имеют место утверждения [484, 633].

Теорема 6.4. *Если x^* – оптимальное решение задачи* (6.1), (6.2), *то существует такое t^* $(t^* = f(x^*))$, что (x^*, t^*) является оптимальным решением задачи* (6.32)–(6.35).

Доказательство. Пусть x^* – оптимальное решение задачи (6.1), (6.2). Тогда имеем

$$f(x^*) = \max_{i \in I} \frac{\varphi_i(x^*)}{\psi_i(x^*)} \le f(x) = \max_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)}$$

для любого $x \in S$. Одновременно с этим для любого $i \in I$ имеют место неравенства

$$\frac{\varphi_i(x^*)}{\psi_i(x^*)} \leq \max_{i \in I} \frac{\varphi_i(x^*)}{\psi_i(x^*)} = f(x^*) = t^*.$$

Тогда имеем $\varphi_i(x^*) - t^*\psi_i(x^*) \leq 0, \quad i \in I$, т.е. (x^*, t^*) является допустимым решением задачи (6.32)–(6.35).

Покажем теперь, что (x^*, t^*) является оптимальным решением задачи (6.32)–(6.35). Допустим обратное, что существует такой $\bar{t}$, для которого $t^* > \bar{t}$. Тогда $(x^*, \bar{t})$ является допустимым решением задачи (6.32)–(6.35), и из ограничения (6.33) следует, что

$$\frac{\varphi_i(x^*)}{\psi_i(x^*)} \leq \bar{t} < t^* = \max_{i \in I} \frac{\varphi_i(x^*)}{\psi_i(x^*)},$$

для любого $i \in I$. Последнее строгое неравенство противоречит оптимальности решения x^* задачи (6.1), (6.2).

Таким образом (x^*, t^*) является оптимальным решением задачи (6.32)–(6.35). Теорема доказана.

Теорема 6.5. *Если* (x^*, t^*) *– оптимальное решение задачи* (6.32)–(6.35)*, то* x^* *является оптимальным решением задачи* (6.1), (6.2).

Доказательство. Пусть (x^*, t^*) – оптимальное решение задачи (6.32)–(6.35). Тогда x^* удовлетворяет условиям (6.34), т.е. является допустимым решением задачи (6.1), (6.2). Так как t^* является оптимальным значением функционала (6.32), а x^* удовлетворяет ограничениям (6.33), то имеем $t^* \leq t$ для любого t, удовлетворяющего ограничениям (6.33) при любом значении $x \in S$. Тогда получим неравенства

$$\frac{\varphi_i(x^*)}{\psi_i(x^*)} \leq t^*, i \in I \text{ и } \max_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} = t^* \leq t = \max_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)},$$

для любого $x \in S$.

Таким образом, имеют место неравенства:

$$\max_{i \in I} \frac{\varphi_i(x^*)}{\psi_i(x^*)} = t^* \leq t = \max_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)}$$

для любого $x \in S$, т.е. получим неравенства

$$\max_{i \in I} \frac{\varphi_i(x^*)}{\psi_i(x^*)} \leq \max_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} \quad \text{или же} \quad f(x^*) \leq f(x).$$

Следовательно, x^* является оптимальным решением задачи (6.1), (6.2). Теорема доказана.

Заметим, что критерий оптимальности для задачи (6.32)–(6.35) может быть сформулирован следующим образом: найти те значения переменных $x \in S$, для которых переменная t, удовлетворяющая ограничениям (6.33) принимает минимальное значение. Другими словами, необходимо найти значение x^* переменных $x \in S$, для которых определенное по формуле

$$t^* = \max_{i \in I} \frac{\varphi_i(x^*)}{\psi_i(x^*)}$$

значение переменной t будет минимальным.

Построим функцию Лагранжа для задачи (6.32), (6.33) по формуле

$$L(x, t, u) = t + \sum_{i \in I} u_i(\varphi_i(x) - t\psi_i(x)),$$

где $u = \{u_i, \quad i \in I\}$ – множители Лагранжа для ограничений (6.33), $u_i \geq 0, \quad i \in I$.

Тогда задача (6.32)–(6.35), и тем самым и (6.1), (6.2) сводится к решению задачи нахождения седловой точки функции Лагранжа $L(x, t, u)$, т.е. к решению задачи

$$L(x^*, t^*, u^*) = \max_{u \geq 0} \min_{t \geq 0} \min_{x \in S} L(x, t, u). \tag{6.36}$$

В свою очередь задачу (6.36) можно свести к решению двойственной по Лагранжу задачи

$$\max_{u \geq 0} L^*(u), \tag{6.37}$$

где

$$L^*(u) = \min_{t \geq 0} \min_{x \in S} L(x, t, u). \tag{6.38}$$

Функция $L^*(u)$ определена для любого $u \geq 0$ и является кусочно-линейной, вогнутой и недифференцируемой. Поэтому для решения задачи (6.37) используются субградиентные методы, на каждом шаге которых необходимо решить задачу (6.38) при фиксированных значениях переменных u.

Пусть для решения задачи (6.37) используется некоторый субградиентный метод, на r-м шаге которого имеем некоторую точку u^r. Тогда на $(r+1)$-м $\quad (r = 0, 1, 2, \dots)$ шаге необходимо выполнить следующие три основных этапа.

1. Решить задачу (6.38) при фиксированных $u = u^r$, т.е. найти оптимальное решение $(x^*(u^r), t^*(u^r))$ задачи

$$\min_{t\geq 0,\, x\in S} \Big[t + \sum_{i\in I} u_i^r \big(\varphi_i(x) - t\psi_i(x)\big)\Big].$$

2. Вычислить обобщенный градиент функции $L^*(u)$ в точке $u = u^r$ по формуле

$$g_i(u^r) = \varphi_i(x^*(u^r)) - t^*(u^r)\psi_i(x^*(u^r)), \quad i \in I.$$

3. Найти новые значения u^{r+1} по формуле

$$u_i^{r+1} = \max\{0, u_i^r + \gamma_{r+1} g_i(u^r)\}, \quad i \in I,$$

где γ_{r+1} – величина шага в субградиентном методе.

Теорема 6.6. *Пусть* u^* *– оптимальное решение задачи* (6.37), *а* (x^*, t^*) *– оптимальное решение задачи* (6.38) *при фиксированных* $u = u^*$. *Тогда* (x^*, t^*) *– оптимальное решение задачи* (6.32)–(6.35), *а* x^* *– оптимальное решение задачи* (6.1), (6.2).

Доказательство. Так как при $u = u^*$ точка (x^*, t^*) является оптимальным решением задачи выпуклого программирования (6.38), то в соответствии с теоремой Куна-Таккера имеем:

1) $x^* \in S$ и тем самым, x^* удовлетворяет ограничениям (6.34) и (6.2);

2) $t^* + \sum_{i\in I} u_i^*(\varphi_i(x^*) - t^*\psi_i(x^*)) \leq t + \sum_{i\in I} u_i^*(\varphi_i(x) - t\psi_i(x))$ для любого $x \in S$;

3) $\varphi_i(x^*) - t^*\psi_i(x^*)) \leq 0, \quad i \in I$, т.е. решение (x^*, t^*) является допустимым для задачи (6.32)–(6.35);

4) $u_i^*(\varphi_i(x^*) - t^*\psi_i(x^*)) = 0, \quad i \in I.$

Тогда из 2) и 4) получим

$$t^* - t \leq \sum_{i\in I} u_i^*(\varphi_i(x) - t\psi_i(x)).$$

Так как $u_i^* \geq 0$ для всех $i \in I$, то для любых x и t, удовлетворяющих ограничениям (6.33) и (6.34) имеем $\sum_{i\in I} u_i^*\big(\varphi_i(x) - t\psi_i(x)\big) \leq 0$. Таким образом, имеем $t^* \leq t$, т.е. решение (x^*, t^*) является оптимальным для задачи (6.32)–(6.35), и одновременно с этим, x^* является оптимальным решением задачи (6.1), (6.2). Теорема доказана.

Рассмотрим теперь задачу (6.38), т.е. при фиксированных значениях двойственных переменных $u = \overline{u}$ необходимо решить следующую задачу:

$$t + \sum_{i \in I} \overline{u}_i(\varphi_i(x) - t\,\psi_i(x)) \to \min, \tag{6.39}$$

$$h_k(x) \le 0 \quad (k = \overline{1,p}), \tag{6.40}$$

$$t \ge 0. \tag{6.41}$$

При фиксированных $t \ge 0$ задача (6.39)–(6.41) является задачей выпуклого программирования. Тогда решение задачи (6.39)–(6.41) сводится к нахождению такого решения $x^* \in S$, которое в паре с некоторым значением $t^* \ge 0$ дает минимальное значение функционала (6.39). Для решения задачи (6.39)–(6.41) может быть использован некоторый приближенный алгоритм, который при оптимальных значениях двойственных переменных $u = u^*$ задачи (6.37) обеспечил бы получение оптимального решения задачи (6.39)–(6.41). Такой алгоритм может быть построен, если при очередном решении задачи (6.39)–(6.41) фиксировать значение параметра t по формуле

$$t = t^*(x^*(u^{r-1})) = f(x^*(u^{r-1})) = \max_{i \in I} \frac{\varphi_i(x^*(u^{r-1}))}{\psi_i(x^*(u^{r-1}))},$$

где $x^*(u^{r-1}) \in S$ – решение задачи (6.38), найденное на предыдущем шаге субградиентного метода.

Пусть перед выполнением очередного r-го шага субградиентного метода имеем значения переменных $u = u^r$ и предыдущее решение $(x^*(u^{r-1}), t^*(x^*(u^{r-1}))$ задачи (6.39)–(6.41) . Тогда для решения задачи (6.39)–(6.41) на r-м шаге субградиентного метода используем следующие три основных этапа.

1. Фиксируем значения переменной t' по формуле

$$t' = t^*(x^*(u^{r-1})).$$

2. Находим оптимальное решение $x^*(u^r)$ задачи выпуклого программирования

$$\sum_{i \in I} u_i^r(\varphi_i(x) - t'\psi_i(x)) \to \min, \tag{6.42}$$

$$h_k(x) \le 0 \quad (k = \overline{1,p}). \tag{6.43}$$

3. Находим новое значение параметра t по формуле

$$t^*(x^*(u^r)) = \max_{i \in I} \frac{\varphi_i(x^*(u^r))}{\psi_i(x^*(u^r))}.$$

Следует отметить, что задача (6.42), (6.43) решается на каждом шаге субградиентного метода с незначительными изменениями коэффициентов u_i^r и t'. Поэтому в методе решения задачи выпуклого программирования (например, метод внутренней точки) за исходную точку на r-м шаге субградиентного метода следует выбрать полученное на предыдущем шаге решение $x^*(u^{r-1})$ задачи (6.42), (6.43).

Пусть на последнем r-м шаге субградиентного метода $u^* = u^r$ является оптимальным решением задачи (6.37). Находим x^*– оптимальное решение задачи (6.42), (6.43) при фиксированном $t' = f(x^*(u^r))$ и полагаем $t^* = f(x^*)$. Тогда (x^*, t^*) будет оптимальным решением задачи (6.32)–(6.35), а x^* – оптимальным решением задачи (6.1), (6.2).

9. Задача обобщенного дробно-линейного программирования. Рассмотрим теперь случай, когда функции

$$\{\varphi_i(x),\, i \in I\};\ \{\psi_i(x),\, i \in I\}\ \text{ и }\ \{h_k(x)\quad (k = \overline{1,p})\}$$

являются линейными. Тогда получим следующую обобщенную дробно-линейную задачу:

$$\min_{x \in S} \left[f(x) = \max_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} \right], \tag{6.44}$$

в которой

$$x = (x_1, x_2, \ldots, x_n) \in R^n, \quad I = \{1, 2, \ldots, m\},$$

$$\varphi_i(x) = \sum_{j=1}^{n} c_{ij} x_j + c_i^0, \quad \psi_i(x) = \sum_{j=1}^{n} d_{ij} x_j + d_i^0,$$

а $S = \left\{ x : \sum_{j=1}^{n} a_{kj} x_j \le b_k \quad (k = \overline{1,p}); \quad x_j \ge 0 \quad (j = \overline{1,n}) \right\}$ является выпуклым многогранным множеством.

Для решения задачи обобщенного дробно-выпуклого программирования (6.44) могут быть использованы различные модификации параметрического метода, приведенные в работах [52, 53, 62, 63, 90, 153, 156] для решения обобщенных дробно-выпуклых задач. В работе [395] предложен метод внутренних точек, в [407] – приближенный алгоритм, а в [144] задача (6.44) рассматривается как задача теории игр. Двойственные задачи для (6.44) рассмотрены в работе [157], которые построены на основе параметрического анализа.

Задача (6.44) является задачей обобщенного дробно-линейного программирования в виде задачи дискретного минимакса с дробно-линейными функциями и линейными ограничениями. Она может быть записана в эквивалентной форме в виде задачи билинейного программирования следующего вида

$$t \to \min,$$
$$\varphi_i(x) - t\,\psi_i(x) \le 0, \quad i \in I,$$
$$x \in S,$$

в которой имеются билинейные ограничения, содержащие произведение $t \cdot x_j$ переменных t и x_j.

Предположим, что $\varphi_i(x) \ge 0$, а $\psi_i(x) > 0, \quad i \in I$ для любого $x \in S$. Тогда функция $f(x)$ является квазивыпуклой и недифференцируемой на S. Для решения задачи (6.44) или для ее эквивалентной задачи билинейного программирования могут быть использованы рассмотренные выше методы для нелинейного случая, а именно субградиентные методы решения прямой и двойственной задач, параметрический прямой и двойственной метод, алгоритм частичной линеаризации, алгоритмы субградиентного метода и схемы декомпозиции по ограничениям, в которых учитывается линейность функций $\varphi_i(x), \psi_i(x)$ и $h_k(x)$. В данном случае внутренние задачи в итеративных методах будут задачами линейного программирования.

А. *Двойственная задача.* В качестве двойственной задачи обобщенного дробно-линейного программирования (6.44) может быть использована следующая задача дробного билинейного программирования

$$v^* = f(x^*) = \min_{x\in S}\max_{i\in I}\frac{\sum\limits_{j=1}^{n} c_{ij}x_j + c_i^0}{\sum\limits_{j=1}^{n} d_{ij}x_j + d_i^0} = \max_{u\in U}\min_{x\in S}\frac{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} c_{ij}x_j u_i + \sum\limits_{i=1}^{m} c_i^0 u_i}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} d_{ij}x_j u_i + \sum\limits_{i=1}^{m} d_i^0 u_i},$$

в которой

$$U = \Big\{u_i\colon\ \sum_{i=1}^{m} u_i = 1, \ \ u_i \ge 0, \ \ i \in I\Big\}.$$

Определим квазивыпуклую и недифференцируемую функцию

$$\Theta(u) = \min_{x\in S}\frac{C(x,u)}{D(x,u)},$$

где билинейные функции $C(x,u)$ и $D(x,u)$ имеют вид

$$C(x,u) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_j u_i + \sum_{i=1}^{m} c_i^0 u_i,$$

$$D(x,u)=\sum_{i=1}^{m}\sum_{j=1}^{n}d_{ij}x_ju_i+\sum_{i=1}^{m}d_i^0u_i.$$

Рассмотрим задачу

$$\max_{u\in U}\Theta(u),$$

которая является двойственной для задачи (6.44).

Если для решения данной двойственной задачи применить субградиентные методы, то на каждом шаге, при фиксированных значениях $u=u^r$ решается задача дробно-линейного программирования

$$\frac{\sum\limits_{j=1}^{n}\alpha_j^rx_j\ +\ \alpha_0^r}{\sum\limits_{j=1}^{n}\beta_j^rx_j\ +\ \beta_0^r}\to\min,$$

$$\sum_{j=1}^{n}a_{kj}x_j\le b_k\quad (k=\overline{1,p}),$$

$$x_j\ge 0,$$

где

$$\alpha_j^r=\sum_{i=1}^{m}c_{ij}u_i^r\quad (j=\overline{1,n}),$$

$$\beta_j^r=\sum_{i=1}^{m}d_{ij}u_i^r\quad (j=\overline{1,n}),$$

$$\alpha_0^r=\sum_{i=1}^{m}c_i^0u_i^r,$$

$$\beta_0^r=\sum_{i=1}^{m}d_i^0u_i^r.$$

Если же для решения двойственной задачи $\max\limits_{u\in U}\Theta(u)$ применить параметрический метод, то при фиксированных u^r и v_r вместо задачи дробно-линейного программирования $\min\limits_{x\in S}\dfrac{\sum\limits_{j=1}^{n}\alpha_j^rx_j+\alpha_0^r}{\sum\limits_{j=1}^{n}\beta_j^rx_j+\beta_0^r}$ решаем задачу линейного программирования $\min\limits_{x\in S}\sum\limits_{j=1}^{n}\gamma_j^rx_j$, где $\gamma_j^r=\alpha_j^r-v_r\beta_j^r\quad (j=\overline{1,n})$.

Б. *Параметрический метод.* В параметрическом методе задача (6.44) сводится к решению следующей задачи параметрического программирования:

$$F(v) = \min_{x \in S} \max_{i \in I} \big(\varphi_i(x) - v\psi_i(x)\big), \tag{6.45}$$

которая представляет собой параметрическую задачу дискретного минимакса с билинейным функционалом. Однако, если фиксировать параметр v, то получим обычную задачу линейного дискретного минимакса. Для определения значений параметра v используем алгоритм параметрического метода.

В алгоритме параметрического метода необходимо выполнить следующие шаги.

Шаг 0. Находим допустимую точку $x^0 \in S$. Полагаем $v_1 = f(x^0)$ и $r = 1$.

Шаг r. Пусть $x^{r-1} \in S$ решение задачи (6.45), полученное на предыдущем шаге при фиксированном значении параметра v. Находим оптимальное решение x^r задачи (6.45) при $v_r = f(x^{r-1})$ и если $F(v_r) = 0$, то x^r является оптимальным решением задачи (6.44).

При $v = v_r$ задача дискретного линейного минимакса (6.45) сводится к эквивалентной задаче линейного программирования следующего вида:

$$t \to \min, \tag{6.46}$$

$$\sum_{j=1}^{n} q_{ij} x_j - t \le q_i^0 \quad (i = \overline{1,m}), \tag{6.47}$$

$$x \in S, \quad t \ge 0, \tag{6.48}$$

в которой $q_{ij} = c_{ij} - v_r d_{ij}$ $(i = \overline{1,m};\ j = \overline{1,n})$, а $q_i^0 = -c_i^0 + v_r d_i^0$ $(i = \overline{1,m})$.

Таким образом решение задачи дискретного минимакса с дробно-линейными функциями и линейными ограничениями (6.44) сводится к решению последовательности задач линейного программирования (6.46)–(6.48), полученных в результате применения параметрического метода решения задачи (6.45), которая соответствует задаче нахождения корня уравнения $F(v) = 0$.

B. *Метод частичной линеаризации.* Задача (6.44) переписывается в следующей эквивалентной форме:

$$t \to \min, \tag{6.49}$$

$$\sum_{j=1}^{n}(c_{ij} - td_{ij})x_j - d_i^0 t \leq -c_i^0 \quad (i = \overline{1,m}), \tag{6.50}$$

$$x \in S, \tag{6.51}$$

которая является билинейной задачей из-за произведения $t \cdot x_j$ переменных t и x_j в ограничениях (6.50). Для ее решения используется итеративный алгоритм с применением метода частичной линеаризации на каждом его шаге [62, 63], который заключается в замене произведения $t \cdot x_j$ на его линейное приращение в точке (x^r, t_r) путем фиксирования переменной $t = t_r$, где t_r определяется на предыдущем шаге за счет полученного решения x^r для переменных x.

В данном случае функции частичной линеаризации $H_i^r(x,t)$ в точке (x^r, t_r) на каждом шаге имеют вид:

$$H_i^r(x,t) = \varphi_i(x) - t_r\psi_i(x) - (t - t_r)\psi_i(x^r) =$$

$$= \sum_{j=1}^{n}(c_{ij} - t_r d_{ij}) - (t - t_r)\sum_{j=1}^{n} d_{ij}x_j^r + c_i^0 - t_r d_i^0 - (t - t_r)d_i^0 \quad (i = \overline{1,m}).$$

Алгоритм решения задачи (6.44) по такому параметрическому методу содержит следующие шаги.

Шаг 0. Пусть $x^0 \in S$ и $t_0 = f(x^0) = \max\limits_{i \in I} \dfrac{\varphi_i(x^0)}{\psi_i(x^0)}$. Тогда точка (x^0, t_0) является допустимым решением задачи (6.49)–(6.51). Полагаем $r = 1$ и переходим к шагу 1.

Шаг 1. Находим оптимальное решение (x^r, t_r) следующей задачи линейного программирования, полученной в результате частичной линеаризации задачи (6.49)–(6.51) в точке (x^{r-1}, t_{r-1})

$$t \to \min, \tag{6.52}$$

$$\sum_{j=1}^{n} q_{ij}x_j - q_i t \leq w_i \quad (i = \overline{1,m}), \tag{6.53}$$

$$x \in S, \tag{6.54}$$

в которой

$$q_{ij} = c_{ij} - t_{r-1} d_{ij} \quad (i = \overline{1,m};\ j = \overline{1,n}),$$

$$q_i = \sum_{j=1}^{n} d_{ij} x_j^{r-1} + d_i^0 \quad (i = \overline{1,m}),$$

$$w_i = -t_{r-1} \sum_{j=1}^{n} d_{ij} x_j^{r-1} - (c_i^0 - t_{r-1} d_i^0) \quad (i = \overline{1,m}).$$

Шаг 2. Если $t_r = t_{r-1}$, то x^r является оптимальным решением задачи (6.44). Останов, в противном случае переходим к шагу 3.

Шаг 3. Находим $t_r = f(x^r) = \max\limits_{i \in I} \dfrac{\varphi_i(x^r)}{\psi_i(x^r)}$, полагаем $r = r + 1$ и переходим к первому шагу алгоритма.

В алгоритмах **Б** и **В** решается последовательность линейных задач (6.46)–(6.48) и (6.52)–(6.54), в которых к первоначальным ограничениям прибавляются еще m ограничений, значения коэффициентов которых меняются на каждом шаге алгоритмов. Наличие таких ограничений не позволяет применить для их решения специальные алгоритмы, когда первоначальные ограничения задачи (6.44) имеют специальную структуру (например, блочно-диагональную или транспортного типа).

Г. *Схема декомпозиции по ограничениям.* Для решения задачи (6.44) специальной структуры или большой размерности можно применить алгоритм, основанный на схеме декомпозиции по ограничениям и субградиентном методе недифференцируемой оптимизации.

Для задачи (6.44) с учетом ее эквивалентной задачи (6.49)–(6.51) построим функцию Лагранжа по формуле

$$L(x,t,u) = t + \sum_{i=1}^{m} u_i \left(\sum_{j=1}^{n} (c_{ij} - t d_{ij}) x_j + c_i^0 - t d_i^0 \right),$$

в которой $u = \{u_i \geq 0, \quad i = \overline{1,m}\}$ – множители Лагранжа.

Тогда задача (6.44) сводится к решению задач

$$\max_{u \geq 0} L^*(u), \tag{6.55}$$

$$L^*(u) = \min_{t \geq 0} \min_{x \in S} L(x,t,u). \tag{6.56}$$

Функция $L^*(u)$ определена для любого $u \geq 0$ и является кусочно-линейной, вогнутой и недифференцируемой. Для ее максимизации используется один из алгоритмов субградиентного метода, на каждом шаге которого решается задача (6.56) при фиксированных значениях переменных u. Тогда на $(r+1)$-м шаге субградиентного метода необходимо выполнить следующие три этапа.

1. Решается задача (6.56) при $u = u^r$ и находится оптимальное решение $(x^*(u^r), t^*(u^r))$.

2. Определяются значения обобщенного градиента функции $L^*(u)$ в точке $u = u^r$ по формуле

$$g_i(u^r) = \sum_{j=1}^{n} (c_{ij} - t^*(u^r) d_{ij}) x_j^*(u^r) + c_i^0 - t^*(u^r) d_i^0, \quad i \in I.$$

3. Находятся новые значения переменных u по формуле

$$u_i^{r+1} = \max\{0, u_i^r + h_{r+1} g_i(u_i^r)\}, \quad i \in I,$$

где h_{r+1} – величина шага.

Если u^* – оптимальное решение задачи (6.55), то (x^*, t^*) – решение задачи (6.56) при $u = u^*$. Тогда (x^*, t^*) являются оптимальным решением задачи (6.49)–(6.51), а x^* – задачи (6.44).

Д. *Решение задачи (6.56).* На каждом шаге субградиентного метода при фиксированных двойственных переменных $u = u^r$ решается следующая задача:

$$\min_{x \in S} \min_{t \geq 0} \left[t + \sum_{i=1}^{m} u_i^r \left(\sum_{j=1}^{n} (c_{ij} - t d_{ij}) x_j + c_i^0 - t d_i^0 \right) \right], \tag{6.57}$$

которая является задачей билинейного программирования из-за произведений $t x_j \quad (j = \overline{1, n})$.

При фиксированном $t \geq 0$ задача (6.57) является задачей линейного программирования. Поэтому процесс решения данной задачи сводится к нахождению такого решения $x^* \in S$ и такого значения $t^* \geq 0$, которые в совокупности представляли бы оптимальное решение (x^*, t^*) задачи (6.57). Такой процесс может быть итерационным, на каждой итерации которого приближаемся к значению параметра t, сходящегося к оптимальному решению (x^*, t^*) задачи (6.57). Данный эффект имеет место в субградиентных методах, которые устойчивы относительно точности определения значений субградиентов. Поэтому для решения задачи (6.57) на каждом шаге субградиентного метода используем приближенный алгоритм решения задачи (6.57).

Для решения задачи (6.57) может быть использован одноитерационный приближенный алгоритм, в результате чего при фиксированных значениях переменных $u = u^*$ задачи (6.55) получим оптимальное решение задачи (6.57), а следовательно и исходной задаче (6.44). Такой алгоритм может быть построен, если при решении задачи (6.57) будем фиксировать значение параметра t по следующей формуле:

$$t^* = f(x^*(u^*)),$$

в которой $x^*(u^*) \in S$ – решение задачи (6.57), полученное на предыдущем шаге субградиентного метода.

Допустим, что после выполнения r шагов субградиентного метода имеем значение переменных $u = u^r$. Тогда задача (6.57) на очередном шаге субградиентного метода решается в следующем порядке:

- фиксируем значение t по формуле

$$t' = \begin{cases} 0, & \text{для} \quad r = 1; \\ t^*(u^r), & \text{для} \quad r \geq 2. \end{cases}$$

- находим оптимальное решение $x^*(u^r)$ задачи линейного программирования

$$\min_{x \in S} \sum_{j=1}^{n} q_j x_j \tag{6.58}$$

в которой $q_j = \sum_{i=1}^{m} u_i^r (c_{ij} - t' d_{ij}) \quad (j = \overline{1, n}).$

- находим новое значение для t по формуле

$$t^*(u^r) = f(x^*(u^r)).$$

В отличие от параметрического метода и метода частичной линеаризации, внутренняя задача линейного программирования (6.58) решается на множестве исходных ограничений, что является очень важным фактором при решение задач специальной структуры, такие как транспортного типа, блочной структуры и другие задачи, для решения которых имеются специальные эффективные методы и алгоритмы.

Пара полученных значений $(x^*(u^r), t^*(u^r))$ будет приближенным решением задачи (6.57), которые при $k \to \infty$ в субградиентном методе стремятся к оптимальному решению (x^*, t^*) задачи (6.57).

В предложенном алгоритме на каждом шаге решается задача линейного программирования (6.58), в которой меняются только значения коэффициентов целевой функции, а первоначальные ограничения остаются без изменений. Как уже отмечалось выше, такое обстоятельство имеет важное значение, когда решается задача (6.44) с ограничениями специальной структуры.

6.3. Задача оптимизации суммы дробных функций

Рассмотрим класс задач обобщенного дробного программирования, в котором необходимо оптимизировать сумму из дробных функций, а именно:

$$f(x) = \sum_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} \to \min, \tag{6.59}$$

$$h_k(x) \leq 0 \quad (k = \overline{1,p}), \tag{6.60}$$

в которой функции $\{\varphi_i(x),\ \ i \in I\}$, $\{-\psi_i(x),\ \ i \in I\}$ и $\{h_k(x),\ \ k = \overline{1,p}\}$ – непрерывны, выпуклы и дифференцируемы на R^n. Пусть ограничения (6.60) формируют выпуклое множество $S = \{x\colon\ h_k(x) \leq 0,\ \ k = \overline{1,p}\}$.

Если в дробных задачах типа (6.59), (6.60) предположить, что все $\psi_i(x) \equiv 1,\ \ i \in I$ для любого $x \in S$, то имеем обычную задачу выпуклой оптимизации, т.е. задачу оптимизации для которой локальный минимум совпадает с глобальным. Для решения задачи выпуклого программирования можно использовать общеизвестные методы. Однако, в случаях когда $\psi_i(x) \neq 1$, хотя бы для одного $i \in I$ или некоторого $x \in S$, задача дробной оптимизации (6.59), (6.60) не является задачей выпуклого программирования, т.е. она относится к классу многоэкстремальных задач, для решения которой необходимо использовать методы глобальной оптимизации [88], [194]–[196], [255]–[257], [278, 521].

В работе [445] указано, что для решения многоэкстремальной задачи (6.59), (6.60) предлагаются различные алгоритмы, основная часть которых основаны на идеях параметрического метода [15, 101, 176, 186, 445], методах ветвей и границ [64], [67]–[70], [323, 325, 461, 533, 555] и других методов глобальной оптимизации [70, 71, 159, 96, 462, 463, 569, 549, 551]. Различные алгоритмы глобальной оптимизации предложены для случая, когда ограничения (6.60) формирует некоторое многогранное множество, т.е. оптимизации нелинейной функции $f(x)$ при линейных ограничениях. Для случая, когда $m = 1$ имеем обычную задачу дробно-выпуклого программирования, которая рассмотрена в предыдущей главе, и для решения которой предложены различные методы и алгоритмы. При $m = 1$ и линейных функциях $\varphi_i(x)$, $\psi_i(x)$ и $h_k(x)$ получаем обычную задачу дробно-линейного программирования, которая рассмотрена в четвертой главе.

При $m = 2$ имеем одну из возможных двух задач минимизации функции $f(x) = \varphi_1(x) + \dfrac{\varphi_2(x)}{\psi_2(x)}$ или $f(x) = \dfrac{\varphi_1(x)}{\psi_1(x)} + \dfrac{\varphi_2(x)}{\psi_2(x)}$, которые рассмотрены в работах [9, 10, 102, 109, 248, 249, 355, 431, 477, 631, 700], для решения которых предложены параметрические методы с одним или двумя параметрами. При линейных функциях $\varphi_i(x)$, $\psi_i(x)$ и $h_k(x)$ данные задачи сводятся к решению задач квадратичной оптимизации при линейных ограничениях, или к задачам параметрического программирования. Для решения так-называемых обобщенных задач линейного и дробно-линейного программирования предложены модификации симплекс-метода [9, 10, 101, 248, 249, 431, 700], а также рассмотрены двойственные задачи и соответствующие теоремы двойственности [109, 477].

Для случая, когда $m \geq 2$ для решения задачи минимизации суммы дробно-линейных функций при линейных ограничениях предложены алгоритмы метода ветвей и границ [250, 323, 325, 533, 555], и другие точные и приближенные алгоритмы [135, 136, 186, 250, 305, 306, 313, 585].

Также следует отметить, что задачу (6.59), (6.60) можно рассматривать как на минимум, так и на максимум. Тогда она может быть преобразована в эквивалентную задачу вогнутой минимизации или выпуклой максимизации. Такие задачи рассмотрены в работах [70, 71, 462, 463, 569], в которых в схеме глобальной оптимизации используются различные методы аппроксимации. В работе [208] указано на то, что задача (6.59), (6.60) является NP-полной задачей.

Одним из основных методов решения задачи дробной оптимизации является ее параметрический анализ. В случае задачи минимизации суммы дробно-выпуклых (квазивыпуклых) функций рассматривается параметрический анализ задачи (6.59), (6.60). В работе [15] задача (6.59), (6.60) сводится к многопараметрической задаче:

$$Z(x,\lambda) = \sum_{i\in I} (\varphi_i(x) - \lambda_i \psi_i(x)) \to \min,$$

$$h_k(x) \leq 0 \quad (k = \overline{1,p})$$

для решения которой приводятся различные алгоритмы и методы [15, 101, 176, 186] нахождения значений параметров $\{\lambda_i\}$, обеспечивающие оптимальность полученного решения x^* исходной задачи (6.59), (6.60). Следует отметить, что если $\lambda_i \geq 0$, то функция $Z(x,\lambda)$ является выпуклой по x и линейной по λ. Если рассмотрим функцию

$$F(\lambda) = \min_{x\in S} Z(x,\lambda),$$

то она является кусочно-линейной, недифференцируемой и вогнутой функцией. Тогда в соответствии с параметрическим методом нужно найти решение уравнения $F(\lambda) = 0$. Если предположить, что $\varphi_i(x) \geq 0$, $\psi_i(x) > 0, \;\; i \in I$ для любого $x \in S$, то функция $F(\lambda) \geq 0$ для любого $\lambda_i \geq 0 \;\; (i = \overline{1, m})$. Тогда для нахождения решения уравнения $F(\lambda) = 0$ можно рассматривать задачу $\min\limits_{\lambda} F(\lambda)$, для решения которой по переменным λ можно применить методы недифференцируемой оптимизации. Однако задача $\min\limits_{\lambda} F(\lambda)$ является многоэкстремальной задачей, а субградиентный метод не всегда обеспечит получение глобального минимума. Поэтому для решения задачи (6.59), (6.60) применяются более универсальные методы многоэкстремальной оптимизации типа метода ветвей и границ [64], [67]–[70], [323, 325, 461, 533, 555]. Также можно рассматривать параметрический метод с некоторым правилом выбора значений параметров $\lambda_i, \;\; i \in I$ [15, 101, 176, 186].

Задачу (6.59), (6.60) можно свести к другому виду параметрической задачи, а именно, если обозначить через $\lambda_i = \dfrac{\varphi_i(x)}{\psi_i(x)}, \;\; i \in I$, то получим задачу

$$\sum_{i \in I} \lambda_i \to \min; \tag{6.61}$$

$$\varphi_i(x) - \lambda_i \psi_i(x) = 0, \quad i \in I; \tag{6.62}$$

$$h_k(x) \leq 0 \quad (k = \overline{1, p}); \tag{6.63}$$

$$\lambda_i \geq 0, \;\; i \in I, \tag{6.64}$$

для решения которой предлагаются метод внутренних точек [206] и конические алгоритмы [159].

Рассмотрим для начала некоторые утверждения, показывающие на эквивалентность задачи (6.59), (6.60) и задачи (6.61)–(6.64), которые в дальнейшем будут использованы при построение алгоритмов решения задачи (6.59), (6.60).

Пусть $\lambda = \{\lambda_i \geq 0, \;\; i \in I\}$ вектор параметров, которые согласно (6.62) определяются по формуле

$$\lambda_i = \frac{\varphi_i(x)}{\psi_i(x)}, \;\; i \in I.$$

Имеют место утверждения.

Теорема 6.7. *Если x^* – оптимальное решение задачи* (6.59), (6.60), *то существует такие $\lambda_i^* = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$, что (x^*, λ^*) является оптимальным решением задачи* (6.61)–(6.64).

Доказательство. Пусть x^* – оптимальное решение задачи (6.59), (6.60). Тогда имеем
$$\sum_{i\in I}\frac{\varphi_i(x^*)}{\psi_i(x^*)}\leq\sum_{i\in I}\frac{\varphi_i(x)}{\psi_i(x)}$$
для любого $x\in S$. Если учесть, что для любого $i\in I$ параметры λ_i^* и λ_i определяются однозначно по формулам
$$\lambda_i^*=\frac{\varphi_i(x^*)}{\psi_i(x^*)}\ \text{ и }\ \lambda_i=\frac{\varphi_i(x)}{\psi_i(x)},$$
то имеем неравенство
$$\sum_{i\in I}\lambda_i^*\leq\sum_{i\in I}\lambda_i, \tag{6.65}$$
а пара (x^*,λ^*) удовлетворяет равенству (6.62), т.е. имеем $\varphi_i(x^*)-$ $-\lambda_i^*\psi_i(x^*)=\varphi_i(x^*)-\dfrac{\varphi_i(x^*)}{\psi_i(x^*)}\cdot\psi_i(x^*)=0$, т.е. (x^*,λ^*) является допустимым решением задачи (6.61)–(6.64). Так как неравенство (6.65) выполняется для любого $x\in S$ и для соответствующих параметров λ_i, определенных из равенства (6.62), то пара (x^*,λ^*) является и оптимальным решением задачи (6.61)–(6.64). Теорема доказана.

Теорема 6.8. *Если* (x^*,λ^*) *– оптимальное решение задачи* (6.61)–(6.64)*, то* x^* *является оптимальным решением задачи* (6.59), (6.60)*.*

Доказательство. Пусть (x^*,λ^*) – оптимальное решение задачи (6.61)–(6.64). Тогда $x^*\in S$ и тем самым является допустимым планом задачи (6.59), (6.60). Так как x^* удовлетворяет равенствам (6.62), то имеем $\lambda_i^*=\dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$ и $\lambda_i=\dfrac{\varphi_i(x)}{\psi_i(x)}$, которые удовлетворяют равенствам (6.62) при любом значении $x\in S$. Тогда из оптимальности пары (x^*,λ^*) для задачи (6.61)–(6.64) получим неравенство
$$\sum_{i\in I}\lambda_i^*\leq\sum_{i\in I}\lambda_i,$$
для любого $x\in S$.

Таким образом, согласно формулам определения параметров λ_i^* и λ_i имеет место неравенство:
$$\sum_{i\in I}\frac{\varphi_i(x^*)}{\psi_i(x^*)}\leq\sum_{i\in I}\frac{\varphi_i(x)}{\psi_i(x)}$$

для любого $x \in S$, т.е.

$$f(x^*) \leq f(x).$$

Следовательно, x^* является оптимальным решением задачи (6.59), (6.60). Теорема доказана.

Заметим, что задача (6.61)–(6.64) может быть сформулирована следующим образом: найти те значения переменных $x \in S$, для которых сумма переменных λ_i, определенных из равенствах (6.62), принимает минимальное значение. Другими словами, необходимо найти значения x^* переменных $x \in S$, для которых параметры λ_i^*, определенные по формуле

$$\lambda_i^* = \frac{\varphi_i(x^*)}{\psi_i(x^*)}, \quad i \in I$$

обеспечат минимальное значение функционала (6.61).

В данной монографии для задачи (6.59), (6.60) рассматриваются двойственные аспекты, приводятся параметрические методы, субградиентные методы для решения прямой или двойственной задачи с учетом параметрической задачи $\min\limits_{x \in S} Z(x, \lambda)$. Также рассматриваются методы недифференцируемой оптимизации, которые основаны на функции Лагранжа и схеме декомпозиции по ограничениями в параметрическом методе.

Рассмотрим подробнее эти методы решения задач оптимизации суммы дробно-выпуклых и дробно-линейных функций.

1. Двойственная задача. Если рассмотреть задачу (6.59), (6.60) как общую задачу математического программирования, то один из вариантов двойственной задачи может быть построен на базе функции Лагранжа. Как отмечено в п. 1.7 для задачи (6.59), (6.60) может быть построена следующая функция Лагранжа

$$L(x, z) = \sum_{i=1}^{m} \frac{\varphi_i(x)}{\psi_i(x)} + \sum_{k=1}^{p} z_k h_k(x),$$

которая представляет собой функцию в виде суммы выпуклых и дробно-выпуклых, что говорит о ее невыпуклости по переменными x и с глобальным характером с точки зрения ее оптимизации.

Для анализа двойственной задачи для задачи (6.59), (6.60) рассматривается задача нахождения седловой точки функции Лагранжа в виде $\max\limits_{z} \min\limits_{x} L(x, z)$. Если ввести новую функцию $L^*(z) = \min\limits_{x} L(x, z)$, то получим задачу $\max\limits_{z} L^*(z)$, которую и берем в качестве двойственной для задачи (6.59), (6.60).

Естественно, что для полученных задач для некоторых z и любых $x \in S$ имеет место неравенство

$$L^*(z) \leq f(x),$$

а для оптимального $x^* \in S$ найдутся такие z^*, для которых $L^*(z^*) = f(x^*)$. Такая точка (x^*, z^*) является седловой точкой функции Лагранжа $L(x, z)$ для задачи (6.59), (6.60). Однако, рассмотрение такого класса двойственных задач не выводит нас из общего свойства глобальной оптимизации для задачи (6.59), (6.60). Далее будут предложены процедуры применения субградиентных методов нахождения седловой точке функции Лагранжа с учетом параметрического метода.

2. Нелинейный многопараметрический метод. В нелинейном многопараметрическом методе решение задачи $\min\limits_{x \in S} Z(x, \lambda)$ и нахождение корня уравнения $F(\lambda) = 0$ может быть проведено по правилу покоординатного фиксирования на каждом шаге параметра λ по формуле $\lambda_i = \dfrac{\varphi_i(x)}{\psi_i(x)}$, $i \in I$.

Тогда, если использовать некоторый итерационный процесс, в котором на r-м шаге при очередном допустимом решением x^*_{r-1}, полученное оптимальное решение x^*_r задачи $\min\limits_{x \in S} Z(x, \lambda^{r-1})$, при $\lambda_i^{r-1} = \dfrac{\varphi_i(x^*_{r-1})}{\psi_i(x^*_{r-1})}$, $i \in I$, будет стремиться к выполнению для каждого $i \in I$ неравенства

$$\left|\varphi_i(x^*_r) - \lambda_i^{r-1}\psi_i(x^*_r)\right| =$$
$$= \left|\left(\frac{\varphi_i(x^*_r)}{\psi_i(x^*_r)} - \lambda_i^{r-1}\right)\psi_i(x^*_r)\right| = \left|\lambda_i^r - \lambda_i^{r-1}\right|\psi_i(x^*_r) \leq \delta_i \psi_i(x^*_r), \quad i \in I,$$

получим, что $|\lambda_i^r - \lambda_i^{r-1}| \leq \delta_i$, $i \in I$, где $\lambda_i^r = \dfrac{\varphi_i(x^*_r)}{\psi_i(x^*_r)}$, $i \in I$, а δ_i – достаточно малые неотрицательные числа. В результате чего имеем

$$|F(\lambda^{r-1})| = |Z(x^*, \lambda^{r-1})| \leq \sum_{i \in I} \left|\lambda_i^r - \lambda_i^{r-1}\right|\psi_i(x^*) \leq \delta,$$

где δ – достаточно малое неотрицательное число.

По такому процессу будем искать два последовательных решения x^*_{r-1} и x^*_r задачи $\min\limits_{x \in S} Z(x, \lambda)$ при фиксированном значение параметра λ, устойчивые не в целом по функции $f(x)$, а по каждой дробной функции $f_i(x) = \dfrac{\varphi_i(x)}{\psi_i(x)}$, $i \in I$. В таком случае решение x^*_r может быть взято в качестве приближенного решения задачи суммы дробно-выпуклых функций (6.59), (6.60).

Если провести анализ устойчивости решений в многопараметрической задачи $F(\lambda) = \min\limits_{x\in S} Z(x,\lambda)$, то оптимальное решение x^* будет оптимально устойчиво, если для каждого параметра λ_i найдется такой интервал значений $[\underline{\lambda}_i, \overline{\lambda}_i]$, для которого при любых значений $\lambda_i \in [\underline{\lambda}_i, \overline{\lambda}_i]$, $i \in I$ решение x^* остается оптимальным решением для задачи с фиксированными значениями параметров λ_i, т.е. λ_i является корнем уравнения $F_i(\lambda_i) = \min\limits_{x\in S} (\varphi_i(x) - \lambda_i\psi_i(x)) = 0$, или же для которого имеет место равенство $\varphi_i(x) - \lambda_i\psi_i(x) = 0$.

Такой анализ параметрических задач может быть использован для решения задачи $\min\limits_{x\in S} Z(x,\lambda)$ и нахождения корня уравнения $F(\lambda) = 0$, с учетом корня λ_i каждого уравнения $F_i(\lambda_i) = 0, \quad i \in I$.

Тогда в многопараметрическом методе решения задачи оптимизации суммы дробных функций (6.59), (6.60) может быть использован следующий итерационный процесс.

Шаг 0. Находим допустимую точку $x^0 \in S$ путем решения оптимизационной задачи $\min\limits_{t}\{t : \quad h_k(x) \leq t, \quad k = \overline{1,p}\}$. Если в полученном решение (x^0, t^0) данной задачи имеем $t^0 = 0$, то x^0 является допустимым решением, т.е. $x^0 \in S$. Если же $t^0 \neq 0$, то исходная задача (6.59), (6.60) не имеет допустимых решений.

Пусть $x^0 \in S$, тогда полагаем $\lambda_i^0 = \dfrac{\varphi_i(x^0)}{\psi_i(x^0)} \geq 0, \quad i \in I$, т.е. λ_i^0 является корнем уравнения $F_i(\lambda_i) = 0$ при $x = x^0$.

Шаг r. Пусть $x^{r-1} \in S$ – допустимое решение, полученной на предыдущем шаге, а $\lambda_i^{r-1} = \dfrac{\varphi_i(x^{r-1})}{\psi_i(x^{r-1})} \geq 0, \quad i \in I$, т.е. λ_i^{r-1} является корнем уравнения $F_i(\lambda_i^{r-1}) = 0$ при $x = x^{r-1}$.

Тогда:

- находим оптимальное решение x^r задачи выпуклого программирования

$$\min_{x\in S} \sum_{i\in I} \left(\varphi_i(x) - \lambda_i^{r-1}\psi_i(x)\right);$$

- полагаем $\lambda_i^r = \dfrac{\varphi_i(x^r)}{\psi_i(x^r)} \geq 0, \quad i \in I$;

- если $|\lambda_i^r - \lambda_i^{r-1}| \leq \delta_i, \ i \in I$, то получим $|F(\lambda^r)| \leq \sum\limits_{i\in I} \left|F_j(\lambda_j^r)\right| = \sum\limits_{i\in I} \left|\lambda_i^r - \lambda_i^{r-1}\right| \psi_i(x^r) \leq \delta$, где значения $\delta_i \geq 0, \quad i \in I$ и $\delta \geq 0$ выбираются соответствующим образом, для того чтобы обеспечить сходимость данного процесса.

Если предположить, что имеются некоторые способы корректировки параметров λ_i^r на каждом шаге итерационного процесса, то полученная последовательность точек $\{x^r\}$ сходится к оптимальному решению x^* задачи (6.59), (6.60).

3. Алгоритм метода частичной линеаризации. Для решения невыпуклой задачи (6.61)–(6.64) может быть использован метод частичной линеаризации, который сводил-бы ее к задаче выпуклого программирования, а именно, минимизации линейной функции при выпуклых ограничениях.

Если рассмотреть функции

$$H_i(x, \lambda_i) = \varphi_i(x) - \lambda_i \psi_i(x), \quad i \in I,$$

которые являются билинейно-выпуклыми, то можно провести частичную линеаризацию таких функций в точке (x^r, λ^r) по переменным $\lambda_i,\ i \in I$ в виде следующих выражений

$$H_i^r(x, \lambda_i) = \varphi_i(x) - \lambda_i^r \psi_i(x) - (\lambda_i - \lambda_i^r)\psi_i(x^r), \quad i \in I.$$

Тогда вместо задачи (6.61)–(6.64) можно рассматривать задачу выпуклого программирования:

$$\sum_{i \in I} \lambda_i \to \min,$$

$$\varphi_i(x) - \lambda_i^r \psi_i(x) - \lambda_i \psi_i(x^r) + \lambda_i^r \psi_i(x^r) \le 0, \quad i \in I,$$

$$h_k(x) \le 0 \quad (k = \overline{1,p}),$$

$$\lambda_i \ge 0 \quad i \in I.$$

Тогда для решения задачи (6.61)–(6.64) используем следующий алгоритм параметрического метода с выполнением операции частичной линеаризации на каждом его шаге.

Шаг 0. Пусть имеем допустимую точку $x^0 \in S$, определяем значения параметров $\lambda_i^0 = \dfrac{\varphi_i(x^0)}{\psi_i(x^0)}, \quad i \in I.$

Шаг r. Пусть $x^{r-1} \in S$ – допустимое решение, найденное на предыдущем шаге, а $\lambda_i^{r-1} = \dfrac{\varphi_i(x^{r-1})}{\psi_i(x^{r-1})}, \quad i \in I$. Тогда:

- находим решение (x^r, λ^r) задачи выпуклого программирования

$$\sum_{i\in I} \lambda_i \to \min,$$

$$\varphi_i(x) - \lambda_i^{r-1}\psi_i(x) - \lambda_i\psi_i(x^{r-1}) \le -\lambda_i^{r-1}\psi_i(x^{r-1}), \quad i \in I,$$

$$h_k(x) \le 0 \quad (k = \overline{1,p}),$$

$$\lambda_i \ge 0 \quad i \in I;$$

- если $|\lambda_i^r - \lambda_i^{r-1}| \le \delta_i$, тогда x^r является решением задачи (6.61)–(6.64), а также и первоначальной задачи (6.59), (6.60). Останов, в противном случае

- полагаем $\lambda_i^r = \dfrac{\varphi_i(x^r)}{\psi_i(x^r)}$, $i \in I$ и выполняем очередной $(r+1)$-й шаг алгоритма.

Данный алгоритм, как и многопараметрический метод, генерирует последовательность точек (x^r, λ^r), которые стремятся к оптимальному решению задачи (6.61)–(6.64), и, следовательно, и к оптимальному решению исходной задачи оптимизации суммы дробно-выпуклых функций (6.59), (6.60).

4. Субградиентные методы. Как отмечено ранее, для решения задачи выпуклой оптимизации субградиентные методы могут быть использованы в двух вариантах. В первом варианте субградиентный метод применяется для оптимизации по переменных прямой задачи, а во втором варианте – по переменным двойственной задачи. Рассмотрим особенности применения субградиентных методов решения задачи суммы дробно-выпуклых функций (6.59), (6.60).

А. *Субградиентный метод решения задачи* (6.59), (6.60). Решение задачи (6.59), (6.60) субградиентным методом проводится с использованием эквивалентной параметрической задачи выпуклого программирования

$$F(\lambda) = \min_{x\in S} Z(x,\lambda) = \min_{x\in S} \sum_{i\in I} \left(\varphi_i(x) - \lambda_i\psi_i(x)\right).$$

Так как функции $\varphi_i(x)$ и $-\psi_i(x)$ являются выпуклыми, то при $\lambda \ge 0$ функция $Z(x,\lambda)$ является выпуклой по x и линейной по λ. Если предположить, что функции $\varphi_i(x) \ge 0$, $i \in I$ и $\psi_i(x) > 0$, $i \in I$, то при выборе значений параметров по формуле $\lambda_i = \dfrac{\varphi_i(x)}{\psi_i(x)}$, $i \in I$ для любого $x \in S$, параметры λ_i всегда будут неотрицательны для любого $x \in S$ и тогда функция $Z(x,\lambda)$ при таком фиксированием параметров λ_i будет выпуклой по x. Таким образом, при таких предположениях для нахождения локального, и тем самым глобального, минимума функции $Z(x,\lambda)$ по переменным x при фиксированных $\lambda_i = \dfrac{\varphi_i(x)}{\psi_i(x)}$ может быть использован один из субградиентных методов, в частности метод эллипсоидов, как это было сделано для задач выпуклого и дробно-выпуклого программирования (см. п. 2.5. и 5.3.).

Рассмотрим случай задачи (6.59), (6.60), когда множество S определено ограничениями (6.2), т.е. $S = \{x \in R^n : h_k(x) \le 0, \quad k = \overline{1,p}\}$, а $\varphi_i(x) \ge 0$, $\psi_i(x) > 0$, $\quad i \in I$ для любого $x \in S$. Тогда для решения параметрической задачи обобщенного дробного программирования

$$F(\lambda) = \min_{x\in S} Z(x,\lambda) = \min_{x\in S} \sum_{i\in I} \left(\varphi_i(x) - \lambda_i \psi_i(x)\right),$$

применяем метод эллипсоидов (см. п. 6.3.), который в данном случае имеет следующие особенности при вычисления значений обобщенного градиента.

Пусть на r-м шаге метода эллипсоидов имеем точку x_r. Тогда значения обобщенного градиента функции $Z(x,\lambda)$ в точке x_r определяются по формуле

$$g(x_r) = \begin{cases} g_{_Z}(x_r) = \sum\limits_{i\in I} \left(g_{\varphi_i}(x_r) - \lambda_i^r g_{\psi_i}(x_r)\right), & \text{если} \quad h_{k_r}(x_r) \le 0; \\ g_h(x_r) = g_{h_{k_r}}(x_r), \quad \text{если} \quad h_{k_r}(x_r) > 0, \end{cases}$$

где k_r – индекс, для которого достигается поточечный максимум по функциям ограничений, а $h_{k_r}(x_r)$ – значение этого максимума в точке x_r, т.е. k_r – индекс, для которого $h_{k_r}(x_r) = \max\limits_{1\le k\le p} h_k(x_r)$;

λ_i^r – значения параметров λ_i, которые определяются для текущей точки x_r по формуле

$$\lambda_i^r = \begin{cases} \dfrac{\varphi_i(x_r)}{\psi_i(x_r)}, & \text{если} \quad \varphi_i(x_r) \ge 0 \quad \text{и} \quad \psi_i(x_r) > 0; \\ 0, \quad \text{в противном случае}, \end{cases}$$

$g_{_Z}(x_r)$ – значение субградиента функции $Z(x,\lambda)$ в точке x_r при параметрах $\lambda_i = \lambda_i^r, \quad i \in I$;

$g_h(x_r)$ – значение субградиента, определенное для одной из функций ограничений;

$g_{\varphi_i}(x_r)$, $g_{\psi_i}(x_r)$, $\quad i \in I$ и $g_{h_{k_r}}(x_r)$ – значения субградиентов в точке x_r соответственно для функций $\varphi_i(x)$, $\psi_i(x)$, $\quad i \in I$ и $h_{k_r}(x)$.

Полученное решение x^* по методу эллипсоидов для задачи

$$\min_{x\in S} Z(x,\lambda) = \min_{x\in S} \sum_{i\in I} \left(\varphi_i(x) - \lambda_i^* \psi_i(x)\right),$$

при $\lambda_i^* = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$, $i \in I$ будет оптимальным решением для исходной задачи (6.59), (6.60), а именно для задачи

$$\min_{x\in S} f(x) = \min_{x\in S} \sum_{i\in I} \frac{\varphi_i(x)}{\psi_i(x)}.$$

Действительно, так как x^* – оптимальное решение задачи $\min\limits_{x\in S} Z(x,\lambda^*) = Z(x^*,\lambda^*)$ при $\lambda_i^* = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$, $i \in I$, то получим, что

$$F(\lambda^*) = Z(x^*,\lambda^*) = \sum_{i\in I} \left(\varphi_i(x^*) - \lambda_i^* \psi_i(x^*)\right) =$$

$$= \sum_{i\in I} \left(\varphi_i(x^*) - \frac{\varphi_i(x^*)}{\psi_i(x^*)} \psi_i(x^*)\right) = 0.$$

Таким образом λ^* является решением уравнения $F(\lambda) = 0$. Тогда в соответствии с параметрическим методом x^* является решением задачи (6.59), (6.60).

Б. *Субградиентный метод решения двойственной задачи.* Рассмотрим особенности применения субградиентных методов при решении двойственной задачи $\max\limits_{z} L^*(z)$ с учетом параметрической задачи. В соответствие с двойственной лагранжевой задачей решение задачи (6.59), (6.60) сводится к нахождению седловой точке (x^*, z^*) функции Лагранжа $L(x,z)$, которая соответствует задачам:

$$L(x^*,z^*) = \max_z \min_x L(x,z) = \min_x \max_z L(x,z).$$

С учетом их разделения на подзадачи по уровням анализа имеем следующие две задачи:

- задачу внутреннего уровня $L^*(z) = \min\limits_{x} L(x,z)$, которая представляет собой задачу минимизации суммы дробно-выпуклых функций и суммы выпуклых функций по переменным x;

- задачу внешнего уровня $\max\limits_{z} L^*(z)$, которая является двойственной задачей для (6.59), (6.60).

Так как по определению, функция $L^*(z)$ является кусочно-линейной, вогнутой и не везде дифференцируемой по переменным z, то для ее максимизации используются методы недифференцируемой оптимизации. Если для этого применить один из субградиентных методов, то на каждом его шаге, при фиксированных значениях переменных z необходимо решить задачу невыпуклого программирования $\min\limits_x L(x,z)$, которая является многоэкстремальной задачей.

Пусть для решения задачи $\max\limits_z L^*(z)$ используется один из субградиентных методов, на r-м шаге которого имеем текущую точку z^r. Тогда на очередном $(r+1)$-м шаге необходимо выполнить следующие три основных этапа.

1. Решить задачу $\min\limits_x L(x,z)$ при фиксированных $z_k = z_k^r \quad (k = \overline{1,p})$, для которой функционал по переменным x имеет вид:

$$F_r(x) = \sum_{i=1}^{m} \frac{\varphi_i(x)}{\psi_i(x)} + \sum_{k=1}^{p} z_k^r h_k(x),$$

которая, как отмечено выше, является многоэкстремальной задачей оптимизации. Для решения задачи безусловной минимизации невыпуклого функционала $F_r(x)$ можно использовать различные алгоритмы и методы, среди которых параметрический метод. Для этого рассматривается функция $Q_r(x,\lambda)$ и параметрическая задача $\beta(\lambda) = \min\limits_x Q_r(x,\lambda)$, в которой

$$Q_r(x,\lambda) = \sum_{i=1}^{m} (\varphi_i(x) - \lambda_i \psi_i(x)) + \sum_{k=1}^{p} z_k^r h_k(x)$$

и находится корень $\lambda(x_r, z^r)$ уравнения $\beta(\lambda) = 0$.

Таким образом, вместо задачи $\min\limits_x F_r(x)$ решаем задачу $\min\limits_x Q_r(x,\lambda)$ при фиксированном значение параметра $\lambda = \lambda_i^{r-1}$. Для этого:

- определяем значение параметра

$$\lambda_i^{r-1} = \begin{cases} \dfrac{\varphi_i(x_{r-1}^*)}{\psi_i(x_{r-1}^*)}, & i \in I, \text{ если } \varphi_i(x_{r-1}^*) \geq 0, \quad \psi_i(x_{r-1}^*) > 0, \\ 0, \text{ в противном случае,} \end{cases}$$

где x_{r-1}^* – найденное решение задачи $\min\limits_x Q_r(x, \lambda^{r-1})$ на предыдущем шаге субградиентного метода;

- находим оптимальное решение $x_r^* = x(z^r)$ задачи выпуклой оптимизации $\min\limits_x Q_r(x, \lambda^{r-1})$.

2. Находим значения обобщенных градиентов функции $L^*(z)$ по переменным z в точке z^r с учетом полученного решения x_r^* и функции $Q_r(x, \lambda^{r-1})$ по формулам

$$g_{L^*}^{z_k} = g_{Q_r}^{z_k} = h_k(x_r^*) \quad (k = \overline{1,p}).$$

3. Находим значения z_k^{r+1} по формуле

$$z_k^{r+1} = \max\left\{0,\ z_k^r + \gamma_{r+1} g_{Q_r}^{z_k}\right\} \quad (k = \overline{1,p}),$$

где γ_{r+1} – величина шага в субградиентном методе.

Пусть z^* – решение задачи $\max\limits_z L^*(z)$, а x^* – решение задач $\min\limits_x L(x, z^*)$, $\min\limits_x F(x)$ или же $\min\limits_x Q(x, \lambda^*)$. Таким образом, при $\lambda_i^* = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$, имеем, что точка x^* является оптимальным решением задачи $\min\limits_x Q(x, \lambda^*)$ и задачи $\min\limits_x F(x)$, т.е. имеем $Q(x^*, \lambda^*) = \min\limits_x Q(x, \lambda^*)$, для которой по теории выпуклого программирования выполняются условия $z_k^* h_k(x^*) = 0 \quad (k = \overline{1,p})$. Тогда получим что

$$Q(x^*, \lambda^*) = \min_x Q(x, \lambda^*) =$$

$$= \sum_{i=1}^m (\varphi_i(x^*) - \lambda_i^* \psi_i(x^*)) = \sum_{i=1}^m \left(\varphi_i(x^*) - \frac{\varphi_i(x^*)}{\psi_i(x^*)} \psi_i(x^*)\right) = 0,$$

т.е. λ^* является решением уравнения $\beta(\lambda) = 0$, и в соответствии с параметрическим методом x^* является решением задачи (6.59), (6.60).

5. Схема декомпозиции по ограничениям. Для решения задачи (6.59), (6.60) (для случая, когда ограничения задачи имеют специальную структуру) рассмотрим алгоритм, который соответствует некоторому итерационному процессу (относительно двойственных переменных u для ограничений (6.60)) недифференцируемой оптимизации.

Для задачи (6.59), (6.60) рассмотрим эквивалентную к ней задачу (6.61)–(6.64) и построим алгоритм решения данной задачи, основанный на функции Лагранжа, схеме декомпозиции по ограничениям и субградиентных методах. Для этого построим функцию Лагранжа для задачи (6.61), (6.62) по формуле

$$L(x, \lambda, u) = \sum_{i \in I} \lambda_i + \sum_{i \in I} u_i(\varphi_i(x) - \lambda_i \psi_i(x)),$$

где $u = \{u_i, \quad i \in I\}$ – множители Лагранжа для ограничений (6.62).

Тогда задача (6.61)–(6.64), и тем самым и (6.59), (6.60) сводится к решению задачи нахождения седловой точки функции Лагранжа $L(x, \lambda, u)$, т.е. к решению задачи

$$L(x^*, \lambda^*, u^*) = \max_u \min_{\lambda \geq 0} \min_{x \in S} L(x, \lambda, u). \tag{6.66}$$

В свою очередь задачу (6.66) можно свести к решению задачи

$$\max_u L^*(u), \tag{6.67}$$

где

$$L^*(u) = \min_{\lambda \geq 0} \min_{x \in S} L(x, \lambda, u). \tag{6.68}$$

Функция $L^*(u)$ определена для любого u и является кусочно-линейной, вогнутой и недифференцируемой. Поэтому для решения задачи (6.67) используются субградиентные методы, на каждом шаге которых необходимо решать задачу (6.68) при фиксированных значениях переменных u.

Пусть для решения задачи (6.67) используется некоторый субградиентный метод. Тогда на $(r+1)$-м $(r = 0, 1, 2, \dots)$ шаге необходимо выполнить следующие три основных этапа.

1. Решить задачу (6.68) при фиксированных $u = u^r$ и найти оптимальное решение $(x^*(u^r), \lambda^*(u^r))$.

2. Определить значения обобщенного градиента функции $L^*(u)$ в точке $u = u^r$ по формуле

$$g_i(u_i^r) = \varphi_i(x^*(u^r)) - \lambda_i^*(u^r)\psi_i(x^*(u^r)), \quad i \in I.$$

3. Найти новые значения u^{r+1} по формуле

$$u_i^{r+1} = u_i^r + \gamma_{r+1} g_i(u_i^r), \quad i \in I,$$

где γ_{r+1} – величина шага.

Теорема 6.9. *Пусть u^* – оптимальное решение задачи* (6.67), *а (x^*, λ^*) – оптимальное решение задачи* (6.68) *при фиксированных $u = u^*$. Тогда (x^*, λ^*) – оптимальное решение задачи* (6.61)–(6.64), *а x^* – оптимальное решение задачи* (6.59), (6.60).

Доказательство. Пусть u^* является оптимальным решением задачи (6.67). Тогда в соответствие с субградиентным методом имеем

$g_i(u_i^*) = \varphi_i(x^*) - \lambda_i^* \psi_i(x^*) = 0$ или $\lambda_i^* = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$, где (x^*, λ^*) является оптимальным решением задачи (6.68) при $u = u^*$, т.е. имеем неравенство $L(x^*, \lambda^*, u^*) \leq L(x, \lambda, u^*)$, для любого $x \in S$ и $\lambda_i = \dfrac{\varphi_i(x)}{\psi_i(x)}$. Из данного неравенства получим следующие соотношения:

$$\sum_{i\in I} \lambda_i^* + \sum_{i\in I} u_i^*(\varphi_i(x^*) - \lambda_i^*\psi_i(x^*)) \leq \sum_{i\in I} \lambda_i + \sum_{i\in I} u_i^*(\varphi_i(x) - \lambda_i\psi_i(x)),$$

$$\sum_{i\in I} \lambda_i^* + \sum_{i\in I} u_i^* \Big(\varphi_i(x^*) - \frac{\varphi_i(x^*)}{\psi_i(x^*)}\psi_i(x^*)\Big) \leq \sum_{i\in I} \lambda_i + \sum_{i\in I} u_i^* \Big(\varphi_i(x) - \frac{\varphi_i(x)}{\psi_i(x)}\psi_i(x)\Big),$$

$$\sum_{i\in I} \lambda_i^* \leq \sum_{i\in I} \lambda_i.$$

Откуда, учитывая формулы определения параметров λ_i^* и λ_i, получим неравенство $f(x^*) \leq f(x)$ для любого $x \in S$.

Таким образом решение (x^*, λ^*) является оптимальным для задачи (6.61)–(6.64), и одновременно с этим x^* является оптимальным решением задачи (6.59), (6.60). Теорема доказана.

Рассмотрим теперь задачу (6.68), т.е. при фиксированных значениях двойственных переменных $u = \overline{u}$ требуется решить следующую задачу:

$$\sum_{i\in I} \lambda_i + \sum_{i\in I} \overline{u}_i(\varphi_i(x) - \lambda_i\psi_i(x)) \to \min, \tag{6.69}$$

$$h_k(x) \leq 0 \quad (k = \overline{1,p}), \tag{6.70}$$

$$\lambda_i \geq 0, \quad i \in I. \tag{6.71}$$

При фиксированных $\lambda_i \geq 0$ задача (6.69)–(6.71) является задачей выпуклого программирования. Поэтому решение задачи (6.69)–(6.71) сводится к нахождению такого решения $x^* \in S$, которое в паре с некоторым вектором λ^* параметров $\lambda_i^* \geq 0$ дают минимальное значение функционала (6.69). Для решения задачи (6.69)–(6.71) может быть использован некоторый приближенный алгоритм, который при оптимальных значениях двойственных переменных $u = u^*$ задачи (6.67) обеспечил бы получение оптимального решения задачи (6.69)–(6.71). Такой алгоритм может быть построен, если при очередном решении задачи (6.69)–(6.71) фиксировать значения параметров λ_i по формуле

$$\lambda_i = \lambda_i(x^*(u^{r-1})) = \frac{\varphi_i(x^*(u^{r-1}))}{\psi_i(x^*(u^{r-1}))}, \quad i \in I,$$

где $x^*(u^{r-1}) \in S$ – решение задачи (6.68), найденное на предыдущем шаге субградиентного метода.

Пусть перед выполнением очередного r-го шага субградиентного метода имеем значения переменных $u = u^r$ и предыдущее решение $(x^*(u^{r-1}), \lambda^*(u^{r-1}))$ задачи (6.69)–(6.71). Тогда задачу (6.69)–(6.71) на r-м шаге субградиентного метода решаем в следующем порядке:

- фиксируем значения переменных λ'_i по формуле

$$\lambda'_i = \lambda^*_i(x^*(u^{r-1})) = \frac{\varphi_i(x^*(u^{r-1}))}{\psi_i(x^*(u^{r-1}))}, \quad i \in I;$$

- находим оптимальное решение $x^*(u^r)$ задачи выпуклого программирования

$$\sum_{i \in I} u_i^r (\varphi_i(x) - \lambda'_i \psi_i(x)) \to \min, \tag{6.72}$$

$$h_k(x) \leq 0 \quad (k = \overline{1,p}); \tag{6.73}$$

- находим новые значения параметров λ_i по формуле

$$\lambda^*_i(x^*(u^r)) = \frac{\varphi_i(x^*(u^r))}{\psi_i(x^*(u^r))}, \quad i \in I.$$

Следует отметить, что задача (6.72), (6.73) решается на каждом шаге субградиентного метода с незначительными изменениями коэффициентов u_i^r и λ'_i. Поэтому в методе решения задачи выпуклого программирования в качестве исходной точки на каждом шаге субградиентного метода следует взять полученное на предыдущем шаге решение $x^*(u^{r-1})$ задачи (6.72), (6.73).

Пусть на последнем r-м шаге субградиентного метода $u^* = u^r$ является оптимальным решением задачи (6.67). Находим x^*– оптимальное решение задачи (6.72), (6.73) при фиксированных $\lambda'_i = \dfrac{\varphi_i(x^*(u^r))}{\psi_i(x^*(u^r))}$ и полагаем $\lambda^*_i = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$. Тогда (x^*, λ^*) будет оптимальным решением задачи (6.61)–(6.64), а x^* – оптимальным решением задачи (6.59), (6.60).

6. Задача минимизации суммы дробно-линейных функций. Рассмотрим теперь случай, когда функции

$$\{\varphi_i(x),\, i \in I\};\ \{\psi_i(x),\, i \in I\}\ \text{ и }\ \{h_k(x)\ \ (k = \overline{1,p})\}$$

являются линейными. Тогда получим следующую обобщенную дробно-линейную задачу в виде минимизации суммы дробно-линейных функций при линейных ограничениях:

$$\min_{x \in S} \left[f(x) = \sum_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} \right], \tag{6.74}$$

в которой

$$x = (x_1, x_2, \ldots, x_n) \in R^n, \quad I = \{1, 2, \ldots, m\},$$

$$\varphi_i(x) = \sum_{j=1}^{n} c_{ij} x_j + c_i^0, \quad \psi_i(x) = \sum_{j=1}^{n} d_{ij} x_j + d_i^0, \quad i \in I,$$

а $S = \left\{ x : \sum_{j=1}^{n} a_{kj} x_j \leq b_k \quad (k = \overline{1,p}); \quad x_j \geq 0 \quad (j = \overline{1,n}) \right\}$ является выпуклым многогранным множеством.

Задача (6.74) является задачей обобщенного дробно-линейного программирования в виде задаче минимизации суммы дробно-линейных функционалов. Задача (6.74) рассмотрена в работах [101, 135, 136, 186, 250, 305, 306, 313, 323, 325, 533, 555, 585], для решения которых предлагаются параметрические методы, методы ветвей и границ, методы линеаризации и модификации симплекс-метода.

Также рассмотрены обобщенные задачи в виде суммы линейных и дробно-линейных функций. Одно из первых обобщений задач линейного и дробно-линейного программирования рассмотрено в работах [9, 10, 101, 248, 249, 355, 431, 700], в которых такое обобщение приводится в виде суммы линейной и дробно-линейной функций, т.е. в виде

$$f(x) = \sum_{j=1}^{n} p_j x_j + \frac{\sum\limits_{j=1}^{n} c_j x_j + c_0}{\sum\limits_{j=1}^{n} d_j x_j + d_0}.$$

Для такой задачи рассмотрены условия квазивыпуклости функции $f(x)$ [431, 700], двойственные задачи [109, 476], параметрический метод [101] и модификации симплекс-метода [700]. В работах [667, 671] рассмотрены случаи, когда такие задачи имеют блочно-диагональную структуру или являются задачами транспортного типа. В работах [102, 631] рассмотрен случай задачи с функционалом в виде суммы двух дробно-линейных функций, в которых она сводится к задаче линейного и дробно-линейного программирования.

Для задачи (6.74) также может быть построена эквивалентная ей задача билинейного программирования следующего вида

$$\sum_{i=1}^{m} \lambda_i \to \min,$$

$$\sum_{j=1}^{n} (c_{ij} - \lambda_i d_{ij}) x_j + c_i^0 - \lambda_i d_i^0 = 0 \quad (i = \overline{1, m}),$$

$$\sum_{j=1}^{n} a_{kj} x_j \leq b_k \quad (k = \overline{1, p}),$$

$$x_j \geq 0 \quad (j = \overline{1, n}),$$

$$\lambda_i \geq 0 \quad (i = \overline{1, m}).$$

Данная задача имеет по отношению к исходной m дополнительных равенств и m дополнительных переменных, что существенно изменяет структуру исходных ограничений, которые формируют многогранное множество S. Этот факт существенно влияет на методы решения, когда исходные ограничения имеют специальную структуру.

Предположим, что $\varphi_i(x) \geq 0$, а $\psi_i(x) > 0$, $\quad i \in I$ для любого $x \in S$. Тогда, в общем случае, для решения задачи (6.74) могут быть использованы рассмотренные ранее методы для нелинейного случая, с учетом линейности рассматриваемых функций (параметрический метод, алгоритм частичной линеаризации и схема декомпозиции по ограничениям с применением субградиентного метода).

А. *Параметрический метод.* В параметрическом методе задача (6.74) сводится к решению следующей задачи многопараметрического линейного программирования:

$$F(\lambda) = \min_{x \in S} \Big[Z(x, \lambda) = \sum_{i \in I} (\varphi_i(x) - \lambda_i \psi_i(x)) \Big]. \tag{6.75}$$

В алгоритме параметрического метода необходимо выполнить следующие шаги.

Шаг 0. Найти допустимую точку $x^0 \in S$. Положить $\lambda_i^0 = \dfrac{\varphi_i(x^0)}{\psi_i(x^0)}$, $i \in I$.

Шаг r. Пусть $x^{r-1} \in S$ решение задачи (6.75), полученное на предыдущем шаге при фиксированных значениях параметров λ_i^{r-1}.

Тогда следует:

- найти оптимальное решение x^r задачи (6.75) при $\lambda_i^{r-1} = \dfrac{\varphi_i(x^{r-1})}{\psi_i(x^{r-1})}, \quad i \in I$;

- положить $\lambda_i^r = \dfrac{\varphi_i(x^r)}{\psi_i(x^r)}, \quad i \in I$;

- если $|\lambda_i^{r-1} - \lambda_i^r| \le \delta_i, \ i \in I$, где $\delta_i \ge 0$, достаточно малые числа, то взять x^r в качестве решения задачи (6.74). Останов.

При $\lambda_i = \lambda_i^{r-1}, \ i \in I$ задача (6.75) сводится к задаче линейного программирования следующего вида:

$$\sum_{j=1}^{n} q_j x_j \to \min, \tag{6.76}$$

$$x \in S, \tag{6.77}$$

в которой $q_j = \sum\limits_{i=1}^{m} (c_{ij} - \lambda_i^{r-1} d_{ij}) \ \ (j = \overline{1,n})$.

В соответствии с теорией параметрического программирования, если задача (6.76), (6.77) будет иметь последовательно одно и тоже оптимальное решение в течение двух итераций параметрического метода, то получим $|\lambda_i^r - \lambda_i^{r-1}| \le \delta_i$, и тем самым получим решение уравнения $F(\lambda) = 0$. Действительно, если $x^r = x^{r-1}$, то имеем $\lambda_i^r = \dfrac{\varphi_i(x^r)}{\psi_i(x^r)} = \dfrac{\varphi_i(x^{r-1})}{\psi_i(x^{r-1})} = \lambda_i^{r-1}$. Тогда получим, что

$$F(\lambda) = \sum_{i \in I} (\varphi_i(x^r) - \lambda_i^{r-1} \psi_i(x^r)) = \sum_{i \in I} \left(\varphi_i(x^r) - \frac{\varphi_i(x^r)}{\psi_i(x^r)} \cdot \psi_i(x^r) \right) = 0.$$

Б. *Метод частичной линеаризации.* Задача (6.74) переписывается в следующей эквивалентной форме:

$$\sum_{i=1}^{m} \lambda_i \to \min, \tag{6.78}$$

$$\sum_{j=1}^{n} (c_{ij} - \lambda_i d_{ij}) x_j - d_i^0 \lambda_i = -c_i^0 \quad (i = \overline{1,m}), \tag{6.79}$$

$$x \in S, \ \lambda_i \ge 0, \quad i \in I, \tag{6.80}$$

которая является билинейной задачей из-за произведения $\lambda_i \cdot x_j$ переменных λ_i и x_j. Для ее решения используется итеративный алгоритм с применением процедуры частичной линеаризации на каждом его шаге.

Алгоритм решения задачи (6.74) по такому алгоритму содержит следующие шаги.

Шаг 0. Пусть $x^0 \in S$ и $\lambda_i^0 = \dfrac{\varphi_i(x^0)}{\psi_i(x^0)}$, $i \in I$. Тогда точка (x^0, λ^0) является допустимым решением задачи (6.74).

Шаг r. Пусть имеем допустимую точку $x^{r-1} \in S$, найденную на предыдущем шаге, определяем значения $\lambda_i^{r-1} = \dfrac{\varphi_i(x^{r-1})}{\psi_i(x^{r-1})}$, $i \in I$. Тогда:

- находим оптимальное решение (x^r, λ^r) полученной в результате частичной линеаризации в точке (x^{r-1}, λ^{r-1}) задачи линейного программирования:

$$\sum_{i=1}^{m} \lambda_i \to \min, \tag{6.81}$$

$$\sum_{j=1}^{n} q_{ij} x_j - q_i \lambda_i = w_i \quad (i = \overline{1,m}), \tag{6.82}$$

$$x \in S,\ \lambda_i \geq 0, \quad i \in I, \tag{6.83}$$

в которой

$$q_{ij} = c_{ij} - \lambda_i^{r-1} d_{ij} \quad (i = \overline{1,m};\ j = \overline{1,n}),$$

$$q_i = \sum_{j=1}^{n} d_{ij} x_j^{r-1} + d_i^0 \quad (i = \overline{1,m}),$$

$$w_i = -\lambda_i^{r-1} \sum_{j=1}^{n} d_{ij} x_j^{r-1} - (c_i^0 - \lambda_i^{r-1} d_i^0) \quad (i = \overline{1,m});$$

- если $|\lambda_i^{r-1} - \lambda_i^r| \leq \delta_i$, $i \in I$, то x^r является оптимальным решением задачи (6.74). Останов. В противном случае

- находим $\lambda_i^r = \dfrac{\varphi_i(x^r)}{\psi_i(x^r)}$, $i \in I$ и выполняем очередной $(r+1)$-й шаг алгоритма.

В алгоритмах **А** и **Б** решается последовательность линейных задач (6.76)–(6.78) и (6.81)–(6.83), в которых к первоначальным ограничениям прибавляются еще m ограничений, которые меняются на каждом шаге алгоритма. Наличие таких ограничений не позволяет применить для их решения специальные алгоритмы, когда первоначальные ограничения задачи (6.74) имеют специальную структуру (например, блочно-диагональную или транспортного типа).

В. *Схема декомпозиции по ограничениям.* Для решения задачи (6.74) специальной структуры или большой размерности можно применить алгоритм, основанный на схеме декомпозиции по ограничениям и субградиентном методе недифференцируемой оптимизации.

Для задачи (6.74) построим функцию Лагранжа по формуле

$$L(x, \lambda, u) = \sum_{i=1}^{m} \lambda_i + \sum_{i=1}^{m} u_i \left(\sum_{j=1}^{n} (c_{ij} - \lambda_i d_{ij}) x_j + c_i^0 - \lambda_i d_i^0 \right), \tag{6.84}$$

в которой $u = \{u_i \ \ i \in I\}$ – множители Лагранжа.

Тогда задача (6.74) сводится к решению задач

$$\max_u L^*(u), \tag{6.85}$$

$$L^*(u) = \min_{\lambda \geq 0} \min_{x \in S} L(x, \lambda, u). \tag{6.86}$$

Функция $L^*(u)$ определена для любого u и является кусочно-линейной, вогнутой и недифференцируемой. Для ее максимизации используется один из алгоритмов субградиентного метода, на каждом шаге которого решается задача (6.86) при фиксированных значениях переменных u. Тогда на шаге $r + 1$ субградиентного метода необходимо выполнить следующие три основных этапа.

1. Решить задачу (6.76) при $u = u^r$ и найти оптимальное решение $(x^*(u^r), \lambda^*(u^r))$.

2. Определить значения обобщенного градиента функции $L^*(u)$ в точке $u = u^r$ по формуле

$$g_i(u^r) = \sum_{j=1}^{n} (c_{ij} - \lambda_i^*(u^r) d_{ij}) x_j^*(u^r) + c_i^0 - \lambda_i^*(u^r) d_i^0 \quad (i = \overline{1, m}).$$

3. Найти новые значения переменных u по формуле

$$u_i^{r+1} = u_i^r + h_{r+1} g_i(u^r) \quad (i = \overline{1, m}),$$

где h_{r+1} – величина шага.

Если u^* – оптимальное решение задачи (6.82), то (x^*, λ^*) – решение задачи (6.86) при $u = u^*$. Тогда (x^*, λ^*) являются оптимальным решением задачи (6.79)–(6.81), а x^* – задачи (6.74).

Г. *Алгоритм решения задачи* (6.86). На каждом шаге субградиентного метода при фиксированных двойственных переменных $u = u^r$ решается следующая задача:

$$\min_{x \in S} \min_{\lambda \geq 0} \left[\sum_{i=1}^{m} \lambda_i + \sum_{i=1}^{m} u_i^r \left(\sum_{j=1}^{n} (c_{ij} - \lambda_i d_{ij}) x_j + c_i^0 - \lambda_i d_i^0 \right) \right], \qquad (6.87)$$

которая является задачей билинейного программирования из-за произведений $\lambda_i \cdot x_j$.

При фиксированных $\lambda_i \geq 0 \quad (i = \overline{1,m})$ задача (6.87) является задачей линейного программирования. Поэтому процесс решения данной задачи сводится к нахождению такого решения $x^* \in S$ и таких значений $\lambda_i^* \geq 0 \quad (i = \overline{1,m})$, которые в совокупности представляли бы оптимальное решение (x^*, λ^*) задачи (6.87).

Для решения задачи (6.87) может быть использован приближенный алгоритм, в результате чего при фиксированных значениях переменных $u = u^*$ задачи (6.85) получим оптимальное решение задачи (6.87). Такой алгоритм может быть построен, если для решения задачи (6.87) будем фиксировать значения параметров λ_i по следующей формуле:

$$\lambda_i^* = \frac{\varphi_i(x^*(u^*))}{\psi_i(x^*(u^*))}, \quad i \in I,$$

в которой $x^*(u^*) \in S$ – решение задачи (6.87).

Допустим, что после выполнения r шагов субградиентного метода имеем значение переменных $u = u^r$. Тогда задача (6.87) на очередном шаге субградиентного метода решается в следующем порядке:

- фиксируем значения λ_i по формуле

$$\lambda_i' = \begin{cases} 0, & \text{для} \quad r = 1; \\ \lambda_i^*(u^r), & \text{для} \quad r \geq 2, \quad i \in I; \end{cases}$$

- находим оптимальное решение $x^*(u^r)$ задачи линейного программирования

$$\min_{x \in S} \sum_{j=1}^{n} q_j x_j \qquad (6.88)$$

в которой $q_j = \sum_{i \in I} u_i^r [c_{ij} - \lambda_i' d_{ij}] \quad (j = \overline{1,n})$;

- находим новое значение для λ_i по формуле

$$\lambda_i^*(u^r) = \frac{\varphi_i(x^*(u^r))}{\psi_i(x^*(u^r))}, \quad i \in I.$$

Пара полученных значений $(x^*(u^r), \lambda^*(u^r))$ будет приближенным решением задачи (6.87).

В предложенном алгоритме на каждом шаге решается задача линейного программирования (6.88), в которой меняются только значения коэффициентов целевой функции, а первоначальные ограничения остаются без изменений. Как уже отмечалось выше, такое обстоятельство имеет очень важное значение при решении задачи (6.74) с ограничениями специальной структуры.

6.4. Задача мультипликативного дробного программирования

Рассмотрим другой класс обобщенных задач дробного программирования, в который функционал представляет собой произведение дробных функций, а именно:

$$f(x) = \prod_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} \to \min, \tag{6.89}$$

$$h_k(x) \leq 0 \quad (k = \overline{1,p}), \tag{6.90}$$

где функции $\{\varphi_i(x),\ i \in I\}$, $\{-\psi_i(x),\ i \in I\}$ и $\{h_k(x),\ \ k = \overline{1,p}\}$ – непрерывны, выпуклы и дифференцируемы на R^n. Пусть ограничения (6.90) формируют выпуклое множество $S = \{x\colon h_k(x) \leq 0,\ \ k = \overline{1,p}\}$ и предположим, что $\varphi_i(x) > 0,\ \ i \in I$ и $\psi_i(x) > 0,\ \ i \in I$ для любого $x \in S$. Задачу (6.89), (6.90) назовем задачу мультипликативного дробно-выпуклого программирования.

Если в дробных мультипликативных задачах (6.89), (6.90) имеем, что все $\psi_i(x) \equiv 1,\ \ i \in I$ для любого $x \in S$, то получим задачу мультипликативной выпуклой оптимизации, которая является задачей глобальной оптимизации. Тогда данная задача имеет вид $\min\limits_{x \in S} \prod\limits_{i \in I} \varphi_i(x)$, для решения которой предложены различные методы глобальной оптимизации. Среди них можно выделить различные модификации метода ветвей и границ [65] и другие приближенные и эвристические методы [72, 310, 427, 428] глобальной оптимизации.

В случае, когда функции $\varphi_i(x)\ \ i \in I$ и $h_k(x)\ \ (k = \overline{1,p})$ являются линейными, имеем задачу линейной мультипликативной оптимизации, которая рассмотрена в работах [216, 309, 324, 361] и для решения которой предложены аналогичные методы глобальной оптимизации. Для случая, когда $m = 2$, мультипликативные линейные задачи могут быть сведены к задачам квадратичной оптимизации.

Также могут быть рассмотрены и смешанные обобщенные дробно-выпуклые задачи в виде суммы выпуклых функций плюс произведения дробно-выпуклых функций, или наоборот суммы дробно-выпуклых функций плюс произведения выпуклых функций. Функция $f(x)$ в таких задачах может иметь одну из представленных ниже структур, в которых входят:

- смешанные функционалы в виде сложения суммы и произведения выпуклых или дробно-выпуклых функций

$$f(x) = \sum_{i \in I} \varphi_i(x) + \prod_{j \in J} \frac{\alpha_j(x)}{\beta_j(x)},$$

$$f(x) = \sum_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} + \prod_{j \in J} \alpha_j(x),$$

- смешанные функционалы в виде сложения произведения выпуклых и дробно-выпуклых функций

$$f(x) = \prod_{i \in I_1} \varphi_i(x) + \prod_{j \in J} \frac{\alpha_j(x)}{\beta_j(x)},$$

в которых $\{\varphi_i(x), \quad i \in I\}$, $\{-\psi_i(x), \quad i \in I\}$, $\{\alpha_j(x), \quad j \in J\}$ и $\{-\beta_j(x), \quad j \in J\}$ – непрерывные, выпуклые и дифференцируемые на R^n, а I и J – множества соответствующих индексов.

Такие функции также могут быть определены и для разных групп переменных, например x и y. Тогда могут быть рассмотрены следующие смешанные обобщенные задачи дробно-выпуклого и выпуклого программирования

$$f(x) = f_1(x) + f_2(y) \to \min,$$

$$h_k(x) + g_k(y) \leq 0 \quad (k = \overline{1, p}),$$

в которых $f_1(x)$ – произведение или сумма дробно-выпуклых функций, $f_2(y)$ – произведение или сумма выпуклых функций, а $h_k(x)$ и $g_k(y)$ – выпуклые функции, $k = \overline{1, p}$.

В общем случае задача (6.89), (6.90) является многоэкстремальной и для ее решения в работах [251, 273, 308] приводятся алгоритмы глобальной оптимизации. Однако, по сравнению с предыдущими задачами обобщенного дробного программирования, данный класс задач дробной оптимизации, является наиболее сложным, так как с ростом количества функций, участвующих в формировании функционала (6.89), растет общая его степень нелинейности. Поэтому для решения такого класса задач не удалось найти особые методы их решения, отличные от общих методов глобальной оптимизации. Одновременно с нелинейными и линейными задачами мультипликативного программирования, рассмотренные в работах [65, 72, 216, 309, 310, 427, 428], аналогичные алгоритмы предлагаются и для мультипликативных дробных задач [251, 308]. В данных работах для решения мультипликативных дробных задач оптимизации предложены конический метод, метод ветвей и границ и другие приближенные методы и алгоритмы.

Задача мультипликативной дробной оптимизации может быть рассмотрена в виде общей задачи дробного программирования с высокой степенью нелинейности, а именно минимизировать на S дробный функционал

$$f(x) = \prod_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} = \frac{\prod\limits_{i \in I} \varphi_i(x)}{\prod\limits_{i \in I} \psi_i(x)} = \frac{\varphi(x)}{\psi(x)},$$

для решения которой можно использовать общий нелинейный параметрический метод [172, 260], в которых функции $\varphi(x)$ и $\psi(x)$ рассматриваются как общие нелинейные функции. Тогда имеем параметрическую задачу

$$F(v) = \min_{x \in S} \Big\{ Q(x, v) = \varphi(x) - v\psi(x) = \prod_{i \in I} \varphi_i(x) - v \prod_{i \in I} \psi_i(x) \Big\}.$$

Для задачи (6.89), (6.90) также может быть рассмотрена эквивалентная к ней параметрическая задача:

$$\prod_{i \in I} \lambda_i \to \min; \tag{6.91}$$

$$\varphi_i(x) - \lambda_i \psi_i(x) = 0, \quad i \in I; \tag{6.92}$$

$$h_k(x) \le 0 \quad (k = \overline{1, p}); \tag{6.93}$$

$$\lambda_i > 0, \quad i \in I. \tag{6.94}$$

Пусть $\lambda = \{\lambda_i > 0, \quad i \in I\}$ вектор параметров, которые согласно (6.92) определяются по формуле

$$\lambda_i = \frac{\varphi_i(x)}{\psi_i(x)}, \quad i \in I.$$

Целевые функции в таких задачах могут быть как дробно-линейными, так и дробно-выпуклыми, а функции $h_k(x)$ – линейными или выпуклыми.

Для задач (6.89), (6.90) и (6.91)–(6.94) имеют место следующие утверждения, показывающие на их эквивалентность, которые в дальнейшем используются для построения алгоритмов решения исходной задачи (6.89), (6.90).

Теорема 6.10. *Если* x^* *– оптимальное решение задачи* (6.89), (6.90)*, то существует такие* $\lambda_i^* = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$*, что* (x^*, λ^*) *является оптимальным решением задачи* (6.91)–(6.94).

Доказательство. Пусть x^* – оптимальное решение задачи (6.89), (6.90). Тогда имеем

$$\prod_{i\in I} \frac{\varphi_i(x^*)}{\psi_i(x^*)} \le \prod_{i\in I} \frac{\varphi_i(x)}{\psi_i(x)}$$

для любого $x \in S$. Если учесть, что для любого $i \in I$ параметры λ_i^* и λ_i определяются однозначно по формулам

$$\lambda_i^* = \frac{\varphi_i(x^*)}{\psi_i(x^*)} \quad \text{и} \quad \lambda_i = \frac{\varphi_i(x)}{\psi_i(x)},$$

то имеет место неравенство

$$\prod_{i\in I} \lambda_i^* \le \prod_{i\in I} \lambda_i, \tag{6.95}$$

а пара (x^*, λ^*) удовлетворяет равенству (6.92), т.е. имеем $\varphi_i(x^*) - -\lambda_i^*\psi_i(x^*) = \varphi_i(x^*) - \dfrac{\varphi_i(x^*)}{\psi_i(x^*)} \cdot \psi_i(x^*) = 0$, т.е. (x^*, λ^*) является допустимым решением задачи (6.91)–(6.94). Так как неравенство (6.95) выполняется для любого $x \in S$ и для соответствующих параметров λ_i, определенных из равенства (6.92), то пара (x^*, λ^*) является и оптимальным решением задачи (6.91)–(6.94). Теорема доказана.

Теорема 6.11. *Если* (x^*, λ^*) *– оптимальное решение задачи* (6.91)–(6.94)*, то* x^* *– оптимальное решение задачи* (6.89), (6.90).

Доказательство. Пусть (x^*, λ^*) – оптимальное решение задачи (6.91)–(6.94). Тогда $x^* \in S$ является допустимым планом задачи (6.89), (6.90). Так как x^* удовлетворяет равенствам (6.92), то имеем $\lambda_i^* =$

$= \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$ и $\lambda_i = \dfrac{\varphi_i(x)}{\psi_i(x)}$, которые удовлетворяют равенствам (6.92) при любом значении $x \in S$. Тогда из оптимальности пары (x^*, λ^*) получаем неравенство

$$\prod_{i\in I} \lambda_i^* \leq \prod_{i\in I} \lambda_i,$$

для любого $x \in S$.

Таким образом имеют место неравенства:

$$\prod_{i\in I} \frac{\varphi_i(x^*)}{\psi_i(x^*)} \leq \prod_{i\in I} \frac{\varphi_i(x)}{\psi_i(x)}$$

для любого $x \in S$, т.е. $f(x^*) \leq f(x)$. Следовательно, x^* является оптимальным решением задачи (6.89), (6.90). Теорема доказана.

Заметим, что задача (6.91)–(6.94) может быть сформулирована следующим образом: найти те значения переменных $x \in S$, для которых произведение переменных λ_i, определенные из равенства (6.92) принимает минимальное значение. Другими словами, необходимо найти значение x^* переменных $x \in S$, для которых параметры λ_i^* определенные по формуле

$$\lambda_i^* = \frac{\varphi_i(x^*)}{\psi_i(x^*)}, \quad i \in I$$

обеспечат минимальное значение функционала (6.91).

В данной монографии для задачи (6.89), (6.90) рассматривается функция Лагранжа и ее двойственные аспекты, приводятся параметрические методы, субградиентные методы решения прямой или двойственной задачи, а также схемы декомпозиции по ограничениям в сочетании с параметрическим методом и методом недифференцируемой оптимизации. Эти методы ниже конкретизируются для решения задач оптимизации произведения дробно-выпуклых и дробно-линейных функций.

1. Двойственная задача. Если рассмотреть задачу (6.89), (6.90) как общую задачу математического программирования, то один из вариантов двойственной задачи может быть построен на базе функции Лагранжа, которая в данном случае имеет вид

$$L(x,z) = \prod_{i\in I} \frac{\varphi_i(x)}{\psi_i(x)} + \sum_{k=1}^{p} z_k h_k(x),$$

которая представляет собой функцию, полученную в результате сложения суммы выпуклых и произведения дробно-выпуклых функций, что говорит о ее невыпуклости по переменными x и ее глобальном характере с точки зрения оптимизации.

Для анализа двойственности задачи (6.89), (6.90) рассматривается задача нахождения седловой точки функции Лагранжа в виде $\max_z \min_x L(x,z)$. Если ввести новую функцию $L^*(z) = \min_x L(x,z)$, то получим задачу $\max_z L^*(z)$, которую и берем в качестве двойственной для задачи (6.89), (6.90).

Естественно, что для полученных задач при некоторых z и любых $x \in S$ имеет место неравенство $L^*(z) \leq f(x)$, а для оптимального $x^* \in S$ найдутся такие z^*, для которых $L^*(z^*) = f(x^*)$. Такая точка (x^*, z^*) является седловой точкой функции Лагранжа $L(x,z)$ для задачи (6.89), (6.90). Однако, рассмотрение такого класса двойственных задач не выводит нас из общего свойства глобальной оптимизации для задачи (6.89), (6.90).

Далее будут предложены процедуры применения субградиентных методов нахождения седловой точки функции Лагранжа с учетом параметрического метода.

Для начала рассмотрим параметрические методы и алгоритм частичной линеаризации для решения задачи (6.89), (6.90).

2. Нелинейный однопараметрический метод. Для решения задачи (6.89), (6.90) рассматривается параметрическая задача с одним параметром

$$F(v) = \min_{x\in S}\Big[Q(x,v) = \varphi(x) - v\psi(x) = \prod_{i\in I}\varphi_i(x) - v\prod_{i\in I}\psi_i(x)\Big].$$

Тогда в соответствии с параметрическим методом необходимо найти корень уравнения $F(v) = 0$. Для этого выполняем следующие шаги.

Шаг 0. Берем $v = 0$ и находим решение x_0 задачи мультипликативного выпуклого программирования $\min_{x\in S}\varphi(x) = \min_{x\in S}\prod_{i\in I}\varphi_i(x)$ и полагаем $v_0 = \varphi(x_0) > 0$.

Шаг r. Пусть имеем v_{r-1} – значение параметра v, найденное на предыдущем шаге.

Тогда:

- находим решение x_r задачи мультипликативного выпуклого программирования $\min_{x\in S} Q(x,v)$ при фиксированном значение $v = v_{r-1} > 0$, т.е. решаем задачу

$$\min_{x\in S}\Big[\prod_{i\in I}\varphi_i(x) - v_{r-1}\prod_{i\in I}\psi_i(x)\Big];$$

- полагаем $v_r = f(x_r) > 0$;

- если $|F(v_r)| \leq \delta$, то v_r является корнем уравнения $F(v) = 0$ с заданной точностью, определяемой $\delta \geq 0$. Тогда x_r является решение исходной задачи мультипликативного дробного программирования (6.89), (6.90).

3. Нелинейный многопараметрический метод. Другая параметрическая функция может быть построена, если ввести параметр λ_i для каждой пары функций $\varphi_i(x)$ и $\psi_i(x)$, связанных между собой выражением $\lambda_i = \dfrac{\varphi_i(x)}{\psi_i(x)}$, $i \in I$. Тогда получим следующую многопараметрическую задачу

$$F(\lambda) = \min_{x \in S} \left[Z(x, \lambda) = \prod_{i \in I} \bigl(\varphi_i(x) - \lambda_i \psi_i(x)\bigr) = \prod_{i \in I} z_i(x, \lambda_i) \right],$$

в которой $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ – вектор параметров $\lambda_i > 0$, $i \in I$, а $z_i(x, \lambda_i) = \varphi_i(x) - \lambda_i \psi_i(x)$, $i \in I$.

В нелинейном многопараметрическом методе решения задачи $\min\limits_{x \in S} Z(x, \lambda)$ и нахождения корня уравнения $F(\lambda) = 0$ может быть проведено по правилу покоординатного фиксирования на каждом его шаге параметра λ по формуле $\lambda_i = \dfrac{\varphi_i(x)}{\psi_i(x)}$, $i \in I$. Тогда, если использовать некоторый итерационный процесс, то полученная последовательность оптимальных решений x_r^*, $r = 1, 2, \ldots$ задачи $\min\limits_{x \in S} Z(x, \lambda_{r-1})$, при $\lambda_i^r = \dfrac{\varphi_i(x_{r-1}^*)}{\psi_i(x_{r-1}^*)}$, $i \in I$ будет стремиться к оптимальному решению x^* для которого будет выполняться неравенства

$$\bigl|\varphi_i(x^*) - \lambda_i^{r-1} \psi_i(x^*)\bigr| =$$

$$= \left| \left(\frac{\varphi_i(x^*)}{\psi_i(x^*)} - \lambda_i^{r-1} \right) \psi_i(x^*) \right| = \bigl|\lambda_i^* - \lambda_i^{r-1}\bigr| \psi_i(x^*) \leq \delta_i \psi_i(x^*), \quad i \in I,$$

т.е. $|\lambda_i^* - \lambda_i^{r-1}| \leq \delta_i$, $i \in I$, где $\lambda_i^* = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$, $i \in I$, а δ_i – достаточно малые неотрицательные числа. В результате чего

$$|F(\lambda^{r-1})| = |Z(x^*, \lambda^{r-1})| \leq \prod_{i \in I} \bigl|\lambda_i^* - \lambda^{r-1}\bigr| \psi_i(x^*) \leq \delta,$$

где δ – достаточно малое неотрицательное число.

По такому процессу будем искать два последовательные решения x^*_{r-1} и x^*_r задачи $\min\limits_{x\in S} Z(x,\lambda)$ при фиксированном значении параметра λ, устойчивые не в целом по функции $f(x)$, а по каждой дробной функции $f_i(x) = \dfrac{\varphi_i(x)}{\psi_i(x)}$, $i \in I$. В таком случае решение x^*_r может быть взято в качестве приближенного решения задачи произведения дробно-выпуклых функций (6.89), (6.90).

Если провести анализ устойчивости решений в многопараметрической задачи $F(\lambda) = \min\limits_{x\in S} Z(x,\lambda)$, то оптимальное решение x^* будет оптимально устойчиво, если для каждого параметра λ_i найдется такой интервал значений $[\underline{\lambda}_i, \overline{\lambda}_i]$, для которого при любых значениях $\lambda_i \in [\underline{\lambda}_i, \overline{\lambda}_i]$, $i \in I$ решение x^* остается оптимальным решением для задачи с фиксированными значениями параметров λ_i, т.е. λ_i является корнем уравнения $F_i(\lambda_i) = \min\limits_{x\in S}(\varphi_i(x) - \lambda_i\psi_i(x)) = 0$, или же для которого имеет место равенство $\varphi_i(x^*) - \lambda_i\psi_i(x^*) = 0$.

Такой анализ параметрических задач может быть использован для решения задачи $\min\limits_{x\in S} Z(x,\lambda)$ и нахождения корня уравнения $F(\lambda) = 0$, с учетом корня λ_i каждого уравнения $F_i(\lambda_i) = 0, \quad i \in I$.

Тогда в многопараметрическом методе решения задачи оптимизации произведения дробных функций (6.89), (6.90) может быть использован следующий итерационный процесс.

Шаг 0. Находим допустимую точку $x^0 \in S$ путем решения оптимизационной задачи $\min\limits_t\{t : \ \ h_k(x) \le t, \quad k = \overline{1,p}\}$. Если в полученном решение (x^0, t^0) данной задачи имеем $t^0 = 0$, то x^0 является допустимым решением, т.е. $x^0 \in S$. Если же $t^0 \neq 0$, то исходная задача (6.89), (6.90) не имеет допустимых решений.

Пусть $x^0 \in S$, тогда полагаем $\lambda_i^0 = \dfrac{\varphi_i(x^0)}{\psi_i(x^0)} \ge 0, \quad i \in I$, т.е. λ_i^0 является корнем уравнения $F_i(\lambda_i) = 0$ при $x = x^0$.

Шаг r. Пусть $x^{r-1} \in S$ – допустимое решение, полученной на предыдущем шаге, а $\lambda_i^{r-1} = \dfrac{\varphi_i(x^{r-1})}{\psi_i(x^{r-1})} \ge 0, \quad i \in I$, т.е. λ_i^{r-1} является корнем уравнения $F_i(\lambda_i^{r-1}) = 0$ при $x = x^{r-1}$.

Тогда:

- находим оптимальное решение x^r задачи выпуклого программирования

$$\min_{x\in S}\prod_{i\in I}\left(\varphi_i(x) - \lambda_i^{r-1}\psi_i(x)\right);$$

- полагаем $\lambda_i^r = \dfrac{\varphi_i(x^r)}{\psi_i(x^r)} \geq 0, \ \ i \in I$;

- если $|\lambda_i^r - \lambda_i^{r-1}| \leq \delta_i, \ i \in I$, то получим $|F(\lambda^r)| \leq \prod\limits_{i \in I} |F_i(\lambda_i^*)| =$ $= \prod\limits_{i \in I} \left|\lambda_i^r - \lambda_i^{r-1}\right| \psi_i(x^r) \leq \delta$, где значения $\delta_i \geq 0, \ \ i \in I$ и $\delta \geq 0$ выбираются соответствующим образом, для того чтобы обеспечить сходимость данного процесса.

Если предположить, что имеются некоторые способы корректировке параметров λ_i^r на каждом шаге итерационного процесса, то полученная последовательность точек $\{x^r\}$ сходится к оптимальному решению x^* задачи (6.89), (6.90).

4. Алгоритм метода частичной линеаризации. Для решения невыпуклой задачи (6.91)–(6.94) может быть использован метод частичной линеаризации, который сводил-бы ее к задаче оптимизации нелинейного функционала на множестве выпуклых ограничений.

Если рассмотреть функции

$$H_i(x, \lambda_i) = \varphi_i(x) - \lambda_i \psi_i(x), \ \ i \in I,$$

которые являются билинейно-выпуклыми, то можно провести частичную линеаризацию таких функций в точке (x^r, λ^r) по переменным $\lambda_i, \ i \in I$ в виде следующих выражений

$$H_i^r(x, \lambda_i) = \varphi_i(x) - \lambda_i^r \psi_i(x) - (\lambda_i - \lambda_i^r)\psi_i(x^r), \ \ i \in I.$$

Тогда вместо задачи (6.91)–(6.94) можно рассматривать задачу выпуклого программирования:

$$\prod_{i \in I} \lambda_i \to \min,$$

$$\varphi_i(x) - \lambda_i^r \psi_i(x) - \lambda_i \psi_i(x^r) + \lambda_i^r \psi_i(x^r) \leq 0, \ \ i \in I,$$

$$h_k(x) \leq 0 \quad (k = \overline{1, p}),$$

$$\lambda_i \geq 0 \quad i \in I.$$

Тогда для решения задачи (6.91)–(6.94) используем следующий алгоритм параметрического метода с выполнением операции частичной линеаризации на каждом его шаге.

Шаг 0. Пусть имеем допустимую точку $x^0 \in S$ и определяем значения параметров $\lambda_i^0 = \dfrac{\varphi_i(x^0)}{\psi_i(x^0)}, \ \ i \in I.$

Шаг r. Пусть $x^{r-1} \in S$ – допустимое решение, найденное на предыдущем шаге, а $\lambda_i^{r-1} = \dfrac{\varphi_i(x^{r-1})}{\psi_i(x^{r-1})}$, $i \in I$. Тогда:

- находим решение (x^r, λ^r) задачи выпуклого программирования

$$\prod_{i \in I} \lambda_i \to \min,$$

$$\varphi_i(x) - \lambda_i^{r-1}\psi_i(x) - \lambda_i\psi_i(x^{r-1}) \le -\lambda_i^{r-1}\psi_i(x^{r-1}), \quad i \in I,$$

$$h_k(x) \le 0 \quad (k = \overline{1,p}),$$

$$\lambda_i \ge 0 \quad i \in I;$$

- если $|\lambda_i^r - \lambda_i^{r-1}| \le \delta_i$, тогда x^r является решением задачи (6.91)–(6.94), а также и первоначальной задачи (6.89), (6.90). Останов, в противном случае

- полагаем $\lambda_i^r = \dfrac{\varphi_i(x^r)}{\psi_i(x^r)}$, $i \in I$ и выполняем очередной $(r+1)$-й шаг алгоритма.

Данный алгоритм, как и многопараметрический метод, генерирует последовательность точек (x^r, λ^r), которые стремятся к оптимальному решению задачи (6.91)–(6.94), и следовательно и к оптимальному решению исходной задачи оптимизации произведения дробно-выпуклых функций (6.89), (6.90).

5. Субградиентные методы. Как отмечено ранее, для решения задачи выпуклой оптимизации субградиентные методы могут быть использованы в двух вариантах. В первом варианте субградиентный метод применяется для оптимизации по переменным прямой задачи, а во втором варианте – по переменным двойственной задачи. Рассмотрим особенности применения субградиентных методов решения задачи произведения дробно-выпуклых функций (6.89), (6.90).

А. *Субградиентный метод решения задачи* (6.89), (6.90). Решение задачи (6.89), (6.90) субградиентным методом проводится с использованием эквивалентной параметрической задачи $F(\lambda) = \min\limits_{x \in S} Z(x,\lambda) = \min\limits_{x \in S} \prod\limits_{i \in I} (\varphi_i(x) - \lambda_i\psi_i(x))$. Так как функции $\varphi_i(x)$ и $-\psi_i(x)$ являются выпуклыми, то при $\lambda > 0$ функция $Z(x,\lambda)$ является выпуклой по x и линейной по λ. Если предположить, что функции $\varphi_i(x) > 0$, $i \in I$ и $\psi_i(x) > 0$, $i \in I$ то при выборе значений параметров по формуле $\lambda_i = \dfrac{\varphi_i(x)}{\psi_i(x)}$, $i \in I$ для любого $x \in S$, то параметры λ_i всегда будут положительными для любого $x \in S$ и тогда функция $Z(x, \lambda)$ при таком фиксировании параметров λ_i будет выпуклой по x. Таким образом, при таких предположениях для нахождения локального, и тем самым глобального, минимума функции $Z(x, \lambda)$ по переменным x при фиксированных $\lambda_i = \dfrac{\varphi_i(x)}{\psi_i(x)}$ может быть использован один из субградиентных методов, в частности метод эллипсоидов, как это было сделано для задач выпуклого и дробно-выпуклого программирования (см. п. 2.5. и 5.3.).

Рассмотрим случай задачи (6.89), (6.90), когда множество S определено ограничениями (6.2), т.е. $S = \{x \in R^n : h_k(x) \leq 0, \quad k = \overline{1,p}\}$, а $\varphi_i(x) \geq 0$, $\psi_i(x) > 0, \quad i \in I$ для любого $x \in S$. Тогда для решения параметрической задачи обобщенного дробного программирования

$$F(\lambda) = \min_{x \in S} Z(x, \lambda) = \min_{x \in S} \prod_{i \in I} \left(\varphi_i(x) - \lambda_i \psi_i(x)\right),$$

применяем метод эллипсоидов (см. п. 5.3.), который в данном случае имеет следующие особенности при вычисления значений обобщенного градиента.

Пусть на r-м шаге метода эллипсоидов имеем точку x_r. Тогда значения обобщенного градиента функции $Z(x, \lambda)$ в точке x_r определяются по формуле

$$g(x_r) = \begin{cases} g_Z(x_r) = \prod\limits_{i \in I} \left(g_{\varphi_i}(x_r) - \lambda_i^r g_{\psi_i}(x_r)\right), & \text{если} \quad h_{k_r}(x_r) \leq 0; \\ g_h(x_r) = g_{h_{k_r}}(x_r), \quad \text{если} \quad h_{k_r}(x_r) > 0, \end{cases}$$

в которой:

k_r – индекс для которого достигается поточечный максимум по функциям ограничений, а $h_{k_r}(x_r)$ – значение этого максимума в точке x_r, т.е. k_r индекс для которого $h_{k_r}(x_r) = \max\limits_{1 \leq k \leq p} h_k(x_r)$;

λ_i^r – значения параметров λ_i, которые определяются для текущей точки x_r по формуле

$$\lambda_i^r = \begin{cases} \dfrac{\varphi_i(x_r)}{\psi_i(x_r)}, & \text{если} \quad \varphi_i(x_r) > 0 \quad \text{и} \quad \psi_i(x_r) > 0; \\ 0, & \text{в противном случае,} \end{cases}$$

$g_Z(x_r)$ – значение субградиента функции $Z(x, \lambda)$ в точке x_r при параметрах $\lambda_i = \lambda_i^r, \quad i \in I$;

$g_h(x_r)$ – значение субградиента, определенное для одной из функций ограничений;

$g_{\varphi_i}(x_r)$, $g_{\psi_i}(x_r)$, $i \in I$ и $g_{h_{k_r}}(x_r)$ – значения субградиентов в точке x_r соответственно для функций $\varphi_i(x)$, $\psi_i(x)$, $i \in I$ и $h_{k_r}(x)$.

Полученное решение x^* по методу эллипсоидов для задачи

$$\min_{x \in S} Z(x, \lambda) = \min_{x \in S} \prod_{i \in I} \left(\varphi_i(x) - \lambda_i^* \psi_i(x)\right),$$

при $\lambda_i^* = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$, $i \in I$ будет оптимальным решением для исходной задачи (6.89), (6.90), а именно для задачи

$$\min_{x \in S} f(x) = \min_{x \in S} \prod_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)}.$$

Действительно, так как x^* – оптимальное решение задачи $\min\limits_{x \in S} Z(x, \lambda^*) = Z(x^*, \lambda^*)$ при $\lambda_i^* = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$, $i \in I$, получим

$$F(\lambda^*) = Z(x^*, \lambda^*) = \prod_{i \in I} \left(\varphi_i(x^*) - \lambda_i^* \psi_i(x^*)\right) =$$

$$= \prod_{i \in I} \left(\varphi_i(x^*) - \frac{\varphi_i(x^*)}{\psi_i(x^*)} \psi_i(x^*)\right) = 0.$$

Таким образом λ^* является решением уравнения $F(\lambda) = 0$. Тогда в соответствии с параметрическим методом x^* является решением задачи (6.89), (6.90).

Б. *Субградиентный метод решения двойственной задачи.* Рассмотрим особенности применения субградиентных методов при решении двойственной задачи $\max\limits_z L^*(z)$ с учетом параметрической задачи. В соответствие с двойственной лагранжевой задачей решение задачи (6.89), (6.90) сводится к нахождению седловой точке (x^*, z^*) функции Лагранжа $L(x, z)$, которая соответствует задачам:

$$L(x^*, z^*) = \max_z \min_x L(x, z) = \min_x \max_z L(x, z).$$

С учетом их разделения на подзадачи по уровням анализа имеем следующие две задачи:

- задачу внутреннего уровня $L^*(z) = \min\limits_x L(x, z)$, которая представляет собой задачу минимизации суммы дробно-выпуклых функций и суммы выпуклых функций по переменным x;

- задачу внешнего уровня $\max_z L^*(z)$, которая является двойственной задачей для (6.89), (6.90).

Так как по определению, функция $L^*(z)$ является кусочно-линейной, вогнутой и не везде дифференцируемой по переменным z, то для ее максимизации используются методы недифференцируемой оптимизации. Если для этого применить один из субградиентных методов, то на каждом его шаге, при фиксированных значениях переменных z необходимо решать задачу невыпуклого программирования $\min_x L(x,z)$, которая является многоэкстремальной задачей.

Пусть для решения задачи $\max_z L^*(z)$ используется один из субградиентных методов, на r-м шаге которого имеем текущую точку z^r. Тогда на очередном $(r+1)$-м шаге необходимо выполнить следующие три основных этапа.

1. Решить задачу $\min_x L(x,z)$ при фиксированных $z_k = z_k^r \quad (k=\overline{1,p})$, для которой функционал по переменным x имеет вид:

$$F_r(x) = \prod_{i\in I} \frac{\varphi_i(x)}{\psi_i(x)} + \sum_{k=1}^{p} z_k^r h_k(x),$$

которая, как отмечено выше, является многоэкстремальной задачей оптимизации. Для решения задачи безусловной минимизации невыпуклого функционала $F_r(x)$ можно использовать различные алгоритмы и методы, среди которых параметрический метод. Для этого рассматривается функция $Q_r(x,\lambda)$ и параметрическая задача $\beta(\lambda) = \min_x Q_r(x,\lambda)$, в которой

$$Q_r(x,\lambda) = \prod_{i\in I} (\varphi_i(x) - \lambda_i\psi_i(x)) + \sum_{k=1}^{p} z_k^r h_k(x)$$

и находится корень $\lambda(x_r, z^r)$ уравнения $\beta(\lambda) = 0$.

Таким образом, вместо задачи $\min_x F_r(x)$ решаем задачу $\min_x Q_r(x,\lambda)$ при фиксированном значение параметра $\lambda = \lambda_i^{r-1}$.

Для этого определяем значение параметра

$$\lambda_i^{r-1} = \begin{cases} \dfrac{\varphi_i(x_{r-1}^*)}{\psi_i(x_{r-1}^*)}, & i \in I, \text{ если } \varphi_i(x_{r-1}^*) > 0, \quad \psi_i(x_{r-1}^*) > 0, \\ 0, \text{ в противном случае }, \end{cases}$$

где x_{r-1}^* – найденное решение задачи $\min_x Q_r(x, \lambda^{r-1})$ на предыдущем шаге субградиентного метода.

Находим оптимальное решение $x_r^* = x(z^r)$ задачи выпуклой оптимизации $\min\limits_x Q_r(x, \lambda^{r-1})$.

2. Находим значения обобщенных градиентов функции $L^*(z)$ по переменным z в точке z^r с учетом полученного решения x_r^* и функции $Q_r(x, \lambda^{r-1})$ по формулам

$$g_{L^*}^{z_k} = g_{Q_r}^{z_k} = h_k(x_r^*) \quad (k = \overline{1,p}).$$

3. Находим значения z_i^{r+1} по формуле

$$z_k^{r+1} = \max\left\{0,\ z_k^r + \gamma_{r+1} g_{Q_r}^{z_k}\right\} \quad (k = \overline{1,p}),$$

где γ_{r+1} – величина шага в субградиентном методе.

Пусть z^* – решение задачи $\max\limits_z L^*(z)$, а x^* – решение задач $\min\limits_x L(x, z^*)$, $\min\limits_x F(x)$ или же $\min\limits_x Q(x, \lambda^*)$. Таким образом при $\lambda_i^* = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$, имеем, что точка x^* является оптимальным решением задачи $\min\limits_x Q(x, \lambda^*)$ и задачи $\min\limits_x F(x)$, т.е. имеем $Q(x^*, \lambda^*) = \min\limits_x Q(x, \lambda^*)$, для которой по теории выпуклого программирования выполняются условия $z_k^* h_k(x^*) = 0 \quad (k = \overline{1,p})$. Тогда получим что

$$Q(x^*, \lambda^*) = \min_x Q(x, \lambda^*) =$$

$$= \prod_{i\in I} (\varphi_i(x^*) - \lambda_i^* \psi_i(x^*)) = \prod_{i\in I} \left(\varphi_i(x^*) - \frac{\varphi_i(x^*)}{\psi_i(x^*)} \psi_i(x^*)\right) = 0,$$

т.е. λ^* является решением уравнения $\beta(\lambda) = 0$, и в соответствии с параметрическим методом x^* является решением задачи (6.89), (6.90).

6. Схема декомпозиции по ограничениям. Для решения задачи (6.89), (6.90) (для случая, когда ограничения задачи имеют специальную структуру) рассмотрим алгоритм, который соответствует некоторому итерационному процессу (относительно двойственных переменных u для ограничений (6.90)) недифференцируемой оптимизации.

Для задачи (6.89), (6.90) рассмотрим эквивалентную ей задачу (6.91)–(6.94) и построим алгоритм решения данной задачи, основанный на функции Лагранжа, схеме декомпозиции по ограничениям и субградиентных методах. Для этого построим функцию Лагранжа для задачи (6.91), (6.92) по формуле

$$L(x, \lambda, u) = \prod_{i\in I} \lambda_i + \sum_{i\in I} u_i(\varphi_i(x) - \lambda_i \psi_i(x)),$$

где $u = \{u_i, \quad i \in I\}$ – множители Лагранжа для ограничений (6.92), $u_i \geq 0, \quad i \in I$.

Тогда задача (6.91)–(6.94), и тем самым и (6.89), (6.90) сводится к решению задачи нахождения седловой точки функции Лагранжа $L(x, \lambda, u)$, т.е. к решению задачи

$$L(x^*, \lambda^*, u^*) = \max_u \min_{\lambda \geq 0} \min_{x \in S} L(x, \lambda, u). \tag{6.96}$$

В свою очередь задачу (6.96) можно свести к решению задачи

$$\max_u L^*(u), \tag{6.97}$$

где

$$L^*(u) = \min_{\lambda \geq 0} \min_{x \in S} L(x, \lambda, u). \tag{6.98}$$

Функция $L^*(u)$ определена для любого u и является кусочно-линейной, вогнутой и недифференцируемой. Поэтому для решения задачи (6.97) используются субградиентные методы, на каждом шаге которых необходимо решить задачу (6.98) при фиксированных значениях переменных u.

Пусть для решения задачи (6.97) используется некоторый субградиентный метод. Тогда на $(r+1)$-м $\quad (r = 0, 1, 2, \ldots)$ шаге необходимо выполнить следующие три основных этапа.

1. Решить задачу (6.98) при фиксированных $u = u^r$ и найти оптимальное решение $(x^*(u^r), \lambda^*(u^r))$.

2. Определить значения обобщенного градиента функции $L^*(u)$ в точке $u = u^r$ по формуле

$$g_i(u^r) = \varphi_i(x^*(u^r)) - \lambda_i^*(u^r)\psi_i(x^*(u^r)), \quad i \in I.$$

3. Найти новые значения u^{r+1} по формуле

$$u_i^{r+1} = u_i^r + \gamma_{r+1} g_i(u^r), \quad i \in I,$$

где γ_{r+1} – величина шага.

Теорема 6.12. *Пусть u^* – оптимальное решение задачи* (6.97), *а (x^*, λ^*) – оптимальное решение задачи* (6.98) *при фиксированных $u = u^*$. Тогда (x^*, λ^*) – оптимальное решение задачи* (6.91)–(6.94), *а x^* – оптимальное решение задачи* (6.89), (6.90).

Доказательство. Пусть u^* является оптимальным решением задачи (6.97). Тогда в соответствие с субградиентным методом имеем $g_i(u^*) = \varphi_i(x^*) - \lambda_i^* \psi_i(x^*) = 0$ или $\lambda_i^* = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$, где (x^*, λ^*) является оптимальным решением задачи (6.98) при $u = u^*$, т.е. имеем неравенство $L(x^*, \lambda^*, u^*) \leq L(x, \lambda, u^*)$, для любого $x \in S$ и $\lambda_i = \dfrac{\varphi_i(x)}{\psi_i(x)}$. Из данного неравенства получим следующие соотношения:

$$\prod_{i\in I} \lambda_i^* + \sum_{i\in I} u_i^*(\varphi_i(x^*) - \lambda_i^* \psi_i(x^*)) \leq \prod_{i\in I} \lambda_i + \sum_{i\in I} u_i^*(\varphi_i(x) - \lambda_i \psi_i(x)),$$

$$\prod_{i\in I} \lambda_i^* + \sum_{i\in I} u_i^* \left(\varphi_i(x^*) - \frac{\varphi_i(x^*)}{\psi_i(x^*)} \psi_i(x^*) \right) \leq \prod_{i\in I} \lambda_i + \sum_{i\in I} u_i^* \left(\varphi_i(x) - \frac{\varphi_i(x)}{\psi_i(x)} \psi_i(x) \right),$$

$$\prod_{i\in I} \lambda_i^* \leq \prod_{i\in I} \lambda_i, \ \ f(x^*) \leq f(x) \ \text{ для любого } \ x \in S.$$

Таким образом решение (x^*, λ^*) является оптимальным для задачи (6.91)–(6.94), и одновременно с этим x^* является оптимальным решением задачи (6.89), (6.90). Теорема доказана.

Рассмотрим теперь задачу (6.98), т.е. при фиксированных значениях двойственных переменных $u = \overline{u}$ требуется решить следующую задачу:

$$\prod_{i\in I} \lambda_i + \sum_{i\in I} \overline{u}_i(\varphi_i(x) - \lambda_i \psi_i(x)) \to \min, \tag{6.99}$$

$$h_k(x) \leq 0 \quad (k = \overline{1,p}), \tag{6.100}$$

$$\lambda_i \geq 0, \quad i \in I. \tag{6.101}$$

При фиксированных $\lambda_i \geq 0$ задача (6.99)–(6.101) является задачей выпуклого программирования. Поэтому решение задачи (6.99)–(6.101) сводится к нахождению такого решения $x^* \in S$, которое в паре с некоторым вектором λ^* параметров $\lambda_i^* \geq 0$ дают минимальное значение функционала (6.99). Для решения задачи (6.99)–(6.101) может быть использован некоторый приближенный алгоритм, который при оптимальных значениях двойственных переменных $u = u^*$ задачи (6.97) обеспечил получение оптимального решения задачи (6.99)–(6.101). Такой алгоритм может быть построен, если при очередном решении задачи (6.99)–(6.101) фиксировать значения параметров λ_i по формуле

$$\lambda_i = \lambda_i(x^*(u^{r-1})) = \frac{\varphi_i(x^*(u^{r-1}))}{\psi_i(x^*(u^{r-1}))},$$

где $x^*(u^{r-1}) \in S$ – решение задачи (6.98), найденное на предыдущем шаге субградиентного метода.

Пусть перед выполнением очередного r-го шага субградиентного метода имеем значения переменных $u = u^r$ и предыдущее решение $(x^*(u^{r-1}), \lambda^*(u^{r-1}))$ задачи (6.99)–(6.101). Тогда задача (6.99)–(6.101) на r-м шаге субградиентного метода решается в следующем порядке:

- фиксируется значения переменных λ'_i по формуле

$$\lambda'_i = \lambda^*_i(x^*(u^{r-1})) = \frac{\varphi_i(x^*(u^{r-1}))}{\psi_i(x^*(u^{r-1}))};$$

- находится оптимальное решение $x^*(u^r)$ задачи выпуклого программирования

$$\sum_{i \in I} u_i^r(\varphi_i(x) - \lambda'_i \psi_i(x)) \to \min, \tag{6.102}$$

$$h_k(x) \leq 0 \quad (k = \overline{1,p}); \tag{6.103}$$

- находятся новые значения параметров λ_i по формуле

$$\lambda^*_i(x^*(u^r)) = \frac{\varphi_i(x^*(u^r))}{\psi_i(x^*(u^r))}, \quad i \in I.$$

Пусть на последнем r-м шаге субградиентного метода $u^* = u^r$ является оптимальным решением задачи (6.97). Находим x^*– оптимальное решение задачи (6.102), (6.103) при фиксированных $\lambda'_i = \dfrac{\varphi_i(x^*(u^r))}{\psi_i(x^*(u^r))}$ и полагаем $\lambda^*_i = \dfrac{\varphi_i(x^*)}{\psi_i(x^*)}$. Тогда (x^*, λ^*) будет оптимальным решением задачи (6.91)–(6.94), а x^* – оптимальным решением задачи (6.89), (6.90).

7. Задача мультипликативного дробно-линейного программирования. Рассмотрим теперь случай, когда функции

$$\{\varphi_i(x),\, i \in I\};\ \{\psi_i(x),\, i \in I\} \ \text{ и } \ \{h_k(x) \quad (k = \overline{1,p})\}$$

являются линейными. Тогда получим следующую обобщенную дробно-линейную задачу в виде минимизации произведения дробно-линейных функций, т.е. задачу мультипликативного дробно-линейного программирования:

$$\min_{x \in S} \left[f(x) = \prod_{i \in I} \frac{\varphi_i(x)}{\psi_i(x)} \right], \tag{6.104}$$

в которой

$$x = (x_1, x_2, \dots, x_n) \in R^n, \quad I = \{1, 2, \dots, m\},$$

$$\varphi_i(x) = \sum_{j=1}^{n} c_{ij} x_j + c_i^0, \quad \psi_i(x) = \sum_{j=1}^{n} d_{ij} x_j + d_i^0,$$

а $S = \left\{x : \sum_{j=1}^{n} a_{kj} x_j \le b_k \quad (k = \overline{1,p}); \quad x_j \ge 0 \quad (j = \overline{1,n})\right\}$ является выпуклым многогранным множеством. Задача (6.104) рассмотрена в работах [169, 308, 312, 447].

Для задачи (6.104) также может быть построена эквивалентная ей задача минимизации нелинейного функционала при линейных и билинейных ограничениях следующего вида

$$\prod_{i \in I} \lambda_i \to \min,$$

$$\sum_{j=1}^{n} (c_{ij} - \lambda_i d_{ij}) x_j + c_i^0 - \lambda_i d_i^0 = 0 \quad (i = \overline{1,m}),$$

$$\sum_{j=1}^{n} a_{kj} x_j \le b_k \quad (k = \overline{1,p}),$$

$$x_j \ge 0 \quad (j = \overline{1,n}),$$

$$\lambda_i \ge 0 \quad (i = \overline{1,m}).$$

Данная задача имеет по отношению к исходной m дополнительных равенств и m дополнительных переменных, что существенно изменяет структуру исходных ограничений, которые формируют многогранное множество S. Этот факт существенно влияет на методы решения, когда исходные ограничения имеют специальную структуру.

Предположим, что $\varphi_i(x) > 0$, а $\psi_i(x) > 0$, $\quad i \in I$ для любого $x \in S$. Тогда функция $f(x)$ является квазивыпуклой и недифференцируемой на S. Для решения задачи (6.104) могут быть использованы рассмотренные ранее методы для нелинейного случая (параметрический метод, алгоритм частичной линеаризациии и схема декомпозиции по ограничениям с применением субградиентного метода).

А. *Параметрический метод.* В параметрическом методе задача (6.104) сводится к решению следующей задачи параметрического программирования:

$$F(\lambda) = \min_{x \in S} Z(x, \lambda) = \min_{x \in S} \prod_{i \in I} (\varphi_i(x) - \lambda_i \psi_i(x)), \tag{6.105}$$

которая имеет вид:

$$Z(x, \lambda) = \prod_{i \in I} \left(\sum_{j=1}^{n} q_{ij}(\lambda_i) x_j + q_i^0(\lambda_i) \right) \to \min, \tag{6.106}$$

$$\sum_{j=1}^{n} a_{kj} x_j \le b_k \quad (k = \overline{1, p}), \tag{6.107}$$

$$x_j \ge 0, \tag{6.108}$$

где

$$q_{ij}(\lambda_i) = c_{ij} - \lambda_i d_{ij},$$
$$q_i^0 = c_i^0 - \lambda_i d_i^0.$$

При фиксированных значениях параметров $\lambda_i, \;\; i \in I$ задача (6.106)–(6.108) является задачей мультипликативного линейного программирования и в целом она является задачей NP-полной и тем самым глобальной оптимизации. Для решения таких задач в работах [216, 309, 324, 361] предложены различные алгоритмы, такие как метод секущих плоскостей, эвристические методы, метод ветвей и границ, параметрический симплекс-метод.

С другой стороны, функционал $Z(x, \lambda)$ является мультипликативной функцией от n переменных, и в результате выполнения операции умножения, суммирования и приведения подобия получаем полиномиальную функцию, которую можно записать в общем виде следующим образом

$$Z(x, \lambda) = \sum_{i \in I} \beta_i(\lambda) \prod_{j=1}^{n} x_j^{\alpha_{ij}},$$

в котором $\beta_i(\lambda)$ формируют ненулевые коэффициенты мономов, а α_{ij} – целые положительные числа, которые представляют собой степени соответствующих переменных x_j входящих в i-м мономе.

В алгоритме параметрического метода необходимо выполнить следующие шаги.

Шаг 0. Найти допустимую точку $x^0 \in S$. Положить $\lambda_i^0 = \dfrac{\varphi_i(x^0)}{\psi_i(x^0)}$, $i \in I$.

Шаг r. Пусть $x^{r-1} \in S$ решение задачи (6.105), полученное на предыдущем шаге при фиксированных значениях параметров λ_i^{r-1}.

Тогда следует:

- найти оптимальное решение x^r задачи (6.105) при $\lambda_i^{r-1} = \dfrac{\varphi_i(x^{r-1})}{\psi_i(x^{r-1})}, \quad i \in I;$

- положить $\lambda_i^r = \dfrac{\varphi_i(x^r)}{\psi_i(x^r)}, \quad i \in I;$

- если $|\lambda_i^{r-1} - \lambda_i^r| \leq \delta_i,\ i \in I$, где $\delta_i \geq 0$, достаточно малые числа, то взять x^r в качестве решения задачи (6.104). Останов.

Для $\lambda = \lambda^{r-1}$ задача (6.105) сводится к задаче нелинейного программирования (6.106)–(6.108), которая может быть записана в виде общей задачей полиномиальной оптимизации

$$\min_{x \in S} \sum_{i \in I} \gamma_i \prod_{j=1}^{n} x_j^{\alpha_{ij}},$$

где $\gamma_i = \beta_i(\lambda^{r-1}) \quad i \in I$.

В соответствии с теорией параметрического программирования, если задача (6.106)–(6.108) будет иметь последовательно одно и тоже оптимальное решение в течение двух итераций параметрического метода, то получим $|\lambda_i^r - \lambda_i^{r-1}| \leq \delta_i$, и тем самым получим решение уравнения $F(\lambda) = 0$. Действительно, если $x^r = x^{r-1}$, то имеем $\lambda_i^r = \dfrac{\varphi_i(x^r)}{\psi_i(x^r)} = \dfrac{\varphi_i(x^{r-1})}{\psi_i(x^{r-1})} = \lambda_i^{r-1}$. Тогда получим, что

$$F(\lambda) = \prod_{i \in I} (\varphi_i(x^r) - \lambda_i^{r-1}\psi_i(x^r)) = \prod_{i \in I} \left(\varphi_i(x^r) - \frac{\varphi_i(x^r)}{\psi_i(x^r)} \cdot \psi_i(x^r) \right) = 0.$$

Б. *Метод частичной линеаризации.* Задача (6.104) переписывается в следующей эквивалентной форме:

$$\prod_{i \in I} \lambda_i \to \min, \tag{6.109}$$

$$\sum_{j=1}^{n} (c_{ij} - \lambda_i d_{ij}) x_j - d_i^0 \lambda_i = -c_i^0 \quad (i = \overline{1, m}), \tag{6.110}$$

$$x \in S, \tag{6.111}$$

которая является нелинейной задачей из-за произведения $\lambda_i \cdot x_j$ переменных λ_i и x_j в ограничениях (6.110) и нелинейности функционала (6.109). Для ее решения используется итеративный приближенный алгоритм с применением метода частичной линеаризации на каждом его шаге и замене произведения параметров λ_i на их суммирования.

Алгоритм решения задачи (6.104) по такому методу содержит следующие шаги.

Шаг 0. Пусть $x^0 \in S$ и $\lambda_i^0 = \dfrac{\varphi_i(x^0)}{\psi_i(x^0)}$, $i \in I$. Тогда точка (x^0, λ^0) является допустимым решением задачи (6.104).

Шаг r. Пусть имеем допустимую точку $x^{r-1} \in S$, найденную на предыдущем шаге, определяем значения $\lambda_i^{r-1} = \dfrac{\varphi_i(x^{r-1})}{\psi_i(x^{r-1})}$, $i \in I$. Тогда:

- находим оптимальное решение (x^r, λ^r) полученной в результате частичной линеаризации в точке $(x^{r-1}, \lambda^{r-1}$ следующей задачи линейного программирования:

$$\sum_{i \in I} \lambda_i \to \min, \tag{6.112}$$

$$\sum_{j=1}^{n} q_{ij} x_j - q_i \lambda_i = w_i \quad (i = \overline{1, m}), \tag{6.113}$$

$$x \in S,\ \lambda_i \geq 0, \quad i \in I, \tag{6.114}$$

в которой

$$q_{ij} = c_{ij} - \lambda_i^{r-1} d_{ij} \quad (i = \overline{1, m};\ j = \overline{1, n}),$$

$$q_i = \sum_{j=1}^{n} d_{ij} x_j^{r-1} + d_i^0 \quad (i = \overline{1, m}),$$

$$w_i = -\lambda_i^{r-1} \sum_{j=1}^{n} d_{ij} x_j^{r-1} - (c_i^0 - \lambda_i^{r-1} d_i^0) \quad (i = \overline{1, m});$$

- если $|\lambda_i^{r-1} - \lambda_i^r| \leq \delta_i$, $i \in I$, то x^r является оптимальным решением задачи (6.104). Останов, в противном случае

- находим $\lambda_i^r = \dfrac{\varphi_i(x^r)}{\psi_i(x^r)}$, $i \in I$ и выполняем очередной $(r+1)$-й шаг алгоритма.

В алгоритмах **А** и **Б** решается последовательность линейных мультипликативных задач (6.106)–(6.108) и линейных задач (6.112)–(6.114), в которых к первоначальным ограничениям прибавляются еще m ограничений, которые меняются на каждом шаге алгоритмов. Для решения задачи (6.106)–(6.108) в работах [216, 309, 324, 361] предлагаются различные точные и приближенные алгоритмы. Как правило, в таких алгоритмах решаются вспомогательные задачи линейного программирования, в которых кроме некоторых ограничений $x \in S$ имеются дополнительные ограничения и переменные. Поэтому наличие таких ограничений не позволяет применить для их решения специальные алгоритмы, когда первоначальные ограничения задачи (6.104) имеют специальную структуру (например, блочно-диагональную или транспортного типа).

В. *Схема декомпозиции по ограничениям.* Для решения задачи (6.104) специальной структуры или большой размерности можно применить алгоритм, основанный на схеме декомпозиции по ограничениям и субградиентном методе недифференцируемой оптимизации.

Для задачи (6.104) построим функцию Лагранжа по формуле

$$L(x, \lambda, u) = \prod_{i \in I} \lambda_i + \sum_{i=1}^{m} u_i \left(\sum_{j=1}^{n} (c_{ij} - \lambda_i d_{ij}) x_j + c_i^0 - \lambda_i d_i^0 \right),$$

в которой $u = \{u_i, \quad i \in I\}$ – множители Лагранжа.

Тогда задача (6.104) сводится к решению задач

$$\max_{u} L^*(u), \tag{6.115}$$

$$L^*(u) = \min_{\lambda \geq 0} \min_{x \in S} L(x, \lambda, u). \tag{6.116}$$

Функция $L^*(u)$ определена для любого u и является кусочно-линейной, вогнутой и недифференцируемой. Для ее максимизации используется один из алгоритмов субградиентного метода, на каждом шаге которого решается задача (6.116) при фиксированных значениях переменных u. Тогда на шаге $r+1$ субградиентного метода необходимо выполнить следующие три этапа.

1. Решить задачу (6.116) при $u = u^r$ и найти оптимальное решение $(x^*(u^r), \lambda^*(u^r))$.

2. Определить значения обобщенного градиента функции $L^*(u)$ в точке $u = u^r$ по формуле

$$g_i(u^r) = \sum_{j=1}^{n} (c_{ij} - \lambda_i^*(u^r) d_{ij}) x_j^*(u^r) + c_i^0 - \lambda_i^*(u^r) d_i^0, \quad i \in I.$$

3. Найти новые значения переменных u по формуле

$$u_i^{r+1} = u_i^r + h_{r+1} g_i(u^r), \quad i \in I,$$

где h_{r+1} – величина шага.

Если u^* – оптимальное решение задачи (6.113), то (x^*, λ^*) – решение задачи (6.116) при $u = u^*$. Тогда (x^*, λ^*) являются оптимальным решением задачи (6.109)–(6.111), а x^* – задачи (6.104).

Г. *Алгоритм решения задачи* (6.116). На каждом шаге субградиентного метода при фиксированных двойственных переменных $u = u^r$ решается следующая задача:

$$\min_{x\in S}\min_{\lambda\geq 0}\left[\prod_{i\in I}\lambda_i + \sum_{i=1}^{m} u_i^r\left(\sum_{j=1}^{n}(c_{ij} - \lambda_i d_{ij})x_j + c_i^0 - \lambda_i d_i^0\right)\right], \qquad (6.117)$$

которая является задачей мультипликативного билинейного программирования из-за произведений $\lambda_i x_j$ и произведения переменных λ_i, $i \in I$.

При фиксированных $\lambda_i > 0 \ \ (i = \overline{1,m})$ задача (6.117) является задачей линейного программирования. Поэтому процесс решения данной задачи сводится к нахождению такого решения $x^* \in S$ и таких значений $\lambda_i^* > 0 \ \ (i = \overline{1,m})$, которые в совокупности представляли бы оптимальное решение (x^*, λ^*) задачи (6.117).

Для решения задачи (6.117) может быть использован приближенный алгоритм, в результате чего при фиксированных значениях переменных $u = u^*$ задачи (6.115) получим оптимальное решение задачи (6.117). Такой алгоритм может быть построен, если для решения задачи (6.117) будем фиксировать значения параметров λ_i по следующей формуле:

$$\lambda^* = \frac{\varphi_i(x^*(u^*))}{\psi_i(x^*(u^*))},$$

в которой $x^*(u^*) \in S$ – решение задачи (6.117).

Допустим, что после выполнения r шагов субградиентного метода имеем значение переменных $u = u^r$. Тогда задача (6.117) на очередном шаге субградиентного метода решается в следующем порядке:

- фиксируем значения λ_i по формуле

$$\lambda_i' = \begin{cases} 0, & \text{для} \quad r = 1; \\ \lambda_i^*(u^r), & \text{для} \quad r \geq 2, \ i \in I; \end{cases}$$

- находим оптимальное решение $x^*(u^r)$ задачи линейного программирования

$$\min_{x\in S}\sum_{j=1}^{n} q_j x_j, \qquad (6.118)$$

в которой $q_j = \sum_{i \in I} u_i^r \big(c_{ij} - \lambda_i' d_{ij} \big) \quad (j = \overline{1, n});$

- находим новое значение для λ_i по формуле

$$\lambda_i^*(u^r) = \frac{\varphi_i(x^*(u^r))}{\psi_i(x^*(u^r))}, \quad i \in I.$$

Пара полученных значений $(x^*(u^r), \lambda^*(u^r))$ будет приближенным решением задачи (6.117).

В предложенном алгоритме на каждом шаге решается задача линейного программирования (6.118), в которой меняются только значения коэффициентов, а первоначальные ограничения остаются без изменений. Как уже отмечалось выше, такое обстоятельство имеет очень важное значение при решении задачи (6.114) с ограничениями специальной структуры.

6.5. Задача оптимизации отношения максимума и минимума конечного числа функций

Рассмотрим задачу дробной оптимизации, в которой требуется минимизировать отношение двух функций, полученных в результате применения операции взятия поточечного максимума от конечного числа выпуклых функций и операции взятия поточечного минимума от конечного числа вогнутых функций, т.е. имеем задачу дробного программирования следующего вида:

$$f(x) = \frac{\varphi(x)}{\psi(x)} = \frac{\max\limits_{i \in I} \varphi_i(x)}{\min\limits_{i \in I} \psi_i(x)} \to \min, \tag{6.119}$$

$$h_k(x) \leq 0 \quad (k = \overline{1, p}), \tag{6.120}$$

в которой функции $\{\varphi_i(x),\, i \in I\}$, $\{-\psi_i(x),\, i \in I\}$ и $\{h_k(x),\ k = \overline{1,p}\}$ – непрерывны, выпуклы и недифференцируемы на R^n. Предполагается также, что функции $\varphi_i(x) \geq 0, \quad i \in I$, а $\psi_i(x) > 0, \quad i \in I$ для всех $x \in S$, где S – множество допустимых решений x, удовлетворяющие ограничениям (6.120), т.е. $S = \{x : \ h_k(x) \leq 0, \quad k = \overline{1,p}\}$.

Для задачи (6.119), (6.120) можно рассматривать две отдельные задачи

$$\min_{x \in S} \Big[\varphi(x) = \max_{i \in I} \varphi_i(x) \Big], \tag{6.121}$$

$$\max_{x\in S}\Big[\psi(x)=\min_{i\in I}\psi_i(x)\Big]. \tag{6.122}$$

Каждая из задач (6.121) и (6.122), в отдельности, являются задачами дискретного минимакса и для их решения можно применять приведенные в п. 5.1 методы, а именно методы сведения к задачам выпуклого программирования или недифференцируемой оптимизации с применением субградиентных методов.

Тогда, если решения задач (6.121) и (6.122) совпадают, то они будут оптимальными решениями и для исходной задачи (6.119), (6.120). Однако, такое имеет место не всегда.

Поэтому для решения задач (6.119), (6.120) применяются другие методы, среди которых двойственный параметрический метод, субградиентные методы и схемы декомпозиции по ограничениям.

Сначала рассмотрим эквивалентную задачу в виде серии эквивалентных задач дробного программирования:

$$\min_{x\in S} f(x)=\min_{x\in S}\frac{\varphi(x)}{\psi(x)}=\min_{x\in S}\max_{\substack{u\in U\\ v\in V}}\frac{\varphi(x,u)}{\psi(x,v)}=\min_{x\in S}\max_{\substack{u\in U\\ v\in V}}\frac{\sum\limits_{i=1}^{m}u_i\varphi_i(x)}{\sum\limits_{i=1}^{m}v_i\psi_i(x)}, \tag{6.123}$$

в которой

$$S=\{x\colon\ h_k(x)\le 0,\quad k=\overline{1,p}\},$$

$$U=\Big\{u\colon\ \sum_{i=1}^{m}u_i=1,\quad u_i\ge 0,\quad i=\overline{1,m}\Big\},$$

$$V=\Big\{v\colon\ \sum_{i=1}^{m}v_i=1,\quad v_i\ge 0,\quad i=\overline{1,m}\Big\}.$$

Задача (6.123) эквивалентна исходной задаче (6.119), (6.120), которая используется в дальнейшем для построения двойственной задачи. С другой стороны, в соответствие с теорией минимакса, задача (6.123) эквивалентна другой задаче дробной оптимизации

$$\max_{\substack{u\in U\\ v\in V}}\min_{x\in S}\frac{\sum\limits_{i=1}^{m}u_i\varphi_i(x)}{\sum\limits_{i=1}^{m}v_i\psi_i(x)}. \tag{6.124}$$

Тогда решение задачи (6.124) может быть разбита на решение двух оптимизационных задач

1) внешняя задача

$$\max_{\substack{u\in U\\ v\in V}} \Theta(u,v), \tag{6.125}$$

2) внутренняя задача

$$\Theta(u,v)=\min_{x\in S}\frac{\varphi(x,u)}{\psi(x,v)}=\min_{x\in S}\frac{\sum\limits_{i=1}^{m} u_i\varphi_i(x)}{\sum\limits_{i=1}^{m} v_i\psi_i(x)}. \tag{6.126}$$

Задача (6.125) может быть рассмотрена как двойственная к задаче (6.119), (6.120), а функция $\Theta(u,v)$ является квазивогнутой и недифференцируемой.

Для задачи (6.126) рассмотрим параметрическую задачу

$$F(u,v,\lambda)=\min_{x\in S}\sum_{i=1}^{m}\big(u_i\varphi_i(x)-\lambda v_i\psi_i(x)\big). \tag{6.127}$$

Тогда в соответствии с параметрическим методом для оптимальных решений x^*, u^* и v^* задачи (6.124) имеем $\lambda^*=f(x^*)$ и $F(u^*,v^*,\lambda^*)=0$, или же $F(u^*,v^*,\Theta(u^*,v^*))=0$. Тогда вместо исходной задачи (6.119), (6.120) или эквивалентной ей задачи (6.124) решаем ее двойственную задачу (6.125). Для ее решения применим двойственный параметрический метод и субградиентный метод.

1. Двойственный параметрический метод. Для решения эквивалентной задачи (6.123) используем двойственный параметрический метод, который содержит следующие шаги.

Шаг 0. Пусть $u^0\in U$ и $v^0\in V$. Находим решение $x^*(u^0,v^0)$ задачи дробно-выпуклого программирования

$$\lambda_0=\Theta(u^0,v^0)=\min_{x\in S}\frac{\sum\limits_{i=1}^{m} u_i^0\varphi_i(x)}{\sum\limits_{i=1}^{m} v_i^0\psi_i(x)},$$

для решения которой можно применить любые имеющиеся методы и алгоритмы, в частности параметрический метод для решения эквивалентной задачи выпуклого программирования (6.127) вида

$$\min_{x\in S}\sum_{i=1}^{m}\big(u_i^0\varphi_i(x)-\lambda v_i^0\psi_i(x)\big).$$

Шаг r. Пусть на $(r-1)$-м шаге получены следующие значения: (u^{r-1}, v^{r-1}) – значения переменных (u,v); $x^*(u^{r-1}, v^{r-1})$ – значения переменных x; $\lambda_{r-1} = \Theta(u^{r-1}, v^{r-1})$ – значения параметра λ. Тогда на r-м шаге двойственного параметрического метода выполняем следующие процедуры.

1. Находим оптимальное решение (u^r, v^r) задачи (6.127), которая при фиксированном $\lambda = \lambda_{r-1}$ и $x^*(u^{r-1}, v^{r-1})$ имеет вид задачи линейного программирования

$$\sum_{i=1}^{m} a_i u_i + \sum_{i=1}^{m} b_i v_i \to \max, \tag{6.128}$$

$$\sum_{i=1}^{m} u_i = 1, \quad u_i \geq 0, \quad i \in I, \tag{6.129}$$

$$\sum_{i=1}^{m} v_i = 1, \quad v_i \geq 0, \quad i \in I, \tag{6.130}$$

в которой

$$a_i = u_i^{r-1} \varphi_i(x^*(u^{r-1}, v^{r-1})) \geq 0, \quad i \in I;$$
$$b_i = -v_i^{r-1} \psi_i(x^*(u^{r-1}, v^{r-1})) \leq 0, \quad i \in I.$$

Решение данной задачи определяется по формулам:

$$u_i^r = \begin{cases} 1, & \text{для } \ i = i_{r_1}, \\ 0, & \text{для } \ i = \overline{1,m}, \ i \neq i_{r_1}; \end{cases}$$

$$v_i^r = \begin{cases} 1, & \text{для } \ i = i_{r_2}, \\ 0, & \text{для } \ i = \overline{1,m}, \ i \neq i_{r_2}, \end{cases}$$

где i_{r_1} – значение индекса i для которого $a_{i_{r_1}} = \max\limits_{i \in I} a_i$; i_{r_2} – значение индекса i для которого $b_{i_{r_2}} = \min\limits_{i \in I} b_i$.

2. Если $F(u^r, v^r, \lambda_{r-1}) = 0$, то (u^{r-1}, v^{r-1}) – оптимальное решение двойственной задачи (6.125), а $x^*(u^{r-1}, v^{r-1})$ – оптимальное решение исходной задачи (6.119), (6.120). Останов. В противном случае переходим к процедуре 3.

3. Находим новое значение параметра λ путем решения задачи дробно-выпуклого программирования

$$\lambda_r = \Theta(u^r, v^r) = \min_{x \in S} \frac{\sum\limits_{i=1}^{m} u_i^r \varphi_i(x)}{\sum\limits_{i=1}^{m} v_i^r \psi_i(x)} = \min_{x \in S} \frac{\varphi_{i_{r_1}}(x)}{\psi_{i_{r_2}}(x)} = \frac{\varphi_{i_{r_1}}(x^*(u^r, v^r))}{\psi_{i_{r_2}}(x^*(u^r, v^r))},$$

т.е. параметрическим методом при $\lambda_{r-1} = \Theta(u^{r-1}, v^{r-1})$ находим решение $x^*(u^r, v^r)$ задачи выпуклого программирования

$$\min_{x \in S} \sum_{i=1}^{m} \big(u_i^r \varphi_i(x) - \Theta(u^{r-1}, v^{r-1}) v_i^r \psi_i(x)\big) =$$
$$= \min_{x \in S} \big(\varphi_{i_{r_1}}(x) - \Theta(u^{r-1}, v^{r-1}) \psi_{i_{r_2}}(x)\big).$$

Так как $u_i^r = 1$ для $i = i_{r_1}$, для которого имеем $a_{i_{r_1}} = \max\limits_{i \in I} a_i$, а остальные $u_i^r = 0$ для $i = \overline{1,m}$, $\;i \neq i_{r_1}$, а также $v_i^r = 1$ для $i = i_{r_2}$, для которого имеем $b_{i_{r_2}} = \min\limits_{i \in I} b_i$, а остальные $v_i^r = 0$ для $i = \overline{1,m}$, $i \neq i_{r_2}$, то данная задача выпуклого программирования решается только относительно функций $\varphi_{i_{r_1}}(x)$ и $\psi_{i_{r_2}}(x)$, т.е. параметрическим методом получаем решение для задачи дробно-выпуклого программирования

$$\lambda_r = \Theta(u^r, v^r) = \min_{x \in S} \frac{\varphi_{i_{r_1}}(x)}{\psi_{i_{r_2}}(x)} = \frac{\varphi_{i_{r_1}}\left(x^*(u^r, v^r)\right)}{\psi_{i_{r_2}}\left(x^*(u^r, v^r)\right)}.$$

В результате применения двойственного параметрического метода получаем итерационный процесс относительно параметра λ_r, в результате которого полученная последовательность $\{\lambda_r\}$ стремится к корню λ^* уравнения $F(u^*, v^*, \lambda^*) = 0$, т.е. получим равенство

$$\sum_{i=1}^{m} \big(u_i^* \varphi_i(x^*) - \lambda^* v_i^* \psi_i(x^*)\big) = 0,$$

в которой:

(u^*, v^*) – оптимальное решение соответствующей задачи линейного программирования (6.128)–(6.130), в которой $u_{i_1^*}^* = 1$, а $u_i^* = 0$ для всех $i = \overline{1,m}$, $\;i \neq i_1^*$, а i_1^* – значение индекса i, для которого имеем $a_{i_1^*} = \max\limits_{i \in I}\ a_i$, а также $v_{i_2^*}^* = 1$, а $v_i^* = 0$ для всех $i = \overline{1,m}$, $\;i \neq i_2^*$, а i_2^* – значение индекса i, для которого имеем $b_{i_2^*} = \min\limits_{i \in I}\ b_i$;

x^* – оптимальное решение задачи выпуклого программирования

$$\min_{x \in S} \big(\varphi_{i_1^*}(x) - \lambda^* \psi_{i_2^*}(x)\big).$$

Таким образом, для параметров (u^*, v^*) и решения x^* имеем равенства

$$\lambda^* = f(x^*) = \frac{\varphi_{i_1^*}(x^*)}{\psi_{i_2^*}(x^*)} = \frac{\sum\limits_{i=1}^{m} u_i^* \varphi_i(x^*)}{\sum\limits_{i=1}^{m} v_i^* \psi_i(x^*)} = \frac{\max\limits_{i \in I} \varphi_i(x^*)}{\min\limits_{i \in I} \psi_i(x^*)} = \min_{x \in S} \frac{\max\limits_{i \in I} \varphi_i(x)}{\min\limits_{i \in I} \psi_i(x)},$$

т.е. x^* является оптимальным решением исходной задачи обобщенного дробно-выпуклого программирования (6.119), (6.120).

2. Субградиентный метод решения двойственной задачи. Для решения двойственной задачи (6.125) применим субградиентный метод недифференцируемой оптимизации, а именно решаем задачу $\max\limits_{\substack{u\in U\\ v\in V}} \Theta(u,v)$. Пусть имеем точку (u^r,v^r), тогда на $(r+1)$-м шаге субградиентного метода необходимо выполнить следующие три основных этапа.

1. При фиксированном $\lambda_r = \Theta(u^r,v^r)$ решить задачу выпуклого программирования

$$\min_{x\in S}\sum_{i=1}^{m}\big(u_i^r\varphi_i(x)-\lambda_r v_i^r\psi_i(x)\big)$$

и найти ее оптимальное решение $x^*(u^r,v^r)$.

2. Найти значения субградиента функции $F(u,v,\Theta(u^r,v^r)) = F(u,v,\lambda)$ в точке (u^r,v^r) по формулам

$$g_F^u(u^r,v^r)=\big\{g_F^{u_i}(u^r,v^r)=\varphi_i(x^*(u^r,v^r)),\quad i=\overline{1,m}\big\},$$

$$g_F^v(u^r,v^r)=\big\{g_F^{v_i}(u^r,v^r)=-\lambda_r\psi_i(x^*(u^r,v^r)),\quad i=\overline{1,m}\big\}.$$

3. Найти новую точку (u^{r+1},v^{r+1}) по формулам $u^{r+1}=\Pi_U(\widetilde{u}^{r+1})$ и $v^{r+1}=\Pi_V(\widetilde{v}^{r+1})$, где $\widetilde{u}_i^{r+1}=\max\big\{0,\ u_i^r+\gamma_{r+1}g_F^{u_i}(u^r,u^r)\big\}$, $i=\overline{1,m}$, $\widetilde{v}_i^{r+1}=\max\big\{0,\ v_i^r+\gamma_{r+1}g_F^{v_i}(u^r,u^r)\big\}$, $i=\overline{1,m}$, а $\Pi_U(\widetilde{u}^{r+1})$ – оператор проектирования точки $\widetilde{u}^{r+1}$ на линейное многообразие $U = \Big\{u:\sum\limits_{i=1}^{m}u_i=1,\quad u_i\ge 0,\quad i\in I\Big\}$; $\Pi_V(\widetilde{v}^{r+1})$ – оператор проектирования точки $\widetilde{v}^{r+1}$ на линейное многообразие $V=\Big\{v:\sum\limits_{i=1}^{m}v_i=1,\ v_i\ge 0,\ i\in I\Big\}$.

Пусть (u^*,v^*) – оптимальное решение задачи недифференцируемой оптимизации (6.125), а $\lambda^*=\Theta(u^*,v^*)$, тогда полученное решение x^* задачи выпуклого программирования

$$\min_{x\in S}\sum_{i=1}^{m}(u_i^*\varphi_i(x)-\lambda^* v_i^*\psi_i(x))$$

является оптимальным решением исходной задачи (6.119), (6.120), причем

$$\lambda^*=\Theta(u^*,v^*)=f(x^*),\qquad \text{а}\qquad F^*(u^*,v^*,\lambda^*)=0.$$

3. Субградиентный метод решения задачи (6.119), (6.120). Пусть имеем общую задачу дробного программирования (6.119), (6.120) и соответствующую ей параметрическую задачу

$$F(\lambda) = \min_{x \in S} \left[Z(x, \lambda) = \max_{i \in I} \varphi_i(x) - \lambda \min_{i \in I} \psi_i(x) \right]$$

или же задачу

$$F(\lambda) = \min_{x \in S} \left[Z(x, \lambda) = \varphi(x) - \lambda \psi(x) \right],$$

в которой функции $\varphi(x) = \max_{i \in I} \varphi_i(x)$ и $\psi(x) = \min_{i \in I} \psi_i(x)$ являются недифференцируемыми функциями, $\varphi(x)$ – выпукла, а $\psi(x)$ – вогнута на S, притом $\varphi(x) \geq 0$, $\psi(x) > 0$ для любого $x \in S$. Тогда функция $Z(x, \lambda)$ является выпуклой недифференцируемой функцией при $\lambda \geq 0$.

Применим метод эллипсоидов для решения задачи $\min_{x \in S} Z(x, \lambda)$ при условии обеспечения неотрицательности значения параметра λ на множестве допустимых решений S. Тогда в соответствии с методом эллипсоидов минимизации выпуклых функций (см. п. 5.3.) значения обобщенных градиентов функции $Z(x, \lambda)$ в точке x_r определяются по следующей формуле:

$$g(x_r) = \begin{cases} g_Z(x_r) = g_{\varphi_{i_{r_1}}}(x_r) - \lambda_r g_{\psi_{i_{r_2}}}(x_r), & \text{если } \ h_{k_r}(x_r) \leq 0; \\ g_h(x_r) = g_{h_{k_r}}(x_r), \ \text{ если } \ h_{k_r}(x_r) > 0, \end{cases}$$

в которой:

x_r – найденная точка на r-м шаге;

k_r – индекс, для которого достигается поточечный максимум по функциям ограничений, а $h_{k_r}(x_r)$ – значение этого максимума в точке x_r, т.е. k_r – индекс, для которого $h_{k_r}(x_r) = \max_{1 \leq k \leq p} h_k(x_r)$;

i_{r_1} – индекс, для которого достигается поточечный максимум по функциям $\varphi_i(x_r)$, т.е. i_{r_1} – индекс, для которого $\varphi_{i_{r_1}}(x_r) = \max_{i \in I} \varphi_i(x_r)$;

i_{r_2} – индекс, для которого достигается поточечный минимум по функциям $\psi_i(x_r)$, т.е. i_{r_2} – индекс, для которого $\psi_{i_{r_2}}(x_r) = \min_{i \in I} \psi_i(x_r)$;

λ_r – значение параметра λ, которое определяется в текущей точке x_r по формуле

$$\lambda_r = \begin{cases} f(x_r) = \dfrac{\varphi_{i_{r_1}}(x_r)}{\psi_{i_{r_2}}(x_r)}, & \text{если } \ \varphi_{i_{r_1}}(x_r) \geq 0 \ \text{ и } \ \psi_{i_{r_2}}(x_r) > 0; \\ 0, \ \text{ в противном случае;} \end{cases}$$

$g_{z}(x_r)$ – значение субградиента функции $Z(x,\lambda)$ в точке x_r при фиксированном значении параметра $\lambda=\lambda_r$;

$g_h(x_r)$ – значение субградиента, определенное для одной из функций ограничений $h_k(x)\le 0 \quad (k=\overline{1,p})$;

$g_{\varphi_{i_{r_1}}}(x_r)$, $g_{\psi_{i_{r_2}}}(x_r)$ и $g_{h_{k_r}}(x_r)$ – значения субградиентов в точке x_r соответственно для функций $\varphi_{i_{r_1}}(x)$, $\psi_{i_{r_2}}(x)$ и $h_{k_r}(x)$.

Пусть x^* оптимальное решение задачи $\min\limits_{x\in S} Z(x,\lambda)$, найденное методом эллипсоидов при $\lambda^*=f(x^*)$. Тогда x^* будет оптимальным решением и для исходной задачи (6.119), (6.120). Действительно, так как x^* является оптимальным решением задачи $\min\limits_{x\in S} Z(x,\lambda)$ при фиксированном значением параметра λ, то, в соответствие с методом эллипсоидов, субградиентное множество в точке x^* содержит нулевой вектор, т.е. $0\in g(x^*)$. Так как $x^*\in S$, т.е. является допустимым решением для задачи (6.119), (6.120), то $0\in g_{z}(x^*)$ и тогда имеется хотя бы одно нулевое значение субградиента $g_{z}(x^*)$. Таким образом, имеем равенство $g_{\varphi_{i_1^*}}(x^*)-\lambda^* g_{\psi_{i_2^*}}(x^*)=0$, в котором:

x^* – допустимое решение, т.е. $x^*\in S$;

i_1^* – индекс, для которого достигается поточечный максимум функций $\varphi_i(x)$ в точке x^*, т.е. для которого $\varphi_{i_1^*}=\max\limits_{i\in I}\varphi_i(x^*)$;

i_2^* – индекс, для которого достигается поточечный минимум функций $\psi_i(x)$ в точке x^*, т.е. для которого $\psi_{i_2^*}=\min\limits_{i\in I}\psi_i(x^*)$.

Откуда получим, что

$$\lambda^*=\frac{g_{\varphi_{i_1^*}}(x^*)}{g_{\psi_{i_2^*}}(x^*)}=f(x^*)=\frac{\varphi_{i_1^*}(x^*)}{\psi_{i_2^*}(x^*)}.$$

Тогда имеем

$$F(\lambda^*)=Z(x^*,\lambda^*)=\varphi_{i_1^*}(x^*)-\lambda^*\psi_{i_2^*}(x^*)=\varphi_{i_1}(x^*)-\frac{\varphi_{i_1^*}(x^*)}{\psi_{i_2^*}(x^*)}\psi_{i_2^*}(x^*)=0,$$

т.е. λ^* является корнем уравнения $F(\lambda)=0$, а x^* является оптимальным решением исходной задачи (6.119), (6.120).

4. Эквивалентные задачи. Как отмечено ранее некоторые методы решения обобщенных задач дробно-выпуклого программирования сводятся к решению их эквивалентных выпуклых задач но не с дробным функционалом. Для задачи (6.119), (6.120) можно рассматривать эквивалентную задачу не дробной оптимизации:

$$\lambda\to\min; \tag{6.131}$$

$$\varphi_i(x) \le t_1, \quad i \in I; \tag{6.132}$$

$$\psi_i(x) \ge t_2, \quad i \in I; \tag{6.133}$$

$$t_1 - \lambda t_2 = 0; \tag{6.134}$$

$$h_k(x) \le 0 \quad (k = \overline{1,p}); \tag{6.135}$$

$$t_1 \ge 0, \quad t_2 \ge 0. \tag{6.136}$$

Задача (6.131)–(6.136) является задачей оптимизации линейной функции при выпуклых ограничениях.

Рассмотрим следующие утверждения, показывающие на эквивалентность задач (6.119), (6.120) и (6.131)–(6.136).

Теорема 6.13. *Если* x^* *– оптимальное решение задачи* (6.119), (6.120)*, то существуют такие* t_1^* $(t_1^* = \varphi(x^*))$*,* t_2^* $(t_2^* = \psi(x^*))$ *и* $\lambda^* = t_1^*/t_2^*$*, что* $(x^*, t_1^*, t_2^*, \lambda^*)$ *является оптимальным решением задачи* (6.131)–(6.136)*.*

Доказательство. Пусть x^* – оптимальное решение задачи (6.119), (6.120). Тогда имеем $f(x^*) \le f(x)$, $\dfrac{\varphi(x^*)}{\psi(x^*)} \le \dfrac{\varphi(x)}{\psi(x)}$, или же $\dfrac{\max\limits_{i\in I}\varphi_i(x^*)}{\min\limits_{i\in I}\psi_i(x^*)} \le$ $\le \dfrac{\max\limits_{i\in I}\varphi_i(x)}{\min\limits_{i\in I}\psi_i(x)}$ для любого $x \in S$. Определим параметры t_1^* и t_2^* по формулам $t_1^* = \max\limits_{i\in I}\varphi_i(x^*) = \varphi(x^*)$ и $t_2^* = \min\limits_{i\in I}\psi_i(x^*) = \psi(x^*)$. Откуда получим, что $\varphi_i(x^*) \le t_1^*, \quad i \in I$, а $\psi_i(x^*) \ge t_2^*, \quad i \in I$. Определим $\lambda^* = t_1^*/t_2^*$, откуда имеем $t_1^* - \lambda^* t_2^* = 0$. Таким образом $(x^*, t_1^*, t_2^*, \lambda^*)$ является допустимым решением задачи (6.131)–(6.136).

Покажем теперь, что $(x^*, t_1^*, t_2^*, \lambda^*)$ является оптимальным решением задачи (6.131)–(6.136). Допустим обратное, что существует такой $\overline{\lambda}$, для которого $\lambda^* > \overline{\lambda}$, а $(x^*, t_1^*, t_2^*, \lambda^*)$ является только допустимым решением задачи (6.131)–(6.136). Действительно, из определения параметров $t_1^* = \max\limits_{i\in I}\varphi_i(x^*)$ и $t_2^* = \min\limits_{i\in I}\psi_i(x^*)$ имеем ограничения $\varphi_i(x^*) \le t_1^*, \ i \in I$ и $\psi_i(x^*) \ge t_2^*, \quad i \in I$. Откуда получаем выполнение ограничений (6.132) и (6.133). С другой стороны, с учетом равенства (6.134) имеем $\lambda^* = \dfrac{t_1^*}{t_2^*} = \dfrac{\varphi(x^*)}{\psi(x^*)} = \dfrac{\max\limits_{i\in I}\varphi_i(x^*)}{\min\limits_{i\in I}\psi_i(x^*)}$ и $\overline{\lambda} = \dfrac{t_1^*}{t_2^*} = \dfrac{\max\limits_{i\in I}\varphi_i(x^*)}{\min\limits_{i\in I}\psi_i(x^*)} = \lambda^*$, что противоречит предположению $\overline{\lambda} < \lambda^*$. Таким образом $(x^*, t_1^*, t_2^*, \lambda^*)$ является оптимальным решением задачи (6.131)–(6.136). Теорема доказана.

Теорема 6.14. *Если* $(x^*, t_1^*, t_2^*, \lambda^*)$ *– оптимальное решение задачи* (6.131)–(6.136)*, то* x^* *– оптимальное решение задачи* (6.119), (6.120).

Доказательство. Пусть $(x^*, t_1^*, t_2^*, \lambda^*)$ – оптимальное решение задачи (6.131)–(6.136). Тогда x^* удовлетворяет ограничениям (6.135), т.е. является допустимым решением задачи (6.119), (6.120). Так как λ^* является оптимальным значением функционала (6.131), а x^* удовлетворяет ограничениям (6.132)–(6.134), то имеем $\lambda^* \leq \lambda$ для любого λ, удовлетворяющего равенству (6.134) при любых значениях $x \in S$. Тогда получим неравенства $\varphi_i(x^*) \leq t_1^*$ и $\psi_i(x^*) \geq t_2^*$, $i \in I$, или же равенства $\max\limits_{i\in I} \varphi_i(x^*) = t_1^*$ и $\min\limits_{i\in I} \psi_i(x^*) = t_2^*$. Откуда имеем

$$\lambda^* = \frac{t_1^*}{t_2^*} == \frac{\max\limits_{i\in I} \varphi_i(x^*)}{\min\limits_{i\in I} \psi_i(x^*)} \leq \lambda = \frac{t_1}{t_2} = \frac{\max\limits_{i\in I} \varphi_i(x)}{\min\limits_{i\in I} \psi_i(x)},$$

для любого $x \in S$, т.е. имеем неравенства

$$\frac{\max\limits_{i\in I} \varphi_i(x^*)}{\min\limits_{i\in I} \psi_i(x^*)} \leq \frac{\max\limits_{i\in I} \varphi_i(x)}{\min\limits_{i\in I} \psi_i(x)} \quad \text{или} \quad \frac{\varphi(x^*)}{\psi(x^*)} \leq \frac{\varphi(x)}{\psi(x)}, \quad \text{или же} \quad f(x^*) \leq f(x).$$

Следовательно, x^* является оптимальным решением задачи (6.119), (6.120). Теорема доказана.

Заметим, что задача (6.131)–(6.136) может быть сформулирован следующим образом: найти те значения переменных $x \in S$, для которых переменные t_1 и t_2 удовлетворяют ограничениям (6.132)–(6.134), а λ принимает минимальное значение. Другими словами, необходимо найти значение x^* переменных $x \in S$, для которых определенное по формуле

$$\lambda^* = \frac{t_1^*}{t_2^*} = \frac{\max\limits_{i\in I} \varphi_i(x^*)}{\min\limits_{i\in I} \psi_i(x^*)}$$

значение переменной λ будет минимальным.

5. Схема декомпозиции по ограничениям. Другим, двойственным подходом, при решении задачи (6.119), (6.120) или ее эквивалентной задачи (6.131)–(6.136) является построение функции Лагранжа и рассмотрение двойственных лагранжевых задач с учетом параметрической задачи. Для решения полученной двойственной задачи применяются субградиентные методы недифференцируемой оптимизации.

Для эквивалентной задачи (6.131)–(6.134) построим функцию Лагранжа по формуле

$$L(x,t_1,t_2,\lambda,\beta,u,v) = \lambda+\beta(t_1-\lambda t_2)+\sum_{i=1}^{m} u_i(\varphi_i(x)-t_1)+\sum_{i=1}^{m} v_i(t_2-\psi_i(x)) =$$

$$= \lambda + t_1\Bigg(\beta - \sum_{i=1}^{m} u_i\Bigg) - t_2\Bigg(\lambda\beta + \sum_{i=1}^{m} v_i\Bigg) + \sum_{i=1}^{m}\Bigg(u_i\varphi_i(x) - v_i\psi_i(x)\Bigg),$$

где: β – множитель Лагранжа для равенства (6.134), $u = \{u_i \ge 0, i \in I\}$ – множители Лагранжа для ограничений (6.132), а $v = \{v_i \ge 0, \quad i \in I\}$ – множители Лагранжа для ограничений (6.133). Тогда задача (6.131)–(6.136), и тем самым (6.119), (6.120) сводится к нахождению седловой точки функции Лагранжа $L(x,t_1,t_2,\lambda,u,v,\beta)$, т.е. к решению задачи

$$L(x^*,t_1^*,t_2^*,\lambda^*,u^*,v^*,\beta^*) = \\ = \max_{u,v,\beta} \min_{\substack{x\in S,\lambda\ge 0\\ t_1\ge 0,t_2\ge 0}} L(x,t_1,t_2,\lambda,u,v,\beta). \tag{6.137}$$

В свою очередь задачу (6.137) можно свести к решению двойственной по Лагранжу задачи

$$\max_{u,v,\beta} L^*(u,v,\beta), \tag{6.138}$$

где

$$L^*(u,v,\beta) = \min_{\substack{x\in S,\lambda\ge 0\\ t_1\ge 0,t_2\ge 0}} L(x,t_1,t_2,\lambda,u,v,\beta). \tag{6.139}$$

Функция $L^*(u,v,\beta)$ определяется для любых $u \ge 0$, $v \ge 0$ и $\beta \ge 0$ и является кусочно-линейной, вогнутой и недифференцируемой. Поэтому для решения задачи (6.138) используем субградиентный метод, на каждом шаге которого необходимо решить задачу (6.139) при фиксированных значениях переменных u, v и β.

Пусть для решения задачи (6.138) используется некоторый субградиентный метод на r-м шаге которого имеем некоторую точку (u^r, v^r, β_r). Тогда на $(r+1)$-м шаге необходимо выполнить следующие три основных этапа.

1. Решить задачу (6.139) при фиксированных $u = u^r$, $v = v^r$ и $\beta = \beta_r$, т.е. найти оптимальное решение задачи

$$\min_{\substack{x\in S,\lambda\ge 0\\ t_1\ge 0,t_2\ge 0}} \Bigg[\lambda + t_1\Bigg(\beta_r - \sum_{i=1}^{m} u_i^r\Bigg) - t_2\Bigg(\lambda\beta_r + \sum_{i=1}^{m} v_i^r\Bigg) + \\ + \sum_{i=1}^{m}\Bigg(u_i^r\varphi_i(x) - v_i^r\psi_i(x)\Bigg)\Bigg].$$

2. Определить значения обобщенного градиента функции $L^*(u,v,\beta)$ в точке (u^r,v^r,β_r) по формулам

$$g_i^u(u^r,v^r,\beta_r)=\varphi_i\big(x^*(u^r,v^r,\beta_r)\big)-t_1,\quad i\in I,$$

$$g_i^v(u^r,v^r,\beta_r)=t_2-\psi_i\big(x^*(u^r,v^r,\beta_r)\big),\quad i\in I,$$

$$g^\beta(u^r,v^r,\beta_r)=t_1-\lambda t_2.$$

3. Найти новые значения переменных u^{r+1}, v^{r+1} и β^{r+1} по формулам

$$u_i^{r+1}=\max\{0,\ u_i^r+\gamma_{r+1}g_i^u(u^r,v^r,\beta_r)\},\quad i\in I,$$

$$v_i^{r+1}=\max\{0,\ v_i^r+\gamma_{r+1}g_i^v(u^r,v^r,\beta_r)\},\quad i\in I,$$

$$\beta_{r+1}=\max\left\{0,\ \beta_r+\gamma_{r+1}g^\beta(u^r,v^r,\beta_r)\right\},$$

где γ_{r+1} – величина шага в субградиентном методе.

Теорема 6.15. *Пусть* (u^*,v^*,β^*) *– оптимальное решение задачи* (6.138)*, а* $(x^*,t_1^*,t_2^*,\lambda^*)$ *– оптимальное решение задачи* (6.139) *при фиксированных* $u=u^*,\ v=v^*$ *и* $\beta=\beta^*$*. Тогда* $(x^*,t_1^*,t_2^*,\lambda^*)$ *– оптимальное решение задачи* (6.131)–(6.136)*, а* x^* *– оптимальное решение задачи* (6.119), (6.120).

Доказательство. Так как при $u=u^*,\ v=v^*$ и $\beta=\beta^*$ точка $(x^*,t_1^*,t_2,\lambda^*)$ является оптимальным решением задачи выпуклого программирования (6.139), то в соответствии с теоремой Куна-Таккера имеем:

1) $x^*\in S$ и тем самым, x^* удовлетворяет ограничениям (6.135) и (6.120);

2) $L(x^*,t_1^*,t_2^*,\lambda^*,u^*,v^*,\beta^*)\le L(x,t_1,t_2,\lambda,u^*,v^*,\beta^*)$ для любого $x\in S$;

3) $u_i^*\left(\varphi_i(x^*)-t_1^*\right)=0,\quad i\in I$;

4) $v_i^*\left(t_2^*-\psi_i(x^*)\right)=0,\quad i\in I$;

5) $\beta^*(t_1^*-\lambda^*t_2^*)=0.$

Тогда из 2) с учетом 3)–5) получим

$$\lambda^*-\lambda\le\beta^*(t_1-\lambda t_2)+\sum_{i=1}^m u_i^*\big(\varphi_i(x)-t_1\big)+\sum_{i=1}^m v_i^*\big(t_2-\psi_i(x)\big).$$

Так как $u_i^*\ge0$ и $v_i^*\ge0$ для всех $i\in I$, то для любых x,t_1,t_2 и λ, удовлетворяющих ограничениям (6.132)–(6.134) имеем $\lambda^*\le\lambda$, т.е. решение

$(x^*, t_1^*, t_2^*, \lambda^*)$ является оптимальным для задачи (6.131)–(6.136), и одновременно с этим, x^* является оптимальным решением задачи (6.119), (6.120). Теорема доказана.

Рассмотрим теперь задачу (6.139), т.е. при фиксированных значениях двойственных переменных $u = \overline{u}$, $v = \overline{v}$ и $\beta = \overline{\beta}$ необходимо решить следующую задачу:

$$\lambda(1 - t_2\overline{\beta}) + \alpha_1 t_1 + \alpha_2 t_2 + \sum_{i=1}^{m} \big(\overline{u}_i \varphi_i(x) - \overline{v}_i \psi_i(x)\big) \to \min, \tag{6.140}$$

$$h_k(x) \le 0 \quad (k = \overline{1,p}), \tag{6.141}$$

$$\lambda \ge 0,\ t_1 \ge 0,\ t_2 \ge 0, \tag{6.142}$$

в которой $\alpha_1 = \overline{\beta} - \sum\limits_{i=1}^{m} \overline{u}_i$, $\alpha_2 = -\sum\limits_{i=1}^{m} \overline{v}_i \le 0$.

При фиксированных $\lambda \ge 0$, $t_1 \ge 0$ и $t_2 \ge 0$ задача (6.140)–(6.142) является задачей выпуклого программирования. Тогда решение задачи (6.140)–(6.142) сводится к нахождению такого решения $x^* \in S$, которое в сочетание с некоторыми значениями $\lambda^* \ge 0$, $t_1^* \ge 0$ и $t_2^* \ge 0$, дает минимальное значение функционала (6.140). Для решения задачи (6.140)–(6.142) может быть использован некоторый приближенный алгоритм, который при оптимальных значениях двойственных переменных $u = u^*, v = v^*$ и $\beta = \beta^*$ задачи (6.138) обеспечил-бы получение оптимального решения задачи (6.140)–(6.142). Такой алгоритм может быть построен, если при очередном решении задачи (6.140)–(6.142) фиксировать значения соответствующих параметров по формулам

$$t_1^*\big(x^*(u^{r-1}, v^{r-1}, \beta_{r-1})\big) = \max_{i \in I} \varphi_i\big(x^*(u^{r-1}, v^{r-1}, \beta_{r-1})\big),$$

$$t_2^*\big(x^*(u^{r-1}, v^{r-1}, \beta_{r-1})\big) = \min_{i \in I} \psi_i\big(x^*(u^{r-1}, v^{r-1}, \beta_{r-1})\big),$$

$$\lambda^*\big(x^*(u^{r-1}, v^{r-1}, \beta_{r-1})\big) = \frac{t_1^*\big(x^*(u^{r-1}, v^{r-1}, \beta_{r-1})\big)}{t_2^*\big(x^*(u^{r-1}, v^{r-1}, \beta_{r-1})\big)},$$

где $x^*(u^{r-1}, v^{r-1}, \beta_{r-1}) \in S$ – решение задачи (6.139), найденное на предыдущем шаге субградиентного метода.

Пусть перед выполнением очередного r-го шага субградиентного метода имеем значения переменных $u = u^r, v = v^r, \beta = \beta_r$ и предыдущее решение $\big(x^*(u^{r-1}, v^{r-1}, \beta_{r-1}),\ t_1^*(x^*(u^{r-1}, v^{r-1}, \beta_{r-1})),\ t_2^*(x^*(u^{r-1}, v^{r-1},$

$\beta_{r-1})), \ \lambda^*(x^*(u^{r-1}, v^{r-1}, \beta_{r-1})))$ задачи (6.140)–(6.142). Тогда для решения задачи (6.140)–(6.142) на r-м шаге субградиентного метода используем следующий порядок вычислений.

1. Находим оптимальное решение $x^*(u^r, v^r, \beta_r)$ задачи выпуклого программирования

$$\sum_{i=1}^{m} \big(u_i^r \varphi_i(x) - v_i^r \psi_i(x)\big) \to \min, \tag{6.143}$$

$$h_k(x) \le 0 \quad (k = \overline{1,p}). \tag{6.144}$$

2. Находим новые значения параметров t_1, t_2 и λ по формулам

$$t_1^*\big(x^*(u^r, v^r, \beta_r)\big) = \varphi\big(x^*(u^r, v^r, \beta_r)\big) = \max_{i\in I} \varphi_i\big(x^*(u^r, v^r, \beta_r)\big),$$

$$t_2^*\big(x^*(u^r, v^r, \beta_r)\big) = \psi\big(x^*(u^r, v^r, \beta_r)\big) = \min_{i\in I} \psi_i\big(x^*(u^r, v^r, \beta_r)\big),$$

$$\lambda^*\big(x^*(u^r, v^r, \beta_r)\big) = \frac{t_1^*\big(x^*(u^r, v^r, \beta_r)\big)}{t_2^*\big(x^*(u^r, v^r, \beta_r)\big)}.$$

Пусть на последнем r-м шаге субградиентного метода $u^* = u^r, \ v^* = = v^r$ и $\beta^* = \beta_r$ являются оптимальным решением задачи (6.138). Находим x^*– оптимальное решение задачи (6.143), (6.144) и определяем значения $t_1^* = \varphi(x^*)$, $t_2^* = \psi(x^*)$ и $\lambda^* = t_1^*/t_2^*$. Тогда $(x^*, t_1^*, t_2^*, \lambda^*)$ будет оптимальным решением задачи (6.131)–(6.136), а x^* – оптимальным решением задачи (6.119), (6.120).

6. Задача оптимизации отношения максимума и минимума линейных функций. Рассмотрим теперь случай, когда функции

$$\{\varphi_i(x),\, i \in I\}; \ \{\psi_i(x),\, i \in I\} \ \text{ и } \ \{h_k(x) \ \ (k = \overline{1,p})\}$$

являются линейными. Тогда получим следующую обобщенную дробно-линейную задачу:

$$\min_{x\in S} \left[f(x) = \frac{\varphi(x)}{\psi(x)} = \frac{\max\limits_{i\in I} \varphi_i(x)}{\min\limits_{i\in I} \psi_i(x)} \right], \tag{6.145}$$

в которой

$$x = (x_1, x_2, \ldots, x_n) \in R^n, \quad I = \{1, 2, \ldots, m\},$$

$$\varphi_i(x) = \sum_{j=1}^{n} c_{ij} x_j + c_i^0, \quad \psi_i(x) = \sum_{j=1}^{n} d_{ij} x_j + d_i^0,$$

$$\varphi(x) = \max_{i\in I} \varphi_i(x), \quad \psi(x) = \min_{i\in I} \psi_i(x),$$

а $S = \Big\{x : \sum_{j=1}^{n} a_{kj}x_j \le b_k \quad (k = \overline{1,p}); \quad x_j \ge 0 \quad (j = \overline{1,n})\Big\}$ является выпуклым многогранным множеством, притом $\varphi_i(x) \ge 0, \quad i \in I$, а $\psi_i(x) > 0, \quad i \in I$ для любого $x \in S$.

Так как функции $\varphi_i(x)$ и $\psi_i(x)$ – линейны, то функции $\varphi(x)$ и $\psi(x)$ являются кусочно-линейными и недифференцируемыми, притом $\varphi(x) = \max_{i\in I} \varphi_i(x)$ – выпукла, а $\psi(x) = \min_{i\in I} \psi_i(x)$ – вогнута. Тогда задача (6.145) является задачей оптимизации дробно-выпуклого, кусочно-линейного, не везде дифференцируемого функционала на выпуклом многограннике. Так как дробный функционал $f(x) = \dfrac{\varphi(x)}{\psi(x)}$ является квазивыпуклым, то задача (6.145) является одноэкстремальной, для которой минимальное значение функционала необязательно достигается в крайней точке многогранника S.

Задача (6.145) является задачей обобщенного дробного программирования в виде задачи минимизации отношения кусочно-линейных функций при линейных ограничениях. Она может быть записана в эквивалентной форме в виде задачи билинейного программирования следующего вида

$$\lambda \to \min,$$

$$\varphi_i(x) - t_1 \le 0, \quad i \in I,$$

$$t_2 - \psi_i(x) \le 0, \quad i \in I,$$

$$t_1 - \lambda t_2 = 0,$$

$$x \in S,$$

в которой имеется билинейное ограничение, содержащее произведение переменных λ и t_2.

Предположим, что линейные функции $\varphi_i(x) \ge 0$, а $\psi_i(x) > 0, \quad i \in I$ для любого $x \in S$. Тогда функции $\varphi(x) \ge 0$ и $\psi(x) > 0$ для любого $x \in S$, причем $\varphi(x)$ является кусочно-линейной и выпуклой функцией, а $\psi(x)$ – кусочно-линейной и вогнутой функцией, и в свою очередь, функция $f(x)$ является квазивыпуклой на S. Для решения задачи (6.145) или для ее эквивалентной задачи билинейного программирования могут быть использованы рассмотренные выше методы нелинейного случая, а именно субградиентные методы решения прямой и двойственной задачи, параметрический прямой и двойственной метод, алгоритм частичной линеаризации, алгоритмы субградиентного метода и схемы декомпозиции по ограничениям, в которых учитывается линейность функций

$\varphi_i(x), \psi_i(x)$ и $h_k(x)$. В данном случае внутренние задачи в итеративных методах будут задачами линейного программирования.

A. *Двойственная задача.* В качестве двойственной для задачи (6.145) может быть использована следующая задача дробного билинейного программирования

$$v^* = f(x^*) = \min_{x \in S} \frac{\varphi(x)}{\psi(x)} = \frac{\max\limits_{i \in I} \sum\limits_{j=1}^{n} c_{ij} x_j + c_i^0}{\min\limits_{i \in I} \sum\limits_{j=1}^{n} d_{ij} x_j + d_i^0} =$$

$$= \max_{\substack{u \in U \\ v \in V}} \min_{x \in S} \frac{\sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} c_{ij} x_j u_i + \sum\limits_{i=1}^{m} c_i^0 u_i}{\sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} d_{ij} x_j v_i + \sum\limits_{i=1}^{m} d_i^0 v_i} = \max_{\substack{u \in U \\ v \in V}} \min_{x \in S} \frac{C(x,u)}{D(x,v)}, \tag{6.146}$$

в которой

$$U = \Big\{ u_i \colon \ \sum_{i=1}^{m} u_i = 1, \quad u_i \geq 0, \quad i \in I, \Big\},$$

$$V = \Big\{ v_i \colon \ \sum_{i=1}^{m} v_i = 1, \quad v_i \geq 0, \quad i \in I, \Big\},$$

а билинейные функции $C(x,u)$ и $D(x,v)$ имеют вид

$$C(x,u) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_j u_i + \sum_{i=1}^{m} c_i^0 u_i,$$

$$D(x,v) = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_j v_i + \sum_{i=1}^{m} d_i^0 v_i.$$

Определим квазивыпуклую и недифференцируемую функцию

$$\Theta(u,v) = \min_{x \in S} \frac{C(x,u)}{D(x,v)}, \tag{6.147}$$

и рассмотрим задачу

$$\max_{\substack{u \in U \\ v \in V}} \Theta(u,v), \tag{6.148}$$

которая является двойственной для задачи (6.145).

Для решения задачи (6.148) можно применить субградиентный метод как для задачи недифференцируемой оптимизации, или же двойственный параметрический метод, если ее рассматривать как дробную задачу.

Если для решения двойственной задачи (6.148) применить субградиентные методы, то на каждом его r-м шаге, при фиксированных значениях $u = u^r$ и $v = v^r$, решается задача дробно-линейного программирования $\min\limits_{x\in S} \dfrac{\sum\limits_{j=1}^{n} \alpha_j^r x_j \;+\; \alpha_0^r}{\sum\limits_{j=1}^{n} \beta_j^r x_j \;+\; \beta_0^r}$, в которой

$$\alpha_j^r = \sum_{i=1}^{m} c_{ij} u_i^r, \quad \beta_j^r = \sum_{i=1}^{m} d_{ij} v_i^r \quad (j = \overline{1,n}),$$

$$\alpha_0^r = \sum_{i=1}^{m} c_i^0 u_i^r, \quad \beta_0^r = \sum_{i=1}^{m} d_i^0 v_i^r.$$

Если же для решения двойственной задачи (6.148) в виде дробной задаче (6.146) применить параметрический метод, то при фиксированных u^r, v_r и λ_r вместо задачи дробно-линейного программирования $\min\limits_{x\in S} \dfrac{\sum\limits_{j=1}^{n} \alpha_j^r x_j + \alpha_0^r}{\sum\limits_{j=1}^{n} \beta_j^r x_j + \beta_0^r}$ решаем задачу линейного программирования $\min\limits_{x\in S} \sum\limits_{j=1}^{n} \gamma_j^r x_j$, где $\gamma_j^r = \alpha_j^r - \lambda_r \beta_j^r \quad (j = \overline{1,n})$.

Б. *Параметрический метод.* В параметрическом методе задача (6.145) сводится к решению следующей задачи параметрического программирования:

$$\begin{aligned} F(\lambda) = \min_{x\in S} \Big[Z(x,\lambda) &= \varphi(x) - \lambda\psi(x) = \\ &= \max_{i\in I} \Big(\sum_{j=1}^{n} c_{ij} x_j + c_i^0 \Big) - \lambda \min_{i\in I} \Big(\sum_{j=1}^{n} d_{ij} x_j + d_i^0 \Big) \Big], \end{aligned} \tag{6.149}$$

которая представляет собой параметрическую задачу дискретного минимакса с билинейным функционалом, в которой имеется произведение переменных λ и x_j. Однако, если фиксировать параметр λ, то получим обычную задачу линейного дискретного минимакса. Для определения значений параметра λ используем алгоритм параметрического метода.

В алгоритме параметрического метода необходимо выполнить следующие шаги.

Шаг 0. Находим допустимую точку $x^0 \in S$. Полагаем

$$\lambda_1 = f(x^0) = \frac{\varphi(x^0)}{\psi(x^0)} = \frac{\max\limits_{i\in I} \varphi_i(x^0)}{\min\limits_{i\in I} \psi_i(x^0)} \quad \text{и} \quad r = 1.$$

Шаг r. Пусть $x^{r-1} \in S$ решение задачи (6.149), полученное на предыдущем шаге при фиксированном значении параметра λ. Находим оптимальное решение x^r задачи (6.149) при $\lambda_r = f(x^{r-1})$ и если $F(\lambda_r) = 0$, то x^r является оптимальным решением задачи (6.145).

При $\lambda = \lambda_r$ задача дискретного линейного минимакса (6.149) сводится к эквивалентной задаче линейного программирования следующего вида:

$$t_1 - \lambda_r t_2 \to \min, \tag{6.150}$$

$$\sum_{j=1}^{n} c_{ij} x_j - t_1 \leq -c_i^0 \quad (i = \overline{1, m}), \tag{6.151}$$

$$\sum_{j=1}^{n} d_{ij} x_j - t_2 \geq -d_i^0 \quad (i = \overline{1, m}), \tag{6.152}$$

$$x \in S, \tag{6.153}$$

$$t_1 \geq 0, \quad t_2 \geq 0. \tag{6.154}$$

Таким образом, решение задачи минимизации отношения максимума и минимума линейных функций при линейных ограничениях (6.145) сводится к решению последовательности задач линейного программирования (6.150)–(6.154), полученных в результате применения параметрического метода решения задачи (6.149), что соответствует задаче нахождения корня уравнения $F(\lambda) = 0$.

В. *Метод частичной линеаризации.* Задача (6.145) переписывается в следующей эквивалентной форме:

$$\lambda \to \min, \tag{6.155}$$

$$\sum_{j=1}^{n} c_{ij} x_j - t_1 \leq -c_i^0 \quad (i = \overline{1, m}), \tag{6.156}$$

$$\sum_{j=1}^{n} d_{ij} x_j - t_2 \geq -d_i^0 \quad (i = \overline{1, m}), \tag{6.157}$$

$$t_1 - \lambda t_2 = 0, \tag{6.158}$$

$$x \in S, \tag{6.159}$$

которая является билинейной задачей из-за произведения $t_2 \cdot \lambda$ переменных t_2 и λ в равенстве (6.158). Для ее решения используется итеративный алгоритм с применением метода частичной линеаризации на каждом его шаге, который заключается в замене произведения $t_2 \cdot \lambda$ на его линейное приращение в точке $(x^r, t_1^r, t_2^r, \lambda_r)$ путем фиксирования значения переменной $\lambda = \lambda_r$, определенного на предыдущем шаге и за счет полученного решения x^r по переменным x.

В данном случае функции частичной линеаризации $H^r(x, t_1, t_2, \lambda)$ в точке $(x^r, t_1^r, t_2^r, \lambda_r)$ на каждом шаге имеют вид:

$$H^r(x, t_1, t_2, \lambda) = t_1 - \lambda_r t_2 - (\lambda - \lambda_r)t_2^r.$$

Алгоритм решения задачи (6.145) по такому параметрическому методу содержит следующие шаги.

Шаг 0. Пусть $x^0 \in S$ и $\lambda_0 = f(x^0) = \dfrac{\varphi(x^0)}{\psi(x^0)} = \dfrac{\max\limits_{i\in I} \varphi_i(x^0)}{\min\limits_{i\in I} \psi_i(x^0)}$. Тогда точка $(x^0, t_1^0, t_2^0, \lambda_0)$ является допустимым решением задачи (6.155)–(6.159). Полагаем $r = 1$.

Шаг 1. Находим оптимальное решение $(x^r, t_1^r, t_2^r, \lambda_r)$ следующей задачи линейного программирования, полученной в результате частичной линеаризации задачи (6.155)–(6.159) в точке $(x^{r-1}, t_1^{r-1}, t_2^{r-1}, \lambda_{r-1})$:

$$\lambda \to \min, \tag{6.160}$$

$$\sum_{j=1}^{n} c_{ij} x_j - t_1 \leq -c_i^0 \quad (i = \overline{1, m}), \tag{6.161}$$

$$\sum_{j=1}^{n} d_{ij} x_j - t_2 \geq -d_i^0 \quad (i = \overline{1, m}), \tag{6.162}$$

$$t_1 - \lambda_r t_2 - \lambda = -\lambda_r t_2^r, \tag{6.163}$$

$$x \in S. \tag{6.164}$$

Шаг 2. Если $\lambda_r = \lambda_{r-1}$, то x^r является оптимальным решением задачи (6.145). Останов. В противном случае переходим на шаг 3.

Шаг 3. Находим $t_1^r = \varphi(x^r) = \max\limits_{i\in I} \varphi_i(x^r)$, $t_2^r = \psi(x^r) = \min\limits_{i\in I} \psi_i(x^r)$ и $\lambda_r = f(x^r) = \dfrac{t_1^r}{t_2^r}$, полагаем $r = r + 1$ и переходим к первому шагу алгоритма.

В алгоритмах **Б** и **В** решается последовательность линейных задач (6.150)–(6.154) и (6.160)–(6.164), в которых к первоначальным ограничениям прибавляются еще $2m+1$ ограничение. Наличие таких ограничений не позволяет применить для их решения специальные алгоритмы, когда первоначальные ограничения задачи (6.145) имеют специальную структуру (например, блочно-диагональную или транспортного типа).

Г. *Схема декомпозиции по ограничениям.* Для решения задачи (6.145) специальной структуры или большой размерности можно применить алгоритм, основанный на схеме декомпозиции по ограничениям и субградиентном методе недифференцируемой оптимизации.

Для задачи (6.145) с учетом ее эквивалентной задачи (6.160)–(6.164) построим функцию Лагранжа по формуле

$$L(x, t_1, t_2, \lambda, u, v, \beta) =$$

$$= \lambda + \beta(t_1 - \lambda t_2) + \sum_{i=1}^{m} u_i \left(\sum_{j=1}^{n} (c_{ij}x_j - t_1 + c_i^0 \right) + \sum_{i=1}^{m} v_i \left(t_2 - \sum_{j=1}^{n} (d_{ij}x_j - d_i^0 \right),$$

в которой $u = \{u_i \geq 0, \quad i = \overline{1,m}\}$, $v = \{v_i \geq 0, \ i = \overline{1,m}\}$ и β – множители Лагранжа, соответственно для ограничений (6.161), (6.162) и (6.163).

Тогда задача (6.145) сводится к решению задач

$$\max_{\substack{u \geq 0 \\ v \geq 0}} L^*(u, v, \beta), \tag{6.165}$$

$$L^*(u, v, \beta) = \min_{\substack{x \in S, \lambda \geq 0 \\ t_1 \geq 0, t_2 \geq 0}} L(x, t_1, t_2, \lambda, u, v, \beta). \tag{6.166}$$

Функция $L^*(u, v, \beta)$ определена для любого $u \geq 0, v \geq 0$ и β, является кусочно-линейной, вогнутой и недифференцируемой. Для ее максимизации используется один из алгоритмов субградиентного метода, на каждом шаге которого решается задача (6.166) при фиксированных значениях переменных u, v и β. Тогда на $(r+1)$-м шаге субградиентного метода необходимо выполнить следующие три основных этапа.

1. Решить задачу (6.166) при $u = u^r$, $v = v^r$ и $\beta = \beta_r$ и находится оптимальное решение $\big(x^*(u^r, v^r, \beta_r), t_1^*(u^r, v^r, \beta_r), t_2^*(u^r, v^r, \beta_r), \lambda^*(u^r, v^r, \beta_r)\big)$.

2. Определить значения обобщенного градиента функции $L^*(u, v, \beta)$ в точке $u = u^r$, $v = v^r$, $\beta = \beta_r$ по формулам

$$g_i(u^r) = \sum_{j=1}^{n} c_{ij} x_j^*(u^r, v^r, \beta_r) + c_i^0 - t_1^*(u^r, v^r, \beta_r), \quad i \in I,$$

$$g_i(v^r) = t_2^*(u^r, v^r, \beta_r) - \sum_{j=1}^{n} d_{ij} x_j^*(u^r, v^r, \beta_r) - d_i^0, \quad i \in I,$$

$$g(\beta_r) = t_1^*(u^r, v^r, \beta_r) - \lambda^*(u^r, v^r, \beta_r) \cdot t_2^*(u^r, v^r, \beta_r).$$

3. Найти новые значения переменных u, v и β по формулам

$$u_i^{r+1} = \max\{0, u_i^r + h_{r+1} g_i(u^r)\}, \quad i \in I,$$

$$v_i^{r+1} = \max\{0, v_i^r + h_{r+1} g_i(v^r)\}, \quad i \in I,$$

$$\beta_{r+1} = \beta_r + h_{r+1} g(\beta_r),$$

где h_{r+1} – величина шага.

Если (u^*, v^*, β^*) – оптимальное решение задачи (6.165), то $(x^*, t_1^*, t_2^*, \lambda^*)$ – решение задачи (6.166) при $u = u^*, v = v^*$ и $\lambda = \lambda^*$. Тогда $(x^*, t_1^*, t_2^*, \lambda^*)$ являются оптимальным решением задачи (6.155)–(6.159), а x^* – оптимальным решением задачи (6.145).

Д. *Решение задачи (6.166).* На каждом шаге субградиентного метода при фиксированных значениях двойственных переменных $u = u^r$, $v = v^r$ и $\beta = \beta_r$ решается задача (6.166), которая может быть переписана в следующем виде:

$$\min_{\substack{x \in S, \lambda \geq 0 \\ t_1 \geq 0, t_2 \geq 0}} \left[\lambda + \left(\beta_r - \sum_{i=1}^{m} u_i^r \right) t_1 + \left(\sum_{i=1}^{m} v_i^r - \beta_r \lambda \right) t_2 + \sum_{j=1}^{n} \alpha_j^r x_j + \alpha_0^r \right], \quad (6.167)$$

где $\alpha_j^r = \sum\limits_{i=1}^{m} (c_{ij} u_i^r - d_{ij} v_i^r) \quad (j = \overline{1, n})$, $\alpha_0^r = \sum\limits_{i=1}^{m} (u_i^r c_i^0 - v_i^r d_i^0)$. Задача (6.162) является задачей билинейного программирования из-за произведения $\lambda \cdot t_2$.

При фиксированном $\lambda \geq 0$ задача (6.167) является задачей линейного программирования. Поэтому процесс решения данной задачи сводится к нахождению такого решения $x^* \in S$ и такого значения $\lambda^* \geq 0$, которые в совокупности представляли бы оптимальное решение $(x^*, t_1^*, t_2^*, \lambda^*)$ задачи (6.167). Такой процесс может быть итерационным, на каждой итерации которого приближаемся к значению параметра λ, сходящегося к оптимальному решению $(x^*, t_1^*, t_2^*, \lambda^*)$ задачи (6.167).

Данный эффект имеет место в субградиентных методах, которые устойчивы относительно точности определения значений субградиентов. Поэтому для решения задачи (6.167) на каждом шаге субградиентного метода используем приближенный алгоритм решения задачи (6.167).

Для решения задачи (6.167) может быть использован одноитерационный приближенный алгоритм, в результате которого при фиксированных значениях переменных $u = u^*, v = v^r$ и $\beta = \beta_r$ задаче (6.165) получим оптимальное решение задачи (6.167), а следовательно и исходной задаче (6.145). Такой алгоритм может быть построен, если при решении задачи (6.167) будем фиксировать значения параметров λ, t_1 и t_2 по следующим формулам:

$$t_1^r = \varphi\big(x^*(u^r, v^r, \beta_r)\big), \quad t_2^r = \psi\big(x^*(u^r, v^r, \beta_r)\big), \quad \lambda^r = t_1^r/t_2^r,$$

в которых $x^*(u^r, v^r, \beta_r) \in S$ – решение задачи (6.167), полученное на предыдущем шаге субградиентного метода.

Допустим, что после выполнения r шагов субградиентного метода имеем значения переменных $u = u^r$, $v = v^r$ и $\beta = \beta_r$. Тогда задача (6.167) на очередном шаге субградиентного метода решается в следующем порядке:

- находим оптимальное решение $x^*(u^r, v^r, \beta_r)$ задачи линейного программирования

$$\min_{x \in S} \sum_{j=1}^{n} \alpha_j^r x_j, \tag{6.168}$$

- находим новые значения для переменных t_1, t_2 и λ по формулам:

$$t_1^*(u^r, v^r, \beta_r) = \varphi\big(x^*(u^r, v^r, \beta_r)\big), \quad t_2^*(u^r, v^r, \beta_r) = \psi\big(x^*(u^r, v^r, \beta_r)\big),$$

$$\lambda^*(u^r, v^r, \beta_r) = f\big(x^*(u^r, v^r, \beta^r)\big) = t_1^*/t_2^*,$$

которые будут использованы для определения значений обобщенного градиента.

Полученные значения $\big(x^*(u^r, v^r, \beta_r), \;\; \lambda^*(u^r, v^r, \beta_r), \;\; t_1^*(u^r, v^r, \beta_r), t_2^*(u^r, v^r, \beta_r)\big)$ будут приближенным решением задачи (6.167), которые при $k \to \infty$ в субградиентном методе стремятся к оптимальному решению $(x^*, \lambda^*, t_1^*, t_2^*)$ задачи (6.167).

В отличие от параметрического метода и метода частичной линеаризации, внутренняя задача линейного программирования (6.168) решается на множестве исходных ограничений, что является очень важным фактором при решение задач специальной структуры, таких как транспортного типа, блочной структуры и других задач, для решения которых имеются специальные эффективные методы и алгоритмы.

В предложенном алгоритме на каждом шаге решается задача линейного программирования (6.168), в которой меняются только значения коэффициентов целевой функции, а первоначальные ограничения остаются без изменений. Это обстоятельство имеет важное значение, когда решается задача (6.145) с ограничениями специальной структуры.

6.6. Задача многокритериальной дробно-выпуклой оптимизации

Рассмотрим следующее возможное обобщение задачи дробно-выпуклого программирования в смысле минимизации многокритериального функционала, построенного в виде вектора дробных функций:

$$f(x) = \left(\frac{\varphi_1(x)}{\psi_1(x)}, \frac{\varphi_2(x)}{\psi_2(x)}, \dots, \frac{\varphi_m(x)}{\psi_m(x)}\right) \to \min \tag{6.169}$$

$$h_k(x) \leq 0 \quad (k = \overline{1,p}), \tag{6.170}$$

в которой функции $\{\varphi_i(x),\ \ i \in I\}$, $\{-\psi_i(x),\ \ i \in I\}$ и $\{h_k(x),\ \ k = \overline{1,p}\}$ – непрерывны, выпуклы и недифференцируемы на R^n. Предполагается также, что функции $\varphi_i(x) \geq 0,\ \ i \in I$, а $\psi_i(x) > 0,\ \ i \in I$ для всех $x \in S$, где $S = \{x :\ \ h_k(x) \leq 0,\ \ k = \overline{1,p}\}$, т.е. S – множество решений, удовлетворяющие ограничениям (6.170). Задачу (6.169), (6.170), которую назовем задачей многокритериальной (векторной) дробной оптимизации, рассмотрена в работах [111, 137, 168, 235, 291, 295, 385, 493, 494, 561], для которых может быть применен общий подход их анализа [79, 82, 87, 119, 180, 347, 348, 364, 400, 430, 442, 503, 634, 650, 653, 654] для нахождения эффективных решений. Для дробных задач многокритериальной оптимизации рассмотрены необходимые и достаточные условия оптимальности [54, 138, 298, 299, 340, 343, 346, 398], приведены различные двойственные задачи [54, 56, 57, 91, 114, 116, 138, 179, 279, 292, 298, 299, 301, 321, 322, 340, 343, 346, 374, 386, 398, 412, 534], [536]–[539], рассмотрены недифференцируемые дробные многокритериальные задачи [54, 138, 296, 299, 301, 346, 398], а также предложены различные методы решения исходных многокритериальных задач или их сведения к однокритериальным задачам оптимизации [95, 102, 207, 316, 350, 363, 399, 535].

Для задачи (6.169), (6.170) можно рассматривать ее эквивалентную параметрическую задачу многокритериальной оптимизации

$$F(\lambda) = \min_{x \in S} Z(x, \lambda) = (z_1(x, \lambda_1), z_2(x, \lambda_2), \ldots, z_m(x, \lambda_m)), \qquad (6.171)$$

где $z_i(x, \lambda_i) = \varphi_i(x) - \lambda_i \psi_i(x), \quad i \in I$.

Различные теоретические вопросы, свойства и методы решения задач многокритериальной дробно-выпуклой оптимизации, рассмотрены на основе анализа соответствующих параметрических задач многокритериальной выпуклой оптимизации.

Нами предлагается использовать различные варианты метода сверток, которые сводят решение задачи многокритериальной дробной оптимизации к обобщенным задачам дробного программирования, рассмотренные в данной главе. Приводятся основные понятия эффективных точек для многокритериальных задач и для каждого типа свертки приводится необходимые и достаточные условия получения эффективных решений исходной многокритериальной задачи (6.169), (6.170). Также рассматриваются смешанные многокритериальные задачи, в которых имеются дробные функционалы.

1. Основные определения. Задачи многокритериальной оптимизации рассмотрены одновременно с задачами линейного программирования и задачами теории игр. Основные определения и теоретические результаты приведем в дальнейшем только для дробно-выпуклых функций и многокритериальных дробных задач оптимизации.

Одно из основных понятий в задачах многокритериальной оптимизации является понятие наилучшего решения в смысле его сравнения по векторному критерию $f(x)$, состоящему из m отдельных показателей оценки выбранного решения в качестве оптимального. Поэтому, в таких задачах говорят не об оптимальности полученного решения, а об его эффективности. Для этого решения сравниваются между собой понятиями предпочтения: лучше, хуже, наилучшее, нелучшее, наихудшее, не хуже, или эквивалентны (равноценны). Каждый критерий $f_i(x)$ векторной функции $f(x) = (f_1(x), f_2(x), \ldots, f_m(x))$ называется частным критерием, который в данном случае является дробной функцией. В задачах векторной оптимизации решения оцениваются в совокупности по всем частным критериям. В таких задачах решение x_1 считается не хуже решения x_2, если для всех частных критериев $f_i(x_1) \geq f_i(x_2), \quad i \in I$ и оно не лучше x_2, если $f_i(x_1) \leq f_i(x_2), \quad i \in I$; они эквивалентны, если $f_i(x_1) = f_i(x_2), \quad i \in I$. Решение x^* называется эффективным если для него не существует лучшего.

Поэтому в задачах многокритериальной оптимизации необходимо найти эффективное решение с точки зрения минимизации векторной целевой функции $f(x)$ или же ее максимизации. Также следует отметить, что в смысле эффективных решений (эффективных точек) в задачах векторной оптимизации может выступать не одно отдельное решение, а целое множество таких эффективных решений (точек), которые являются между собой эквивалентными (равноценными). В таких задачах ставится цель, либо нахождения одного представителя множества эффективных точек, либо же описать полное множество таких точек. Эффективные точки также называются оптимальными по Парето, а множество эффективных точек – множеством точек оптимальных по Парето. Также, в теории игр такое множество называют компромиссным или переговорным множеством.

Таким образом, решение x_1 называется эффективным решением задачи векторной оптимизации, если не существует другого решения x_2 лучшее чем x_1, т.е. такое, чтобы $f_i(x_2) \leq f_i(x_1)$, причем хоть для одного частного критерия выполнялось бы строгое неравенство. Другими словами, эффективное решение – это решение неулучшаемое ни по одному из заданных частных критериев.

Для задачи многокритериальной дробной оптимизации (6.169), (6.170) имеем, что точка $x_0 \in S$ оптимальна по Парето (эффективная точка), если не существует другой точки x_1, для которой $\dfrac{\varphi_i(x_1)}{\psi_i(x_1)} \leq$ $\leq \dfrac{\varphi_i(x_0)}{\psi_i(x_0)}$ для любого $i \in I$ и хотя бы для одного $i_0 \in I$ выполняется строгое неравенство $\dfrac{\varphi_{i_0}(x_1)}{\psi_{i_0}(x_1)} < \dfrac{\varphi_{i_0}(x_0)}{\psi_{i_0}(x_0)}$.

Функция Лагранжа. Применение векторной функции Лагранжа при решении задачи многокритериальной дробно-выпуклой задачи (6.169), (6.170) позволяет связать ее с задачей отыскания эффективных седловых пар функции Лагранжа. При этом задача поиска таких пар соответствует нахождению пересечения множества эффективных решений прямой и двойственной задач. Как правило, функция Лагранжа в задачах многокритериальной оптимизации рассматривается для каждого частного критерия, и поэтому она также является многокритериальной, т.е. имеем векторные функции Лагранжа.

Для исходной задачи дробно-выпуклой многокритериальной оптимизации можно построить два типа векторных функций Лагранжа

$$L^r(x,z) = (L_1^r(x,z), L_2^r(x,z), \ldots, L_m^r(x,z)) \quad (r = 1,2),$$

в которых частные критерии определяются по одной из формул

$$L_i^1(x,z) = \frac{\varphi_i(x)}{\psi_i(x)} + \sum_{k=1}^{p} z_k h_k(x), \quad i \in I;$$

$$L_i^2(x,z) = \frac{\varphi_i(x) + \sum\limits_{k=1}^{p} z_k h_k(x)}{\psi_i(x)}, \quad i \in I,$$

в которых $z_k \geq 0 \quad (k = \overline{1,p})$ – множители Лагранжа для ограничений (6.170). Для таких векторных функций Лагранжа рассматриваются задачи нахождения эффективных седловых точек функций $L^1(x,z)$ и $L^2(x,z)$, которые соответствуют задачам

$$\max_{z \geq 0} \min_{x \in S} L^r(x,z) \quad (r = 1,2). \tag{6.172}$$

Для задач (6.172) можно рассматривать две группы задач, которые также являются задачами векторной оптимизации.

Первая группа задач: найти множества Парето-оптимальных решений для векторных задач, которые соответствуют минимизациям функций Лагранжа прямой задачи, т.е. имеем задачи

$$L_r^*(z) = \min_{x \in S} L^r(x,z) \quad (r = 1,2). \tag{6.173}$$

Вторая группа задач: найти множества Парето-оптимальных решений для векторных задач

$$\max_{z \geq 0} L_r^*(z) \quad (r = 1,2), \tag{6.174}$$

в которых $L_r^*(z) = \big(L_{1r}^*(z), L_{2r}^*(z), \ldots, L_{mr}^*(z)\big) \quad (r = 1,2)$, а $L_{ir}^*(z) = L_i^r(x^*,z)$, где x^* – эффективное решение прямой задачи.

Задачи (6.174) являются двойственными задачами для исходной задачи (6.169), (6.170).

Для пары соответствующих двойственных задач (6.173) и (6.174) для $r = 1$ или для $r = 2$ необходимо найти множества Парето оптимальных решений, которые в конечном итоге совпадают и формируют множество пар седловых точек векторной функции Лагранжа $L_1(x,z)$ или $L_2(x,z)$ соответственно. Решение каждой пары двойственных задач заключается в нахождение минимальных точек прямого множества и максимальных точек двойственного множества, соответствующие эффективным седловым точкам функции Лагранжа.

Условия оптимальности. Для задач многокритериальной дробной оптимизации условия оптимальности позволяют разработать различные методы отыскания эффективных решений. На основе выведенных условий оптимальности разрабатываются различные способы проверки эффективности полученных решений эквивалентных однокритериальных или многокритериальных задач. Как отмечено выше, для дробной многокритериальной задачи (6.169), (6.170) рассмотрены различные необходимые и достаточные условия оптимальности [54, 138, 298, 299, 340, 343, 346, 398]. Ниже мы приведем условия оптимальности, связанные с возможными свертками и примененные для задач многокритериальной дробной оптимизации, которые соответствуют рассмотренным выше задач обобщенного дробно-выпуклого программирования.

Метод свертки критериев. Основная трудность при решении задач многокритериальной оптимизации является описание и определение множества эффективных решений [634, 650, 653, 654]. Важной проблемой является и нахождение отдельных точек данного множества. Поэтому, для этих целей, применяются различные методы и приемы. Одним из таких методов является применение процедуры свертки критериев в один (так называемая процедура скаляризации критериев) и сведение задачи многокритериальной оптимизации к решению задачи с одним критерием. В зависимости от метода свертки формируются частные условия оптимальности, которые применяются при разработке конкретных методов решения многокритериальных задач оптимизации.

Если имеем общую задачу многокритериальной оптимизации

$$\min_{x\in S}\big(f_1(x), f_2(x), \ldots, f_m(x)\big), \tag{6.175}$$

в которой функции $f_i(x)$, $\quad i \in I$ – непрерывны и определены в R^n, то возможно применить следующие процедуры свертки сведения ее к задачам однокритериальной (скалярной) оптимизации:

- линейные (аддитивные) свертки

$$G(x) = \sum_{i\in I} \alpha_i f_i(x);$$

- минимизационные свертки

$$G(x) = \min_{i\in I}\big\{\alpha_i f_i(x) + \beta_i\big\};$$

- максимизационные свертки

$$G(x) = \max_{i\in I}\big\{\alpha_i f_i(x) + \beta_i\big\};$$

- мультипликативные свертки

$$G(x) = \prod_{i \in I} \alpha_i f_i(x);$$

- полиномиальные свертки

$$G(x) = \prod_{i \in I} [\alpha_i f_i(x)]^{\beta_i}.$$

Для коэффициентов свертки $\alpha_i, \ \ i \in I$, как правило, требуется чтобы $\alpha_i > 0, \ \ i \in I$ и $\sum\limits_{i \in I} \alpha_i = 1$. Коэффициенты $\beta_i > 0, \ \ i \in I$.

Рассмотрим применение метода сверток для решения многокритериальных задач дробно-выпуклой оптимизации. В зависимости от примененной свертки получаем различные задачи обобщенного дробного программирования. Такие задачи были рассмотрены в предыдущих параграфах данной главы, для решения которых предложены различные методы и алгоритмы. Также следует отметить, что данные свертки применимы как для решения дробно-выпуклых многокритериальных задач, так и для задачах с дробно-линейными частными критериями, в многокритериальных транспортных задачах с дробными частными критериями, в дробных задачах блочной структуры и в задачах дробного сепарабельного программирования.

Рассмотрим некоторые возможные свертки, применимые для решения многокритериальных задач дробной оптимизации.

а) Линейная свертка.

Линейная свертка является наиболее наглядным методом решения общей задачи многокритериальной оптимизации (6.175). Введем вектор весовых коэффициентов $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$, характеризующие каждый критерий в отдельности, удовлетворяющие условиям $\alpha_i > 0$ $(i = \overline{1, m})$, $\sum\limits_{i=1}^{m} \alpha_i = 1$, и назовем их коэффициентами свертки. Тогда многокритериальная задача (6.175) сводится к решению скалярной задачи

$$\min_{x \in S} \sum_{i=1}^{m} \alpha_i f_i(x). \tag{6.176}$$

В результате решения задачи (6.176) получаем оптимальную точку по Парето для задачи (6.175).

Основным достоинством линейной свертки является то, что с ней связаны классические достаточные и необходимые условия оптимальности по Парето, основанные на следующих двух теоремах.

Теорема 6.16. *В задаче* (6.175) *точка* $x^0 \in S$ *оптимальна по Парето, если существует вектор весовых коэффициентов* $\alpha^0 = \{\alpha_i^0 \geq 0,\ i \in I\}$, *для которого выполняется соотношение*

$$\sum_{i=1}^{m} \alpha_i^0 f_i(x^0) = \min_{x \in S} \sum_{i=1}^{m} \alpha_i^0 f_i(x),$$

в которой x^0 *является оптимальным решением задачи* (6.176) *при* $\alpha = \alpha^0$.

Теорема 6.17. *Если в задаче* (6.175) *точка* $x^0 \in S$ *оптимальна по Парето, то существует вектор весовых коэффициентов* $\alpha^0 = \{\alpha_i^0 \geq 0, \quad i \in I\}$, $\sum\limits_{i=1}^{m} \alpha_i^0 = 1$, *для которого выполняется соотношение*

$$\min_{\alpha} \sum_{i=1}^{m} \alpha_i f_i(x^0) = \sum_{i=1}^{m} \alpha_i^0 f_i(x^0).$$

В дальнейшем линейную свертку применяем для решения многокритериальных дробных задач (6.169), (6.170).

Имеет место теорема, которая подтверждает линейную свертку для дробной задачи многокритериальной оптимизации.

Теорема 6.18. *Пусть* S *выпуклое множество, а* $f_i(x) = \dfrac{\varphi_i(x)}{\psi_i(x)}$ – *квазивыпуклые на* S *функции. Тогда* $x^0 \in S$ *является эффективной точкой для задачи векторной дробной оптимизации* (6.169), (6.170), *тогда и только тогда, когда* x^0 *является оптимальным решением задачи*

$$\min_{x \in S} \sum_{i \in I} \alpha_i \frac{\varphi_i(x)}{\psi_i(x)},$$

для некоторого $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m) > 0, \quad \sum\limits_{i \in I} \alpha_i = 1$.

Применение линейной (аддитивной) свертки для решения многокритериальной дробно-выпуклой задачи приводит к обобщенной задаче дробного программирования в виде суммы дробно-выпуклых функций, которая рассмотрена в п. 6.3. и для решения которой можно использовать уже рассмотренные методы и алгоритмы.

б) Минимизационные и максимизационные свертки.

Для многокритериальных задач используют предложенную Гермейером [650] свертку, которая для задачи дробной оптимизации представлена в виде следующей теоремы.

Теорема 6.19. *Пусть S – выпуклое множество, а $f_i(x) = \dfrac{\varphi_i(x)}{\psi_i(x)}$ – квазивыпуклые на S функции. Тогда $x^0 \in S$ является эффективной точкой для задачи векторной дробной оптимизации* (6.169), (6.170), *тогда и только тогда, когда x^0 является оптимальным решением задачи*

$$\min_{x\in S}\max_{i\in I}\alpha_i\frac{\varphi_i(x)}{\psi_i(x)},$$

где $\alpha_i > 0, \quad i \in I, \quad \sum\limits_{i\in I}\alpha_i = 1.$

в) Мультипликативная свертка.

Теорема 6.20. *Пусть S – выпуклое множество, а $f_i(x) = \dfrac{\varphi_i(x)}{\psi_i(x)}$ – квазивыпуклые на S, $\varphi_i(x) > 0$, $\psi_i(x) > 0$ для любых $x \in S$. Тогда $x^0 \in S$ является эффективной точкой для задачи векторной дробной оптимизации* (6.169), (6.170), *тогда и только тогда, когда x^0 является оптимальным решением задачи*

$$\max_{x\in S}\prod_{i\in I}\alpha_i\frac{\varphi_i(x)}{\psi_i(x)},$$

где $\alpha_i > 0,\ i \in I, \quad \sum\limits_{i\in I}\alpha_i = 1.$

г) Смешанные свертки.

Метод сверток для задачи дробно-выпуклой оптимизации может применяться как для исходной задачи (6.169), (6.170), так и для ее эквивалентной параметрической задачи (6.171). Также могут быть построены и другие эквивалентные задачи, которые связаны с дробным программированием, одна из которых получена при применение дробной свертки. В такой свертке многокритериальная дробная задача сводится к задаче дробного программирования с одним критерием.

Для этих целей строятся различные функции, к которым сводится задача дробно-выпуклой многокритериальной оптимизации. Первая функция $g(x, \alpha)$ соответствует применению свертки непосредственно к исходной многокритериальной дробной задачи. В результате применения линейной свертки, функция $g(x,\alpha)$ соответствует одному из классов обобщенных задач дробно-выпуклого программирования.

Вторая функция $G(x,\alpha)$ соответствует дробной свертке, т.е. когда дробная многокритериальная задача сводится к задаче дробного программирования, в которой числитель и знаменатель определяется соответствующей процедурой свертки.

Третья функция $Z(x,\alpha,\lambda)$ соответствует примененной процедуре свертки для эквивалентной задачи многокритериальной параметрической задачи. В данном случае, процедура свертки применяется для выпуклых параметрических функций.

Полученные функции $g(x,\alpha)$, $G(x,\alpha)$ и $Z(x,\alpha,\lambda)$ используются в дальнейшем для построения смешанных сверток.

Рассмотрим основные возможные функции, получаемые в результате применения различных процедур сверток.

1. Аддитивные функции

$$g_1(x,\alpha) = \sum_{i\in I} \alpha_i \frac{\varphi_i(x)}{\psi_i(x)}, \quad G_1(x,\alpha) = \frac{\sum\limits_{i\in I} \alpha_i\varphi_i(x)}{\sum\limits_{i\in I} \alpha_i\psi_i(x)},$$

$$Z_1(x,\alpha,\lambda) = \sum_{i\in I} \alpha_i \left(\varphi_i(x) - \lambda_i\psi_i(x)\right),$$

в которых $\alpha_i > 0,\ i \in I,\quad \lambda_i \geq 0,\ i \in I,\quad \sum\limits_{i\in I} \alpha = 1.$

2. Функции максимума и минимума

$$g_2(x,\alpha) = \max_{i\in I} \alpha_i \frac{\varphi_i(x)}{\psi_i(x)}, \quad G_2(x,\alpha) = \frac{\max\limits_{i\in I} \alpha_i\varphi_i(x)}{\min\limits_{i\in I} \alpha_i\psi_i(x)},$$

$$Z_2(x,\alpha,\lambda) = \max_{i\in I} \alpha_i \left(\varphi_i(x) - \lambda_i\psi_i(x)\right),$$

в которых $\alpha_i > 0,\quad i \in I;\ \lambda_i \geq 0,\quad i \in I,\quad \sum\limits_{i\in I} \alpha = 1.$

3. Мультипликативные функции

$$g_3(x,\alpha) = \prod_{i\in I} \alpha_i \frac{\varphi_i(x)}{\psi_i(x)}, \quad G_3(x,\alpha) = \frac{\prod\limits_{i\in I} \alpha_i\varphi_i(x)}{\prod\limits_{i\in I} \alpha_i\psi_i(x)},$$

$$Z_3(x,\alpha,\lambda) = \prod_{i\in I} \alpha_i \left(\varphi_i(x) - \lambda_i\psi_i(x)\right),$$

в которых $\alpha_i > 0,\quad i \in I;\ \lambda_i \geq 0,\quad i \in I,\quad \sum\limits_{i\in I} \alpha = 1.$

Тогда задача дробно-выпуклой многокритериальной оптимизации (6.169), (6.170) может быть сведена к одной из следующих трех типов однокритериальных задач:

$$\min_{x\in S} g_r(x,\alpha), \quad \min_{x\in S} G_r(x,\alpha), \quad \min_{x\in S} Z_r(x,\alpha,\lambda) \;\; (r=1,2,3), \tag{6.177}$$

которые соответствуют примененной процедуре свертки и рассмотренных выше задач обобщенного дробно-выпуклого программирования. Выбор той или иной однокритериальной задачи зависит от структуры и свойства функций исходной задачи многокритериальной дробной оптимизации (6.169), (6.170).

Однокритериальные задачи (6.177) могут быть использованы для каждой из них в отдельности или в совокупности для каждого класса функций $g(x,\alpha)$, $G(x,\alpha)$ или $Z(x,\alpha,\lambda)$. Таким образом, получаем так-называемые смешанные многокритериальные задачи, в зависимости от применяемых процедур свертки. В данном случае имеем векторные задачи с тремя частными критериями:

$$\min_{x\in S} \left(g_1(x,\alpha),\; g_2(x,\alpha),\; g_3(x,\alpha)\right),$$

$$\min_{x\in S} \left(G_1(x,\alpha),\; G_2(x,\alpha),\; G_3(x,\alpha)\right),$$

$$\min_{x\in S} \left(Z_1(x,\alpha,\lambda),\; Z_2(x,\alpha,\lambda),\; Z_3(x,\alpha,\lambda)\right).$$

Обобщенные задачи многокритериальной дробной оптимизации.

Изложенное выше указывает на необходимость рассмотрения и исследования обобщенных дробных задач многокритериальной оптимизации. В таких задачах предполагается, что частные критерии являются обобщенными дробными функциями, определенными одной из рассмотренных выше операций над дробными функциями, а именно операции взятия максимума, суммирования или мультипликатирования.

Рассмотрим многокритериальные задачи, с частными критериями в виде обобщенных дробных функций

$$\min_{x\in S}\{f_i^r(x), \quad i\in I\} \quad (r=\overline{1,3}), \tag{6.178}$$

в которых функции $f_i^r(x)$, $\; i\in I$ являются обобщенными дробными функциями одного из следующих типов:

- многокритериальные, в которых частные критерии являются дискретным максимумом дробно-выпуклых функций

$$f_i^1 x) = \max_{j\in J_i} \frac{\varphi_{ij}(x)}{\psi_{ij}(x)}, \quad i \in I;$$

- многокритериальные аддитивные дробно-выпуклые функции

$$f_i^2(x) = \sum_{j\in J_i} \frac{\varphi_{ij}(x)}{\psi_{ij}(x)}, \quad i \in I;$$

- многокритериальные мультипликативные дробно-выпуклые функции

$$f_i^3(x) = \prod_{j\in J_i} \frac{\varphi_{ij}(x)}{\psi_{ij}(x)}, \quad i \in I,$$

в которых функции $\varphi_{ij}(x)$, $-\psi_{ij}(x)$ – выпуклы на S, $j \in J_i$, $i \in I$, а множества J_i содержат различные количества дробных функций. Если $||J_i|| = 1$, то имеем обычную задачу многокритериальной дробной оптимизации типа (6.169), (6.170).

Одновременно с задачами (6.178) может быть рассмотрена и смешанная трехкритериальная обобщенная задача

$$\min_{x\in S}\{f_i^1(x), f_i^2(x), f_i^3(x), \quad i \in I\}, \tag{6.179}$$

в которых функции частных критериев являются обобщенными дробными функциями.

Задачи (6.179) могут быть рассмотрены в виде параметрической трехкритериальной выпуклой задачи

$$\min_{x\in S}\{Z_i^1(x,\lambda), Z_i^2(x,\lambda), Z_i^3(x,\lambda), \quad i \in I\}, \tag{6.180}$$

$$Z_i^1(x,\lambda) = \max_{j\in J_i} \left(\varphi_{ij}(x) - \lambda_{ij}\psi_{ij}(x)\right), \quad i \in I,$$

$$Z_i^2(x,\lambda) = \sum_{j\in J_i} \left(\varphi_{ij}(x) - \lambda_{ij}\psi_{ij}(x)\right), \quad i \in I,$$

$$Z_i^3(x,\lambda) = \prod_{j\in J_i} \left(\varphi_{ij}(x) - \lambda_{ij}\psi_{ij}(x)\right), \quad i \in I.$$

Приведенные обобщенные задачи многокритериальной дробной оптимизации (6.178), (6.179) и (6.180) требуют дополнительных исследований, для которых необходимо определить условия оптимальности, рассмотреть двойственные задачи, построить функции Лагранжа и разработать новых методов и алгоритмов их решения. Такие задачи могут быть рассмотрены, когда входящие в них функции являются выпуклыми, линейными или дробными.

Задачи многокритериальной оптимизации с неоднотипными функциями.

Задачи многокритериальной оптимизации могут быть усложнены, если в качестве частных критериев использовать различные типы функций, соответствующие разным задачам оптимизации. Таких функций и задач объединяет только единственная область оптимизации. В таких задачах не требуется, чтобы все частные критерии соответствовали одному классу задач оптимизации, например, линейному, выпуклому, квадратичному или дробному программированию. Таким образом, задача многокритериальной оптимизации может содержать неоднотипные частные критерии и соответствующие им функции.

Рассмотрим общую формулировку многокритериальной задачи с неоднотипными функциями

$$\min_{x \in S} \left\{ \{ f_i^j(x), \quad j \in J_i \}, \quad i \in I \right\}, \tag{6.181}$$

в которой:

j – номер функции из соответствующего i-го класса задач оптимизации,

$f_i^j(x)$ – определенная в R^n j-я функция i-го класса задач оптимизации,

J_i – множество функций i-го класса задач оптимизации,

I – множество классов задач оптимизации, функции из которых входят в многокритериальную неоднородную задачу.

В качестве функции для частных критериев (многокритериальной задачи (6.181) могут быть взяты любые целевые функции задач математического программирования. Рассмотрим неоднотипные многокритериальные задачи, в которых функции частных критериев выбираются с точки зрения дробности таких функций. Тогда в многокритериальной смешанной задаче могут входить функции трех типов, а именно:

- линейные и дробно-линейные функции;
- выпуклые и дробно-выпуклые функции;
- квадратичные и дробно-квадратичные функции.

Также могут быть рассмотрены функции, полученные в результате применения операций суммирования, произведения или взятие максимума на множестве тех же классов задач оптимизации.

Рассмотрим следующую задачу многокритериальной оптимизации с двумя типами функций:

$$\min_{x\in S}\left\{f_j(x),\ j\in J;\quad \frac{\varphi_i(x)}{\psi_i(x)},\ i\in I\right\} \tag{6.182}$$

в которой функции $f_j(x),\ j\in J$ и $\varphi_i(x),\ -\psi_i(x),\ i\in I$ принадлежат одному и тому же классу (линейные, выпуклые, квадратичные и т.п.), а S – соответствующее выпуклое множество.

В таких задачах с помощью анализа параметрической задачи оптимизации можем рассматривать задачу типа

$$\min_{x\in S}\left\{f_j(x),\ j\in J;\quad Z_i(x,\lambda_i)=\varphi_i(x)-\lambda_i\psi_i(x),\ i\in I\right\}. \tag{6.183}$$

Тогда при фиксированных значениях $\overline{\lambda}_i\ge 0$ функции $f_j(x),\ j\in J$ и $Z_i(x,\overline{\lambda}_i),\ i\in I$ принадлежат одному и тому же классу. Если для решения задачи многокритериальной оптимизации (6.183) применить метод линейной аддитивной свертки, то получим однокритериальную задачу

$$\min_{x\in S}\left[\sum_{j\in J}v_jf_j(x)+\sum_{i\in I}u_iZ_i(x,\overline{\lambda}_i\right],$$

функции которой соответствуют одному и тому же классу задач оптимизации.

Таким образом, вместо задачи (6.182) можно рассматривать параметрическую задачу (6.183) и при некоторой процедуре фиксирования значений параметров $\lambda_i,\ i\in I$ провести анализ многокритериальной задачи (6.183) и тем самым найти эффективные решения задачи (6.182).

Другой класс многокритериальных задач оптимизации с дробными функциями может быть рассмотрен, когда используются различные операции над входящими в них функциями. Рассмотрим некоторые из возможных формулировок задач многокритериальной оптимизации с дробными обобщенными функциями:

$$\min_{x\in S}\left\{\sum_{r=1}^{p_j}f_j^r(x),\ j\in J;\ \sum_{k=1}^{q_i}\frac{\varphi_i^k(x)}{\psi_i^k(x)},\ i\in I\right\},$$

$$\min_{x\in S}\left\{\prod_{r=1}^{p_j}f_j^r(x),\ j\in J;\ \prod_{k=1}^{q_i}\frac{\varphi_i^k(x)}{\psi_i^k(x)},\ i\in I\right\},$$

$$\min_{x\in S}\left\{\max_{r=\overline{1,p_j}}f_j^r(x),\ j\in J;\ \max_{k=\overline{1,q_i}}\frac{\varphi_i^k(x)}{\psi_i^k(x)},\ i\in I\right\},$$

или же в общем виде задачу

$$\min_{x \in S} \Big\{ \bigwedge_{r=1}^{p_j} f_j^r(x),\ j \in J;\ \bigwedge_{k=1}^{q_i} \frac{\varphi_i^k(x)}{\psi_i^k(x)},\ i \in I \Big\}, \tag{6.184}$$

где $\bigwedge$ означает одну из возможных операций $\sum, \prod$ или max.

Тогда вместо таких задач можно рассматривать параметрические задачи

$$\min_{x \in S} \Big\{ \bigwedge_{r=1}^{p_j} f_j^r(x),\ j \in J;\ \bigwedge_{k=1}^{q_i} Z_i^k(x, \lambda_i^k),\ i \in I \Big\}, \tag{6.185}$$

в которых $Z_i^k(x, \lambda_i^k) = \varphi_i^k(x) - \lambda_i^k \psi_i^k(x), \quad r = \overline{1, q_i},\ i \in I.$

Тогда при фиксированных $\lambda_i^k \geq 0$, функции $Z_i^k(x, \lambda_i^k)$ имеют те же свойства, что и функции $f_j^r(x)$, и тогда, для решения таких задач при $\overline{\lambda}_i^r$ можно провести анализ многокритериальной задачи (6.184) и найти эффективные решения для задачи (6.185).

Одной из постановок такого класса задач многокритериальной оптимизации функций разных классов может быть рассмотрена задача с двумя типами функций: линейными и дробно-линейными, выпуклыми и дробно-выпуклыми, квадратичными и дробно-квадратичными. Для таких функций также могут быть применены операции суммирования, мультипликатирования или взятия максимума конечного числа функций. В таких задачах можно использовать некоторые процедуры преобразования функций из одного класса в другой.

В задачах многокритериальной оптимизации с разнотипными функциями рассматриваются различные задачи математического программирования. Решение таких задач представляет собой сложный процесс, в которых необходимо учитывать различные аспекты входящих в них функций, среди которых относится их линейность или нелинейность, выпуклость или невыпуклость, дифференцируемость или недифференцируемость, а также и другие свойства функций, влияющие на выбор метода решения соответствующих задач оптимизации.

При решении таких задач также следует учесть возможность перехода от более простых классов задач к более сложным классам задач, охватывающие все остальные простые классы задач оптимизации. В качестве примера, можно рассмотреть последовательность, определяющая сложность задач оптимизации:

- линейные $\Longrightarrow$ нелинейные задачи

линейные транспортные задачи $\Longrightarrow$ задачи линейного программирования $\Longrightarrow$ задачи выпуклого программирования $\Longrightarrow$ задачи нелинейного программирования

- дробно-линейные $\Longrightarrow$ нелинейные задачи

дробно-линейные транспортные задачи $\Longrightarrow$ задачи дробно-линейного программирования $\Longrightarrow$ задачи дробно-выпуклого программирования $\Longrightarrow$ задачи нелинейного программирования

- дифференцируемые $\Longrightarrow$ недифференцируемые задачи

дифференцируемые функции $\Longrightarrow$ почти-дифференцируемые функции $\Longrightarrow$ недифференцируемые функции

В завершение данного параграфа отметим, что приведенные выше задачи многокритериальной дробной оптимизации требуют дальнейшего подробного исследования и разработки методов их решения.

6.7. Блочная задача сепарабельного дробно-выпуклого программирования

Рассмотрим задачу дробно-выпуклого программирования, в которой система ограничений имеет блочно-диагональную структуру, а функционал является сепарабельной функцией

$$F_1(x) = \sum_{j=1}^{p} \frac{\varphi_j(x_j)}{\psi_j(x_j)} \to \min, \tag{6.186}$$

$$\sum_{j=1}^{p} g_j^i(x_j) \le a_i \quad (i = \overline{1,m}), \tag{6.187}$$

$$h_j^l(x_j) \le b_j^l \quad (l = \overline{1,m_j};\ j = \overline{1,p}). \tag{6.188}$$

Переменные данной задачи разбиты на p групп

$$x_j = (x_j^1, x_j^2, \ldots, x_j^{n_j}) \in R^{n_j} \quad (j = \overline{1,p}),$$

которые образуют вектор $x = (x_1, x_2, \ldots, x_p) \in R^n$, где $n = \sum_{j=1}^{p} n_j$. На множестве каждой группы переменных x_j определены непрерывные функции

$$\varphi_j,\ \psi_j,\ g_j^i\ \ (i = \overline{1,m}), \quad h_j^l\ \ (l = \overline{1,m_j}): \quad R^{n_j} \to R.$$

Предположим, что функции φ_j, $-\psi_j$, g_j^i и h_j^l выпуклы по группе переменных x_j $(j = \overline{1,p})$, а a_i $(i = \overline{1,m})$, b_j^l $(l = \overline{1,m_j};\ j = \overline{1,p})$ – заданные числа. Обозначим через S множество значений переменных x, удовлетворяющих ограничениям (6.187) и (6.188).

В задаче (6.186)–(6.188) можно использовать и другие сепарабельные обобщенные дробные функционалы. Тогда получим задачи обобщенного сепарабельного дробного программирования:

$$\min_{x\in S}\left[F_2(x) = \max_{j=\overline{1,p}} \frac{\varphi_j(x_j)}{\psi_j(x_j)}\right], \tag{6.189}$$

или же

$$\min_{x\in S}\left[F_3(x) = \prod_{j=1}^{p} \frac{\varphi_j(x_j)}{\psi_j(x_j)}\right]. \tag{6.190}$$

Приведенную ниже схему решения задачи (6.186)–(6.188) можно с таким же успехом применить и для решения задачи (6.189) или задачи (6.190).

Предположим, что функции $\varphi_j(x_j)$ и $\psi_j(x_j)$ такие, что функционалы $F_1(x), F_2(x)$ и $F_3(x)$ являются квазивыпуклыми, без уточнения необходимых условий, которым они должны удовлетворять.

В общем случае задачи минимизации обобщенных дробных функционалов $F_1(x), F_2(x)$ и $F_3(x)$ при ограничениях (6.187) и (6.188) являются многоэкстремальными и для их решения необходимо использовать общие методы математического программирования. Такие методы приведены в предыдущих параграфах данной главы. Однако имеются случаи, когда они являются одноэкстремальными и возможно нахождение глобального минимума. В таких случаях для решения данных задач можно применить некоторые модификации общих методов дробно-выпуклого программирования.

Рассмотрим каждый возможный вариант постановки данных задач в отдельности.

Пусть $p = 1, \quad n_1 \geq 1, \quad m \geq 1$ и $m_1 = 0$. Тогда для всех трех задач получим задачу дробно-выпуклого программирования

$$\frac{\varphi(x)}{\psi(x)} \to \min,$$

$$g_i(x) \leq a_i \quad (i = \overline{1, m}),$$

которая имеет единственный минимум. Это позволят использовать общие методы решения задач дробно-выпуклого программирования (см. п. 5.1.).

Пусть теперь $p \geq 2, \quad n_j \geq 2, \quad m = 0$ и $m_j \geq 1$. Тогда получим задачи минимизации сепарабельных функционалов $F_1(x), F_2(x)$ и $F_3(x)$ на множестве блочно-диагональных ограничений без связующих ограничений, а именно: найти минимум функционала $F_1(x)$ ($F_2(x)$ или $F_3(x)$) при ограничениях (6.188).

Поскольку данные функционалы являются сепарабельными, то они сводятся к решению, отдельно для каждого j, задач дробно-выпуклого программирования следующего вида:

$$\frac{\varphi_j(x_j)}{\psi_j(x_j)} \to \min, \tag{6.191}$$

$$h_j^l(x_j) \leq b_j^l \quad (l = \overline{1, m_j}). \tag{6.192}$$

Если задачу (6.191), (6.192) решать параметрическим методом, то при фиксированных $\lambda_j = \overline{\lambda}_j$ получим следующие задачи выпуклого программирования:

$$\varphi_j(x_j) - \overline{\lambda}_j \psi_j(x_j) \to \min,$$

$$h_j^l(x_j) \leq b_j^l \quad (l = \overline{1, m_j}).$$

Рассмотрим общий случай задачи (6.186)–(6.188), т.е. когда $p \geq 2$, $n_j \geq 2$, $m \geq 1$ и $m_j \geq 1$. Тогда задача (6.186)–(6.188) будет многоэкстремальной, и мы остановимся на нахождении некоторого локального минимума, а в отдельных случаях, при выполнении соответствующих условий, – глобального.

Приведем метод решения задачи (6.186)–(6.188) с использованием схемы декомпозиции по ограничениям и субградиентных методов.

Для задачи (6.186)–(6.188) построим функцию Лагранжа

$$L_1(x,u)=\sum_{j=1}^{p}\left[\frac{\varphi_j(x_j)}{\psi_j(x_j)}+\sum_{i=1}^{m}u_i g_j^i(x_j)\right]-\sum_{i=1}^{m}u_i a_i, \tag{6.193}$$

где $u=(u_1,u_2,\dots,u_m)$ – множители лагранжа для ограничений (6.187), и рассмотрим задачу

$$\max_{u\geq 0} L_1^*(u), \tag{6.194}$$

в которой

$$L_1^*(u)=\min_{x\in S_1\times\cdots\times S_p} L_1(x,u), \tag{6.195}$$

а S_j – множество значений переменных x_j, удовлетворяющих ограничениям (6.188) для каждого j в отдельности.

Для решения задачи (6.194) используем субградиентный метод, на $(t+1)$-м шаге которого необходимо выполнить следующие три основных этапа.

1. Решить задачу (6.195) при фиксированных $u=u^t$. Так как функция Лагранжа $L_1(x,u)$ является сепарабельной на множестве переменных x, то задача (6.195) решается для каждого блока в отдельности. Таким образом, имеем p задач вида

$$\frac{\varphi_j(x_j)}{\psi_j(x_j)}+\sum_{i=1}^{m}u_i^t g_j^i(x_j)\to\min, \tag{6.196}$$

$$h_j^l(x_j)\leq b_j^l \quad (l=\overline{1,m_j}). \tag{6.197}$$

Задача (6.196), (6.197) является обобщением задач выпуклого и дробно-выпуклого программирования, для решения которой можно использовать общие методы глобальной оптимизации.

Для решения задачи (6.196), (6.197) также можно использовать параметрический метод. Тогда при фиксированном параметре $\lambda=\lambda_j^t=$ $=\dfrac{\varphi_j(x_j^*(u^{t-1}))}{\psi_j(x_j^*(u^{t-1}))}$ необходимо решить задачу минимизации выпуклого функционала

$$\varphi_j(x_j)-\lambda_j^t\psi_j(x_j)+\sum_{i=1}^{m}u_i^t g_j^i(x_j)$$

при ограничениях (6.197).

2. Определить значения обобщенного градиента функции $L_1^*(u)$ в точке u^t по формуле

$$G_i(u^t) = \sum_{j=1}^{p} g_j^i(x_j^*(u^t)) - a_i \quad (i = \overline{1,m}),$$

где $x^*(u^t)$ – решение задачи (6.195).

3. Найти новые значения

$$u_i^{t+1} = \max\left\{0, u_i^t + \gamma_{t+1} G_i(u^t)\right\} \quad (i = \overline{1,m}),$$

где γ_{t+1} – величина шага.

Для задачи минимизации функционала (6.189) или (6.190) при ограничениях (6.187), (6.188) получим следующие вариации предложенного метода.

Функции Лагранжа имеют вид

$$L_2(x,u) = \max_{j=\overline{1,p}} \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \sum_{j=1}^{p}\sum_{i=1}^{m} u_i g_j^i(x_j) - \sum_{i=1}^{m} u_i a_i,$$

и

$$L_3(x,u) = \prod_{j=1}^{p} \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \sum_{j=1}^{p}\sum_{i=1}^{m} u_i g_j^i(x_j) - \sum_{i=1}^{m} u_i a_i.$$

Исходя из приведенной выше общей схемы декомпозиции по ограничениям с применением субградиентных методов, для этих функций Лагранжа при фиксированных значениях $u = u^t$ необходимо решить задачу (6.195). Так как функционалы в данном случае не являются сепарабельными, то для нахождения обобщенного градиента функции $L_2^*(u) = \max\limits_{u \geq 0} L_2(x,u)$ или $L_3^*(u) = \max\limits_{u \geq 0} L_3(x,u)$ применяется приближенный алгоритм. А именно, решение задачи (6.195) для функционалов $L_2(x,u^t)$ и $L_3(x,u^t)$ сводится к решению задач типа (6.196), (6.197) для каждого блока в отдельности.

Пусть задана задача (6.186)–(6.188) и для ее решения использована схема декомпозиции по ограничениям. Из теоремы Куна-Таккера известно, что для того, чтобы x^* было решением задачи (6.186)–(6.188), необходимо существование таких множителей $u^* = \{u_i^*\}$ $(i = \overline{1,m})$ для ограничений (6.187) и $v^* = \{v_j^{l*}\}$ $(l = \overline{1,m_j};\ j = \overline{1,p})$ для ограничений (6.188), для которых выполняется условие:

$$L(x^*,u,v) \leq L(x^*,u^*,v^*) \leq L(x,u^*,v^*), \tag{6.198}$$

где

$$L(x,u,v)=\sum_{j=1}^{p}\frac{\varphi_j(x_j)}{\psi_j(x_j)}+\sum_{j=1}^{p}\sum_{i=1}^{m}u_i(g_j^i(x_j)-a_i)+\sum_{j=1}^{p}\sum_{l=1}^{m_j}v_j^l(h_j^l(x_j)-b_j^l),$$

притом

$$u_i^*\Big(\sum_{j=1}^{p}g_j^i(x_j^*)-a_i\Big)=0\quad(i=\overline{1,m})\tag{6.199}$$

$$v_j^{l*}\left(h_j^l(x_j^*)-b_j^l\right)=0\quad(l=\overline{1,m_j};\ j=\overline{1,p}).\tag{6.200}$$

Покажем, что полученные решения по общей схеме декомпозиции по ограничениям удовлетворяют оптимальным условиям для задач (6.186)–(6.188).

Пусть u^* – оптимальное решение задачи (6.194), найденное методом обобщенного градиента. Тогда для оптимального u^* выполняются условия

$$G_i(u^*)=0\quad(i=\overline{1,m}),$$

т.е.

$$\sum_{j=1}^{p}g_j^i(x_j^*(u^*))-a_i=0\quad(i=\overline{1,m}),$$

где $x_j^*(u^*)$ – оптимальное решение задачи (6.195) при $u=u^*$. Таким образом, условия (6.199) выполняются.

Так как x_j^* является оптимальным решением задачи (6.195), то для него существуют множители v_j^{l*}, для которых выполняются условия (6.200).

Докажем выполнение неравенств (6.198). Пусть x^*, u^* и v^* удовлетворяют вышеупомянутым условиям. Тогда имеем:

$$\frac{\varphi_j(x_j^*)}{\psi_j(x_j^*)}+\sum_{i=1}^{m}u_i^*(g_j^i(x_j^*)-a_i)\le\frac{\varphi_j(x_j)}{\psi_j(x_j)}+\sum_{i=1}^{m}u_i^*(g_j^i(x_j)-a_i)\quad(j=\overline{1,p}).$$

Или же, просуммировав по всем $j=\overline{1,p}$, получаем

$$\sum_{j=1}^{p}\frac{\varphi_j(x_j^*)}{\psi_j(x_j^*)}+\sum_{j=1}^{p}\sum_{i=1}^{m}u_i^*(g_j^i(x_j^*)-a_i)\le\sum_{j=1}^{p}\frac{\varphi_j(x_j)}{\psi_j(x_j)}+\sum_{j=1}^{p}\sum_{i=1}^{m}u_i^*(g_j^i(x_j)-a_i).$$

Тогда с учетом равенства (6.200) получаем, что

$$L(x^*,u^*,v^*)\le L(x,u^*,v^*).$$

С другой стороны, так как u^* является оптимальным решением имеем, что
$$L_1^*(u) \leq L_1^*(u^*)$$
или же неравенство
$$\min_x L_1(x,u) \leq \min_x L_1(x,u^*),$$
которое также выполняется и в оптимальной точке x^* задачи (6.195) при $u = u^*$, т.е. имеем:
$$L_1(x^*,u) \leq L_1(x^*,u^*). \tag{6.201}$$

Так как для задачи (6.196), (6.197) выполняются условия
$$v_j^l(h_j^l(x_j^*) - b_j^l) \leq v_j^{l*}(h_j^l(x_j^*) - b_j^l),$$
то получим
$$\sum_{j=1}^{p}\sum_{l=1}^{m_j} v_j^l(h_j^l(x_j^*) - b_j^l) \leq \sum_{j=1}^{p}\sum_{l=1}^{m_j} v_j^{l*}(h_j^l(x_j^*) - b_j^l)$$
и тогда с учетом неравенства (6.201) имеем
$$L(x^*,u,v) \leq L(x^*,u^*,v^*),$$
откуда следует, что (x^*,u^*,v^*) является седловой точкой функции Лагранжа $L(x,u,v)$, т.е. является оптимальным решением задачи (6.186)–(6.188).

Аналогичным образом можно доказать, что полученные решения задач (6.189), (6.187), (6.188) и (6.190), (6.187), (6.188) по схеме декомпозиции по ограничениям также являются оптимальными.

В задачи сепарабельного дробно-выпуклого блочного программирования можно также использовать функционал
$$F_4(x) = \frac{\max\limits_{j=\overline{1,p}} \varphi_j(x)}{\min\limits_{j=\overline{1,p}} \psi_j(x)},$$
или же рассмотреть многокритериальный дробный функционал вида
$$F_5(x) = \left(\frac{\varphi_1(x_1)}{\psi_1(x_1)}, \frac{\varphi_2(x_2)}{\psi_2(x_2)}, \ldots, \frac{\varphi_p(x_p)}{\psi_p(x_p)}\right).$$

Если для функции $F_5(x)$ построить функцию Лагранжа на множестве связующих ограничений, для которой частные критерии $L_j^5(x_j,u)$ имеют вид

$$L_j^5(x_j,u)=\frac{\varphi_j(x_j)}{\psi_j(x_j)}+A(x,u)\quad (j=\overline{1,p}),$$

где

$$A(x,u)=\sum_{j=1}^{p}\sum_{i=1}^{m}u_i g_j^i(x_j)-\sum_{i=1}^{m}u_i a_i\quad (j=\overline{1,p}),$$

то задача многокритериальной оптимизации $\min\limits_{x\in S}F_5(x)$ сводится к решению задачи нахождения седловой точки для многокритериальной функции Лагранжа $L_j^5(x_j,u)$, т.е. к решению задачи

$$\max_{u\geq 0}\min_{x\in S}\Big[L_5(x,u)=\Big(L_1^5(x_1,u),L_2^5(x_2,u),\ldots,L_p^5(x_p,u)\Big)\Big],$$

которая разбивается на две задачи, соответствующие задачам (6.194) и (6.195). В данном случае функции $L_5^*(u)$ и $L_5(x,u)$ являются векторными функциями и тогда задачи (6.194) и (6.195) являются задачами векторной оптимизации.

Векторная функция $L_5^*(u)$ является вогнутой, кусочно-линейной, недифференцируемой по каждой компоненте и имеет вид

$$L_5^*(u)=\Big(L_1^5(u),L_2^5(u),\ldots,L_p^5(u)\Big),$$

в которой $L_j^5(u)=\sum\limits_{i=1}^{m}q_i(x^*)u_i+f_j(x_j^*)\quad (j=\overline{1,p})$, а $f_j(x_j^*)=$ $=\dfrac{\varphi_j(x_j^*)}{\psi_j(x_j^*)}\quad (j=\overline{1,p}),\quad q_i(x^*)=\sum\limits_{j=1}^{p}g_j^i(x_j^*)-a_i\quad (i=\overline{1,m})$.

Для решения задачи векторной оптимизации $\max\limits_{u\geq 0}L_5^*(u)$ используем метод сверток с применением операции поточечного минимума частных критерий, а именно определяем функцию $L(u)=\min\limits_{1\leq j\leq p}L_5^*(u)$. Тогда имеем задачу $\max\limits_{u\geq 0}\big[L(u)=\min\limits_{1\leq j\leq p}L_j^5(u)\big]$, которая является задачей дискретного максимума с вогнутым и недифференцируемым функционалом $L(u)$.

Для решения такой задачи в п. 6.1. применен субградиентный метод. Пусть имеем некоторую точку u^t, полученной на предыдущем шаге субградиентного метода. Тогда выполняем следующие три основных этапа:

1) находим значение функции

$$L(u^t) = \min_{1\le j\le p} L_j^5(u^t) = \min_{1\le j\le p}\left[\sum_{i=1}^{m} q_i(x^*(u^t))u_i^t + f_j(x_j^*(u^t))\right] =$$

$$= f_{j_t}(x_{j_t}^*(u^t)) + \sum_{i=1}^{m} q_i(x^*(u^t))u_i^t,$$

где $x^*(u^t)$ является эффективным решением векторной задачи $\min\limits_{x\in S_1\times S_2\times\cdots\times S_p} L_5(x,u^t)$, а j_t – индекс функции $f_j(x_j^*(u^t))$, которая имеет минимальное значение в точке $x_j^*(u^t)$;

2) находим значения обобщенного градиента функции $L(u)$ в точке u^t по формуле

$$g_{_L}^i(u^t) = q_i(x^*(u^t)) = \sum_{j=1}^{p} g_j^i(x_j(u^t)) - a_i \quad (i=\overline{1,m});$$

3) находим новую точку u^{t+1} по формуле

$$u_i^{t+1} = \max\{0,\, u_i^t + \gamma_{t+1} g_{_L}^i(u^t)\}.$$

В рассмотренном субградиентном методе на каждом шаге при фиксированном $u = u^t$ решается векторная задача минимизации функции $L_5(x,u^t)$, в которой частные критерии имеют вид

$$L_j^5(x_j,u^t) = \frac{\varphi_j(x_j)}{\psi_j(x_j)} + A(x,u^t) \quad (j=\overline{1,p}),$$

где

$$A(x,u^t) = \sum_{j=1}^{p}\sum_{i=1}^{m} u_i^t g_j^t(x_j) - \sum_{i=1}^{m} u_i^t a_i \quad (j=\overline{1,p}).$$

Для данной векторной задачи используем метод линейной свертки и тогда для ее решения рассмотрим задачу

$$\min_{x\in S_1\times S_2\times\cdots\times S_p} L(x,u^t),$$

в которой

$$L(x,u^t) = \sum_{j=1}^{p} L_j^5(x_j,u^t) = \sum_{j=1}^{p}\left(\frac{\varphi_j(x_j)}{\psi_j(x_j)} + p\sum_{i=1}^{m} u_i^t g_j^t(x_j)\right).$$

Тогда данная задача распадается на p отдельных задач для каждого блока в отдельности, т.е. получим задачу

$$\min_{x_j \in S_j} \left(\frac{\varphi_j(x_j)}{\psi_j(x_j)} + p \sum_{i=1}^{m} u_i^t g_j^t(x_j) \right)$$

для решения которой применим параметрический метод.

Таким образом, при решении сепарабельной многокритериальной задачи дробной оптимизации получаем аналогичные подзадачи в субградиентном методе, как и для задачи (6.186)–(6.188), или задач с обобщенными дробными функционалами (6.189) и (6.190).

6.8. Блочная задача дробно-выпуклого программирования со связующими переменными и ограничениями

Пусть имеем p групп переменных $x_j = (x_j^1, x_j^2, \ldots, x_j^{n_j}) \in R^{n_j}$, образующих вектор $x = (x_1, x_2, \ldots, x_p) \in R^n$, где $n = \sum_{j=1}^{p} n_j$. На множестве каждой группы переменных x_j определим следующие непрерывные функции: $\varphi_j,\ \psi_j,\ g_j^i\ \ (i = \overline{1,m}),\ h_j^l\ \ (l = \overline{1,m_j}) :\ R^{n_j} \to R$.

Пусть также имеются переменные $y = (y_1, y_2, \ldots, y_k) \in R^k$, на которых определены функции $\alpha, \beta, F_0^i\ \ (i = \overline{1,m}),\ F_j^l\ \ (l = \overline{1,m_j};\ j = \overline{1,p}) :$ $R^k \to R$. Предположим, что функции $\varphi_j,\ -\psi_j,\ \ g_j^i\ \ (i = \overline{1,m}),\ h_j^l\ \ (l = \overline{1,m_j})$ выпуклы на множестве значений переменных x_j, а функции $\alpha,\ -\beta,\ F_0^i\ \ (i = \overline{1,m}),\ F_j^l\ \ (l = \overline{1,m_j};\ j = \overline{1,p})$ выпуклы на множестве значений переменных y.

Рассмотрим задачу дробно-выпуклого программирования на множестве переменных (x, y), в которой система ограничений имеет блочно-диагональную структуру со связующими переменными и ограничениями:

$$F(x,y) = \frac{C(x,y)}{D(x,y)} = \frac{\sum_{j=1}^{p} \varphi_j(x_j) + \alpha(y)}{\sum_{j=1}^{p} \psi_j(x_j) + \beta(y)} \to \min, \tag{6.202}$$

$$\sum_{j=1}^{p} g_j^i(x_j) + F_0^i(y) \leq a_i \quad (i = \overline{1,m}), \tag{6.203}$$

$$h_j^l(x_j) + F_j^i(y) \leq b_j^l \quad (l = \overline{1,m_j}; \quad j = \overline{1,p}), \tag{6.204}$$

в которой $a_i \;\; (i = \overline{1,m})$ и $b_j^l \;\; (l = \overline{1,m_j};\; j = \overline{1,p})$ – заданные числа.

Через $M(x,y)$ обозначим множество значений переменных (x,y), удовлетворяющих ограничениям (6.203), (6.204), и предположим, что оно ограничено, и представляет собой некоторый компакт.

Предположим также, что функции $\varphi_j(x_j)$ и $\psi_j(x_j)$ неотрицательны, а $\alpha(y)$ и $\beta(y)$ положительны на компакте $M(x,y)$.

Для решения задачи (6.202)–(6.204) можно построить итеративный алгоритм с использованием субградиентных методов, применяя схему декомпозиции по переменным y, а при решении задачи на множестве переменных x – схему декомпозиции по ограничениям. В таком случае алгоритм решения задачи (6.202)–(6.204) будет иметь два уровня, на каждом из которых используются субградиентные методы.

Рассмотрим алгоритм более подробно.

Первый (внешний) *уровень* . Применяем схему декомпозиции по переменным. Тогда для фиксированных переменных $y = \overline{y}$ получим задачу:

$$\frac{C(x,\overline{y})}{D(x,\overline{y})} = \frac{\sum\limits_{j=1}^{p} \varphi_j(x_j) + \alpha(\overline{y})}{\sum\limits_{j=1}^{p} \psi_j(x_j) + \beta(\overline{y})} \to \min, \tag{6.205}$$

$$\sum_{j=1}^{p} g_j^i(x_j) \leq \overline{a}_i(\overline{y}) \quad (i = \overline{1,m}), \tag{6.206}$$

$$h_j^l(x_j) \leq b_j^l(\overline{y}) \quad (l = \overline{1,m_j};\; j = \overline{1,p}), \tag{6.207}$$

где $\overline{a}_i(\overline{y}) = a_i - F_0^i(\overline{y}), \;\; \overline{b}_j^l(\overline{y}) = b_j^l - F_j^l(\overline{y}) \;\; (l = \overline{1,m_j};\; j = \overline{1,p})$ – постоянные коэффициенты.

На множестве значений переменных x определим функцию

$$\Phi(\overline{y}) = \min_{x \in R(\overline{y})} \frac{C(x,\overline{y})}{D(x,\overline{y})}.$$

Здесь $R(\overline{y})$ – множество значений переменных x, удовлетворяющих ограничениям (6.206), (6.207) при фиксированных значениях переменной $y = \overline{y}$.

Тогда на внешнем уровне итеративного алгоритма решения задачи (6.202)–(6.204) необходимо минимизировать строго квазивыпуклую функцию $\Phi(y)$ при ограничениях $y \geq 0$. Для этого используем метод обобщенного градиентного спуска, на t-м шаге которого необходимо выполнить следующие три основных этапа.

1. Решить задачу (6.205)–(6.207) при фиксированных $y = \overline{y}^t$, т.е. найти оптимальное решение $x(\overline{y}^t)$ и двойственные оценки $\{u_i(\overline{y}^t)\}$ и $\{v_j^l(\overline{y}^t)\}$ соответственно для ограничений (6.206) и (6.207).
2. Определить значения обобщенного градиента функции $\Phi(y)$ в точке $y = \overline{y}^t$ по формуле

$$G(\overline{y}^t) = -\sum_{i=1}^{m} u_i(\overline{y}^t) G_0^i(\overline{y}^t) - \sum_{j=1}^{p}\sum_{i=1}^{m} v_j^l(\overline{y}^t) G_j^l(\overline{y}^t) +$$

$$+\frac{G_\alpha D(x(\overline{y}^t), \overline{y}^t) - G_\beta C(x(\overline{y}^t), \overline{y}^t)}{[D(x(\overline{y}^t), \overline{y}^t)]^2},$$

   где G_α и G_β – обобщенные градиенты функций $\alpha(y)$ и $\beta(y)$, а $G_0^i(\overline{y}^t)$ и $G_j^l(\overline{y}^t)$ – обобщенные градиенты функций $F_0^i(y)$ и $F_j^l(y)$ в точке $y = \overline{y}^t$.
3. Найти новые значения

$$\overline{y}^{t+1} = \overline{y}^t - \gamma_{t+1} G(\overline{y}^t),$$

   где γ_{t+1} – величина шага.

В первом пункте внешнего уровня итеративного алгоритма необходимо решить задачу (6.205)–(6.207). Так как эта задача имеет блочно-диагональную структуру, то, используя схему декомпозиции по ограничениям, сводим ее к решению выпуклых задач для каждого блока в отдельности.

Решение задачи (6.205)–(6.207) составляет *второй* (внутренний) *уровень* итеративного алгоритма. Для этого рассмотрим следующие две задачи:

$$\max_{u \geq 0} L^*(u)$$

и

$$L^*(u) = \min_{x \in S(x)} L(x, u),$$

где $u = \{u_i\}$ – множители Лагранжа функции $L(x, u)$, определенной по формуле

$$L(x,u) = \frac{\sum_{j=1}^{p}\left[\varphi_j(x_j) + \sum_{i=1}^{m} u_i g_j^i(x_j)\right] + \alpha(\overline{y}) - \sum_{i=1}^{m} u_i \bar{a}_i(\overline{y})}{\sum_{j=1}^{p}\psi_j(x_j) + \beta(\overline{y})},$$

а $S(x)$ – множество значений переменных x, удовлетворяющих ограничениям (6.207).

Тогда, на $(s+1)$-м шаге субградиентного метода максимизации функции $L^*(u)$ необходимо выполнить следующие три основных этапа.

1. Решить следующую задачу дробно-выпуклого программирования параметрическим методом при фиксированных $u = u^s$

$$\frac{\sum_{j=1}^{p}\left[\varphi_j(x_j) + \sum_{i=1}^{m} u_i^s g_j^i(x_j)\right] + \alpha_0(u^s, \overline{y})}{\sum_{j=1}^{p}\psi_j(x_j) + \beta(\overline{y})} \to \min, \tag{6.208}$$

$$h_j^l(x_j) \le b_j^l(\overline{y}) \quad (l = \overline{1, m_j};\ j = \overline{1, p}), \tag{6.209}$$

где $\alpha_0(u^s, \overline{y}) = \alpha(\overline{y}) - \sum_{i=1}^{m} u_i^s \overline{a}_i(\overline{y})$.

При фиксированном параметре

$$\lambda = \lambda^s = L(x^*(u^{s-1}), u^s)$$

решить задачу выпуклого программирования

$$\varphi_j(x_j) - \lambda^s \psi_j(x_j) + \sum_{i=1}^{m} u_i^s g_j^i(x_j) \to \min,$$

$$h_j^l(x_j) \le b_j^l(\overline{y}) \quad (l = \overline{1, m_j}).$$

Пусть $x_j^*(u^s)$ – решение задачи, а $v_j^{l*}(u^s)$ – соответствующие значения множителей Лагранжа для данного решения. Тогда $x^*(u^s) = \{x_j^*(u^s)\}$ является решением задачи (6.208), (6.209).

2. Определить значения обобщенного градиента функции $L^*(u)$ в точке $u = u^s$ по формуле

$$G(u^s) = \Big\{ \Big[\sum_{j=1}^{p} g_j^i(x_j^*(u^s)) - \overline{\alpha}_i(\overline{y}) \Big] / D(x^*(u^s), \overline{y}) \Big\}.$$

3. Найти новые значения

$$u^{s+1} = \max(0, u^s + \gamma_{s+1} G(u^s)),$$

где γ_{s+1} – величина шага.

Следует заметить, что поскольку внутренняя задача (6.205)–(6.207) решается приближенно, то значение обобщенного градиента функции $\Phi(\overline{y})$ в точке $\overline{y} = \overline{y}^t$ определяется с некоторой погрешностью. При этом последовательность внутренних задач необходимо решить с некоторой заданной точностью.

Таким образом, в результате решения задачи (6.202)–(6.204) по указанному алгоритму получим оптимальные значения переменных (x, y) и двойственных оценок u и $\{v_j\}$.

Заметим, что если в задаче (6.202)–(6.204) имеются некоторые отклонения от общей постановки, то для ее решения можно использовать другие схемы итеративного алгоритма. Рассмотрим некоторые вариации структуры системы ограничений.

Пусть $m = 0$, т.е. связующие ограничения (6.203) отсутствуют. Тогда, если использовать схему декомпозиции по переменным, то задачу (6.205)–(6.207) можно решить параметрическим методом для каждого блока в отдельности. В конечном итоге получим одноуровневый алгоритм. Аналогично имеет место, когда $m > 0$, но все функции $g_j^i(x_j)$ тождественно равны нулю на множестве переменных x (т.е. отсутствуют связующие ограничения по x).

Если же $m_j = 0 \quad (j = \overline{1,p})$, т.е. блочно-диагональные ограничения (6.204) отсутствуют, то получим задачу, для которой достаточно применить схему декомпозиции по переменным. В этом случае эффективность данного метода достигается за счет специальной структуры ограничений (6.203) при фиксированных значениях y (задача транспортного типа, блочная структура ограничений и т.п.).

Рассмотрим теперь некоторые варианты задачи (6.202)–(6.204) в зависимости от структуры функционала (6.202). В практических задачах возможен случай, когда функционал (6.202) будет иметь вид $C(x, y)/D(y)$. Тогда задача (6.205)–(6.207) будет задачей выпуклого программирования, и по схеме декомпозиции по ограничениям она решается для каждого блока в отдельности.

6.9. Блочные задачи обобщенного дробно-выпуклого программирования со связующими переменными и ограничениями

Рассмотрим обобщенные задачи дробно-выпуклого программирования (в виде сумм выпуклого и дробно-выпуклого или двух дробно-выпуклых функционалов) блочно-диагональной структуры со связующими переменными и связующими ограничениями. Для решения задач со связующими переменными приводится общая схема декомпозиционных алгоритмов с применением схем разложения по переменным и субградиентных методов. Для каждого конкретного вида функционала обобщенной задачи дробно-выпуклого программирования приводятся формулы вычисления обобщенных градиентов и вид подзадач, которые необходимо решить для каждого блока.

1. Постановка задачи. Пусть имеем p групп переменных $x_j = (x_j^1, x_j^2, \ldots, x_j^{n_j}) \in R^{n_j}$, образующих вектор $x = (x_1, x_2, \ldots, x_p) \in R^n$, где $n = \sum\limits_{j=1}^{p} n_j$. На множестве каждой группы переменных x_j определим следующие непрерывные функции: φ_j, ψ_j, g_j^i $(i = \overline{1,m})$, h_j^l $(l = \overline{1,m_j}) : R^{n_j} \to R$.

Пусть также имеются переменные $y = (y_1, y_2, \ldots, y_k) \in R^k$, на которых определены функции: α, β, F_0^i $(i = \overline{1,m})$, F_j^l $(l = \overline{1,m_j};\ j = \overline{1,p}) : R^k \to R$. Предположим, что функции φ_j, $-\psi_j$, g_j^i $(i = \overline{1,m_j})$, h_j^l $(l = \overline{1,m_j})$ выпуклы на множестве значений переменных x_j, а функции α, $-\beta$, F_0^i $(i = \overline{1,m})$, F_j^l $(l = \overline{1,m_j};\ j = \overline{1,p})$ выпуклы на множестве значений переменных y.

Рассмотрим следующие два множества допустимых значений переменных x и y :

$$Q(x,y) = \left\{x, y :\ h_j^l(x_j) + F_j^l(y) \le b_j^l \quad (l = \overline{1,m_j};\ j = \overline{1,p})\right\},$$

$$S(x,y) = \Bigg\{x, y :\ \sum_{j=1}^{p} g_j^i(x_j) + F_0^i(y) \le a_i \quad (i = \overline{1,m}),$$

$$h_j^l(x_j) + F_j^l(y) \le b_j^l \quad (l = \overline{1,m_j};\ \ j = \overline{1,p})\Bigg\},$$

в которых $a = (a_1, \ldots, a_m) \in R^m$; $b_j = (b_j^1, \ldots, b_j^{m_j}) \in R^{m_j}$ $(j = \overline{1,p})$ – заданные числа. Системы ограничений данных множеств имеют следующую структуру:

1) для множества $Q(x, y)$ они имеют блочно-диагональную структуру со связующими переменными;
2) для множества $S(x, y)$ они имеют блочно-диагональную структуру со связующими ограничениями и связующими переменными.

Для функционалов

$$F_1(x, y) = \frac{\sum_{j=1}^{p} \varphi_j(x_j)}{\sum_{j=1}^{p} \psi_j(x_j)} + \frac{\alpha(y)}{\beta(y)},$$

$$F_2(x, y) = \frac{\sum_{j=1}^{p} \varphi_j(x_j)}{\sum_{j=1}^{p} \psi_j(x_j)} + \alpha(y),$$

$$F_3(x, y) = \sum_{j=1}^{p} \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \frac{\alpha(y)}{\beta(y)},$$

$$F_4(x, y) = \sum_{j=1}^{p} \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \alpha(y),$$

$$F_5(x, y) = \max_j \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \frac{\alpha(y)}{\beta(y)},$$

$$F_6(x, y) = \max_j \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \alpha(y),$$

$$F_7(x, y) = \frac{\max_j \varphi_j(x_j)}{\min_j \psi_j(x_j)} + \frac{\alpha(y)}{\beta(y)},$$

$$F_8(x, y) = \frac{\max_j \varphi_j(x_j)}{\min_j \psi_j(x_j)} + \alpha(y),$$

$$F_9(x,y)=\sum_{j=1}^{p}\varphi_j(x_j)+\frac{\alpha(y)}{\beta(y)},$$

$$F_{10}(x,y)=\max_j \varphi_j(x_j)+\frac{\alpha(y)}{\beta(y)}$$

рассмотрим следующие задачи:

$$\min_{(x,y)\in Q(x,y)} F_r(x,y)\quad (r=\overline{1,10}) \tag{6.210}$$

или

$$\min_{(x,y)\in S(x,y)} F_r(x,y)\quad (r=\overline{1,10}). \tag{6.211}$$

Каждая из задач (6.210) или (6.211) не является задачей выпуклого или дробно-выпуклого программирования. Они являются их обобщениями, в которых необходимо минимизировать функционалы в виде суммы выпуклых и дробно-выпуклых функций или суммы двух дробно-выпуклых функций, каждая из которых зависит от разных групп переменных.

Это дает возможность построения декомпозиционных алгоритмов с учетом структуры системы ограничений множеств $Q(x,y)$ и $S(x,y)$. Для решения задач (6.210) и (6.211) эффективными являются итеративные алгоритмы с использованием субградиентных методов.

2. Одноуровневые декомпозиционные алгоритмы. Из структуры системы ограничений множества $Q(x,y)$ и функционалов $F_r(x,y)$ $(r=\overline{1,10})$, видно, что если применять схему декомпозиции по переменным y, то полученные задачи для определения значений обобщенных градиентов будут решаться для каждого блока в отдельности.

Построим функцию Лагранжа для задач (6.210) по общей формуле

$$L_r(x,y,v)=F_r(x,y)+\sum_{j=1}^{p}\sum_{l=1}^{m_j} v_j^l\left[h_j^l(x_j)+F_j^l(y)-b_j^l\right]\quad (r=\overline{1,10}),$$

где $v=\{v_j^l\}$ – множители Лагранжа. Рассмотрим декомпозиционные алгоритмы, построенные на базе схемы декомпозиции по переменным с применением субградиентных методов.

Фиксируем значения переменных $y=\overline{y}$. Тогда получим задачи

$$\Phi_r(\overline{y})=\min_{x\in Q(\overline{y})} F_r(x,\overline{y})\quad (r=\overline{1,10}), \tag{6.212}$$

в которых $Q(\overline{y})$ –множество значений переменных x, удовлетворяющих ограничениям множества $Q(x,y)$ при фиксированных $y=\overline{y}$.

Если каждый блок в отдельности обозначить через

$$Q_j(\overline{y}) = \left\{x_j : h_j^l(x_j) \leq b_j^l - F_j^l(\overline{y}) \;\; (l = \overline{1, m_j})\right\} \quad (j = \overline{1, p}),$$

то множество $Q(\overline{y})$ может быть представлено в виде декартова произведения множеств $Q_j(\overline{y})$, т.е.

$$Q(\overline{y}) = Q_1(\overline{y}) \times \cdots \times Q_p(\overline{y}).$$

Также предположим, что $\varphi_j(x_j) \geq 0$ и $\psi_j(x_j) > 0$ для всех $x_j \in$ $\in Q_j(\overline{y}) \quad (j = \overline{1, p})$.

Тогда функции $\Phi_r(\overline{y})$, определенные по формуле (6.212), являются выпуклыми или квазивыпуклыми на множестве переменных y. Поэтому для решения задач минимизации функций $\Phi_r(\overline{y}) \quad (r = \overline{1, 10})$ используются субградиентные методы, на $(t+1)$-м шаге которых необходимо выполнить следующие три основных этапа.

1. Решить задачу (6.212) при фиксированных $\overline{y} = \overline{y}^t$, т.е. найти оптимальные решения $x^*(\overline{y})$ и соответствующие им двойственные оценки $v^*(\overline{y})$.
2. Определить значения обобщенных градиентов по одной из формул:

$$G_r(\overline{y}^t) = \left\{\frac{G_\alpha \beta(\overline{y}^t) - G_\beta \alpha(\overline{y}^t)}{[\beta(\overline{y}^t)]^2} + \sum_{j=1}^{p} \sum_{l=1}^{m_j} v_j^l(\overline{y}^t) G_{F_j^l}(\overline{y}^t)\right\},$$

$$r = 1, 3, 5, 7, 9, 10;$$

$$G_r(\overline{y}^t) = \left\{G_\alpha + \sum_{j=1}^{p} \sum_{l=1}^{m_j} v_j^l(\overline{y}^t) G_{F_j^l}(\overline{y}^t)\right\},$$

$$r = 2, 4, 6, 8;$$

в которых G_α, G_β – соответственно обобщенные градиенты функций $\alpha(y)$ и $\beta(y)$ в точке $y = \overline{y}^t$; а $G_{F_j^l}(\overline{y}^t)$ – обобщенные градиенты функций $F_j^l(y)$ в той же точке $y = \overline{y}^t$.
3. Найти новые значения

$$\overline{y}^{t+1} = \overline{y}^t - \gamma_{t+1} G_r(\overline{y}^t) \quad (r = \overline{1, 10}),$$

где γ_{t+1} – величина шага.

Следует отметить, что на каждом шаге субградиентного метода необходимо решить задачи (6.212). Именно при решении этих задач видна эффективность применения схемы декомпозиции по переменным. Так как переменные y фиксированы, то при $y = \overline{y}$ на каждом шаге субградиентного метода необходимо минимизировать функционалы $F_r(x, \overline{y})$ на множестве переменных x, удовлетворяющих блочно-диагональным ограничениям.

Поскольку функционалы $F_r(x, y)$, $\quad r = 3, 4, 5, 6, 9, 10$, являются сепарабельными на множестве переменных x, то соответствующие задачи (6.212) решаются для каждого блока в отдельности. Таким образом, на каждом шаге субградиентного метода решаются задачи дробно-выпуклого программирования

$$\min_{x_j \in Q_j(\overline{y})} \frac{\varphi_j(x_j)}{\psi_j(x_j)} \tag{6.213}$$

или выпуклого программирования

$$\min_{x_j \in Q_j(\overline{y})} \varphi_j(x_j) \tag{6.214}$$

для которых находятся оптимальные решения $x^*(y)$ и соответствующие им двойственные оценки $v_j^*(\overline{y}) = \left\{v_j^1(\overline{y}), \ldots, v_j^{m_j}(\overline{y})\right\}$.

При решении задачи (6.213) параметрическим методом фиксируем значение параметра

$$\lambda_j^t = \varphi_j(x_j^*(y^{t-1})/\psi_j(x_j^*(y^{t-1}))$$

и решаем задачу

$$\min_{x_j \in Q_j(y^t)} \{\varphi_j(x_j) - \lambda_j^t \psi_j(x_j)\}.$$

Для минимизации функционалов $F_r(x, \overline{y})$, $r = 1, 2, 7, 8$, задачи (6.212), которые в данных случаях не являются сепарабельными, используется параметрический метод решения задач дробно-выпуклого программирования. Тогда на каждом шаге субградиентного метода при фиксированном значении параметра $\lambda = \lambda^t = F_r(x^*(y^{t-1}), y^t)$, $\quad r = 1, 2, 7, 8$ решаются выпуклые задачи:

$$\min_{x_j \in Q_j(\overline{y})} \left\{\varphi_j(x_j) - \lambda^t \psi_j(x_j)\right\}. \tag{6.215}$$

3. Двухуровневые декомпозиционные алгоритмы. Рассмотрим случай, когда система ограничений имеет не только связующие переменные, но и связующие ограничения. Пусть система ограничений образует множество $S(x, y)$ и для задачи (6.211) построим функции Лагранжа не только на множестве блочных ограничений, но и с учетом связующих ограничений по общей формуле

$$
\begin{aligned}
L_r(x,y,u,v) = F_r(x,y) + \sum_{j=1}^{p}\sum_{l=1}^{m_j} v_j^l \left[h_j^l(x_j) + F_j^l(y) - b_j^l\right] + \\
+ \sum_{i=1}^{m} u_i \Big[\sum_{j=1}^{p} g_j^i(x_j) + F_0^i(y) - a_i\Big] \quad (r = \overline{1,10}),
\end{aligned} \tag{6.216}
$$

где $u = \{u_i\}$ и $v = \left\{v_j^*(\overline{y}) = \left\{v_j^1(\overline{y}), \ldots, v_j^{m_j}(\overline{y})\right\}\right\}$ – соответствующие векторы множителей Лагранжа $u \geq 0, v \geq 0$.

Для решения задач (6.211) приведем двухуровневые алгоритмы, в которых на первом уровне для переменных y используется схема декомпозиции по переменным, а на втором уровне для переменных x – схема декомпозиции по ограничениям.

1) *Схема декомпозиции по переменным*

Первый уровень алгоритма не отличается от алгоритма решения задач (6.210), т.е. фиксируются переменные $y = \overline{y}$ и получаются задачи

$$
\Phi_r(\overline{y}) = \min_{x \in S(x,\overline{y})} F_r(x,\overline{y}) \quad (r = \overline{1,10}), \tag{6.217}
$$

в которых $S(x,\overline{y})$ – множество значений переменных x, удовлетворяющих ограничениям множества $S(x,y)$ при $y = \overline{y}$. Тогда для минимизации функций $\Phi_r(\overline{y})$ $(r = \overline{1,10})$ используются субградиентные методы, на $(t+1)$-м шаге которых необходимо выполнить следующие три основных этапа схемы декомпозиции по переменным.

1. Решить задачу (6.217) при фиксированных $y = \overline{y}^t$, т.е. найти оптимальные решения $x^*(\overline{y})$ и соответствующие им двойственные оценки $u^*(\overline{y})$ и $v^*(\overline{y})$.
2. Определить значения обобщенных градиентов по одной из формул:

$$
G_r(\overline{y}^t) = \left\{ \frac{G_\alpha \beta(\overline{y}^t) - G_\beta \alpha(\overline{y}^t)}{[\beta(\overline{y}^t)]^2} + \right.
$$

$$
\left. + \sum_{j=1}^{p}\sum_{l=1}^{m_j} v_j^l(\overline{y}^t) G_{F_j^l}(\overline{y}^t) + \sum_{i=1}^{m} u_i(\overline{y}^t) G_{F_0^i}(\overline{y}^t) \right\},
$$

$$
r = 1, 3, 5, 7, 9, 10;
$$

$$
G_r(\overline{y}^t) = \left\{ G_\alpha + \sum_{j=1}^{p}\sum_{l=1}^{m_j} v_j^l(\overline{y}^t) G_{F_j^l}(\overline{y}^t) + \sum_{i=1}^{m} u_i(\overline{y}^t) G_{F_0^i}(\overline{y}^t) \right\},
$$

$$
r = 2, 4, 6, 8,
$$

в которых $G_\alpha, G_\beta, G_{F_j^l}$ и $G_{F_0^i}$ – значения обобщенных градиентов функции $\alpha(y), \beta(y), F_j^l(y)$ и $F_0^i(y)$ в точке $y = \overline{y}^t$, а $v_j^l(\overline{y}^t)$ и $u_i(\overline{y}^t)$ – множители Лагранжа функции (6.216), соответствующие ограничениям и вычисленные в точке $y = \overline{y}^t$.

3. Найти новые значения

$$\overline{y}^{t+1} = \overline{y}^t - \gamma_{t+1} G_r(\overline{y}^t) \quad (r = \overline{1,10}),$$

где γ_{t+1} – величина шага.

2) *Схема декомпозиции по ограничениям*

На втором уровне алгоритма применяется схема декомпозиции по ограничениям для решения задач (6.217) при фиксированных $y = \overline{y}$. Так как на первом уровне фиксируются переменные $y = \overline{y}$, то система ограничений множества $S(x, \overline{y})$ имеет блочно-диагональную структуру на множестве переменных x со связующими ограничениями.

Для решения задач (6.217) строятся функции Лагранжа по формуле

$$L_r(x, \overline{y}, u) = F_r(x, \overline{y}) + \sum_{i=1}^{m} u_i \Big[\sum_{j=1}^{p} g_j^i(x_j) + F_0^i(\overline{y}) - a_i \Big] \quad (r = \overline{1,10}) \quad (6.218)$$

и рассматриваются задачи

$$\max_{u \geq 0} L_r^*(\overline{y}, u) \quad (r = \overline{1,10}), \quad (6.219)$$

в которых

$$L_r^*(\overline{y}, u) = \min_{x \in Q(\overline{y})} L_r^*(x, \overline{y}, u) \quad (r = \overline{1,10}). \quad (6.220)$$

Функции $L_r^*(\overline{y}, u)$, определенные по формуле (6.220), являются кусочно-линейными и вогнутыми функциями, и для решения задач (6.219) используются субградиентные методы, на s-м шаге которых необходимо выполнить следующие три основных этапа схемы декомпозиции по ограничениям.

1. Решить задачу (6.220) при $u(\overline{y}) = u^s(\overline{y})$ и найти оптимальные решения $x^*(\overline{y}, u^s)$ и соответствующие значения множителей Лагранжа $v^*(\overline{y})$.
2. Определить значения обобщенных градиентов функций $L_r^*(\overline{y}, u)$ в точке $u(\overline{y}) = u^s(\overline{y})$ по формуле

$$G_r^i(\overline{y}, u^s(\overline{y})) = \sum_{j=1}^{p} g_j^i(x_j^*(\overline{y}, u^s)) + F_0^i(\overline{y}) - a_i \quad (r = \overline{1,10}).$$

3. Найти новые значения

$$u_i^{s+1}(\overline{y}) = \max(0, u_i^s(\overline{y}) + \delta_{s+1} G_r^i(\overline{y}, u^s(\overline{y}))) \quad (r = \overline{1, 10}),$$

где δ_{s+1} – величина шага.

Таким образом, на каждом шаге субградиентного метода необходимо решить задачи (6.220) при фиксированных $u(\overline{y})$. Функционалы этих задач имеют вид:

- для $r = 1$ и $r = 2$

$$\frac{\sum\limits_{j=1}^{n} \varphi_j(x_j)}{\sum\limits_{j=1}^{n} \psi_j(x_j)} + \sum_{j=1}^{p} \sum_{i=1}^{m} u_i(\overline{y}) g_j^i(x_j),$$

- для $r = 3$ и $r = 4$

$$\sum_{j=1}^{n} \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \sum_{j=1}^{p} \sum_{i=1}^{m} u_i(\overline{y}) g_j^i(x_j),$$

- для $r = 5$ и $r = 6$

$$\max_j \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \sum_{j=1}^{p} \sum_{i=1}^{m} u_i(\overline{y}) g_j^i(x_j),$$

- для $r = 7$ и $r = 8$

$$\frac{\max\limits_j \varphi_j(x_j)}{\min\limits_j \psi_j(x_j)} + \sum_{j=1}^{p} \sum_{i=1}^{m} u_i(\overline{y}) g_j^i(x_j),$$

- для $r = 9$

$$\sum_{j=1}^{n} \varphi_j(x_j) + \sum_{j=1}^{p} \sum_{i=1}^{m} u_i(\overline{y}) g_j^i(x_j),$$

- для $r = 10$

$$\max_j \varphi_j(x_j) + \sum_{j=1}^{p} \sum_{i=1}^{m} u_i(\overline{y}) g_j^i(x_j).$$

Как видно, только для $r = 3$, 4 и 9 функционал (6.220) является сепарабельным и решение задачи сводится к задаче

$$\min_{x_j \in Q_j(\overline{y})} \left\{ \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \sum_{i=1}^{m} u_i(\overline{y}) g_j^i(x_j) \right\} \tag{6.221}$$

или

$$\min_{x_j \in Q_j(\overline{y})} \left\{ \varphi_j(x_j) + \sum_{i=1}^{m} u_i(\overline{y}) g_j^i(x_j) \right\}.$$

При решении задачи (6.221) параметрическим методом фиксируем значение параметра λ по формуле

$$\lambda_j^s = \frac{\varphi_j\big(x_j^*(\overline{y}, u^s)\big)}{\psi_j\big(x_j^*(\overline{y}, u^s)\big)}$$

и тогда получим задачу выпуклого программирования вида

$$\min_{x_j \in Q_j(\overline{y})} \left\{ \varphi_j(x_j) - \lambda_j^s \psi_j(x_j) + \sum_{i=1}^{m} u_i(\overline{y}) g_j^i(x_j) \right\}.$$

Решение задач (6.220) для $r = 1$, 2, 5, 6, 7 и 8 усложняется, так как функционалы не являются сепарабельными функциями. В таких случаях для решения задач (6.220) применяются эвристические алгоритмы, в которых последовательно решаются задачи типа (6.213)–(6.215), а также выпуклые задачи

$$\min_{x_j \in Q_j(\overline{y})} \sum_{i=1}^{m} u_i(\overline{y}) g_j^i(x_j).$$

Предложенные алгоритмы решения задач (6.210) и (6.211) были использованы для решения задач обобщенного дробно-линейного программирования блочно-диагональной структуры со связующими переменными и ограничениями.

6.10. Двухуровневые схемы декомпозиции по ресурсам и по переменным для решения сепарабельных задач обобщенного дробно-выпуклого программирования

Рассмотрим задачи обобщенного дробно-выпуклого программирования блочно-диагональной структуры, в которых функционалы являются сепарабельными по переменным x и y. Для их решения применим схемы декомпозиции по переменным на первом уровне и по ресурсам – на втором уровне алгоритма.

1. Постановка задач. Пусть имеем p групп переменных $x_j = (x_j^1, x_j^2, \ldots, x_j^{n_j}) \in R^{n_j}$, образующих вектор $x = (x_1, x_2, \ldots, x_p) \in R^n$, где $n = \sum\limits_{j=1}^{p} n_j$. На множестве каждой группы переменных x_j определим следующие непрерывные функции $\varphi_j,\ \psi_j,\ g_j^i \quad (i = \overline{1,m})$ и вектор-функции $W_j : R^{m_j} \to R$.

Пусть также имеются переменные $y = (y_1, y_2, \ldots, y_k) \in R^k$, на которых определены функции $\alpha, \beta, F_0^i \quad (i = \overline{1,m})$ и вектор-функции $F_j\ (j = (\overline{1,p}) : R^k \to R$. Предположим, что функции $\varphi_j,\ -\psi_j,\ g_j^i$ $(i = \overline{1,m}),\ W_j\ (j = \overline{1,p})$ выпуклы на множестве значений переменных x, а функции $\alpha,\ -\beta,\ F_0^i\ \ (i = \overline{1,m}),\ F_j\ \ (j = \overline{1,p})$ выпуклы на множестве значений переменных y.

Рассмотрим систему ограничений, которая имеет связующие ограничения и связующие переменные:

$$\sum_{j=1}^{p} g_j^i(x_j) + F_0^i(y) \le b_i \quad (i = \overline{1,m}), \tag{6.222}$$

$$W_j(x_j) + F_j(y) \le \overline{b}_j \quad (j = \overline{1,p}), \tag{6.223}$$

где $b = (b_1, b_2, \ldots, b_m)$ и $\overline{b} = (\overline{b}_1, \overline{b}_2, \ldots, \overline{b}_p)$ – заданные числа.

Через $M(x,y)$ обозначим множество значений переменных (x,y), удовлетворяющие ограничениям (6.222), (6.223), и предположим, что оно ограничено.

Рассмотрим задачи:

$$\min_{(x,y)\in M(x,y)} F_r(x,y) \quad (r = \overline{1,6}), \tag{6.224}$$

в которых $F_r(x,y)$ соответствует одному из следующих функционалов:

$$F_1(x,y) = \sum_{j=1}^{p} \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \frac{\alpha(y)}{\beta(y)},$$

$$F_2(x,y) = \sum_{j=1}^{p} \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \alpha(y),$$

$$F_3(x,y) = \max_j \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \frac{\alpha(y)}{\beta(y)},$$

$$F_4(x,y) = \max_j \frac{\varphi_j(x_j)}{\psi_j(x_j)} + \alpha(y),$$

$$F_5(x,y) = \sum_{j=1}^{p} \varphi_j(x_j) + \frac{\alpha(y)}{\beta(y)},$$

$$F_6(x,y) = \max_j \varphi_j(x_j) + \frac{\alpha(y)}{\beta(y)}.$$

2. Двухуровневый алгоритм. Построим функции Лагранжа на множестве системы ограничений (6.222), (6.223) по общей формуле

$$L_r(x,y,u,v) = F_r(x,y) + \sum_{i=1}^{m} u_i \left[\sum_{j=1}^{p} g_j^i(x_j) + F_0^i(y) - b_i \right] +$$

$$+ \sum_{j=1}^{p} v_j \left[W_j(x_j) + F_j(y) - \overline{b}_j \right] \quad (r = \overline{1,6}),$$

где $u = \{u_i\}$ и $v = \{v_j\}$ – соответствующие множители Лагранжа, u – m-мерный вектор, v_j – m_j-мерные векторы.

Для решения задач (6.224) приведем двухуровневые алгоритмы, в которых на первом уровне для переменных y используется схема декомпозиции по переменным, а на втором для переменных x – схема декомпозиции по ресурсам.

1) *Схема декомпозиции по переменным*

Первый уровень алгоритма не отличается от алгоритма решения задач (6.211), т.е. фиксируются переменные $y = \overline{y}$ и получаются задачи:

$$\Phi_r(\overline{y}) = \min_{x \in M(x,\overline{y})} F_r(x,\overline{y}) \quad (r = \overline{1,6}), \tag{6.225}$$

где $M(x,\overline{y})$ – множество значений переменных x, удовлетворяющих ограничениям (6.222), (6.223) при $y = \overline{y}$. Тогда для минимизации функций $\Phi_r(\overline{y})$ $(r = \overline{1,6})$ используются субградиентные методы, на t-м шаге которых необходимо выполнить следующие три этапа схемы декомпозиции по переменным.

1. Решить задачу (6.225) при фиксированных $\overline{y} = \overline{y}^t$, т.е. найти решения $x(\overline{y}^t)$ и соответствующие им двойственные оценки $u(\overline{y}^t)$ и $v(\overline{y}^t)$.

2. Определить значения обобщенных градиентов по одной из формул:

$$G_r(\overline{y}^t) = \left\{ \frac{G_\alpha(\overline{y}^t)\beta(\overline{y}^t) - G_\beta(\overline{y}^t)\alpha(\overline{y}^t)}{\left[\beta(\overline{y}^t)\right]^2} + \right.$$

$$\left. + \sum_{j=1}^{p} v_j(\overline{y}^t) G_{F_j}(\overline{y}^t) + \sum_{i=1}^{m} u_i(\overline{y}^t) G_{F_0^i}(\overline{y}^t) \right\}, \quad r = 1, 3, 5, 6;$$

$$G_r(\overline{y}^t) = \left\{ G_\alpha(\overline{y}^t) + \sum_{j=1}^{p} v_j(\overline{y}^t) G_{F_j}(\overline{y}^t) + \right.$$

$$\left. + \sum_{i=1}^{m} u_i(\overline{y}^t) G_{F_0^i}(\overline{y}^t) \right\}, \quad r = 2, 4.$$

3. Найти новые значения

$$\overline{y}^{t+1} = \overline{y}^t - h_{t+1} G_r(\overline{y}^t) \quad (r = \overline{1,6}),$$

где h_{t+1} – величина шага.

2) *Схема декомпозиции по ресурсам*

На втором уровне алгоритма применяется схема декомпозиции по ресурсам для решения задач (6.225) при фиксированных $y = \overline{y}$. Так как на первом уровне фиксируются переменные $y = \overline{y}$, то система ограничений (6.222), (6.223) имеет блочно-диагональную структуру на множестве переменных x со связующими ограничениями.

Пусть имеем задачу:

$$F_r(x, \overline{y}) \to \min, \tag{6.226}$$

$$\sum_{j=1}^{p} g_j^i(x_j) \le b_i'(\overline{y}) \quad (i = \overline{1,m}), \tag{6.227}$$

$$W_j(x_j) \le \overline{b}_j'(\overline{y}) \quad (j = \overline{1,p}), \tag{6.228}$$

где $b'_i(\overline{y}) = b_i - F_0^i(\overline{y})$; $\quad \overline{b}'_j(\overline{y}) = \overline{b}_j - F_j(\overline{y})$.

Представим ресурсы $b'_i(\overline{y})$ в виде следующих равенств

$$b'_i(\overline{y}) = z'_{i1}(\overline{y}) + z'_{i2}(\overline{y}) + \cdots + z'_{ip}(\overline{y}) \quad (i = \overline{1, m}). \tag{6.229}$$

В векторной форме равенство (6.229) имеет вид

$$b'(\overline{y}) = z'_1(\overline{y}) + z'_2(\overline{y}) + \cdots + z'_p(\overline{y}),$$

где $z'_j(\overline{y})$ – вектор-столбец выделенных ресурсов j-му блоку.

Тогда задача (6.226)–(6.228) разлагается на p подзадач дробно-выпуклого программирования типа

$$\Phi_j^1(x_j(z'_j(\overline{y}))) = \frac{\varphi_j(x_j)}{\psi_j(x_j)} \to \min, \tag{6.230}$$

$$g_j^i(x_j) \leq z'_{ij}(\overline{y}) \quad (i = \overline{1, m}), \tag{6.231}$$

$$W_j(x_j) \leq \overline{b}'_j(\overline{y}), \tag{6.232}$$

или p подзадач минимизации выпуклого функционала

$$\Phi_j^2(x_j(z'_j(\overline{y}))) = \varphi_j(x_j) \tag{6.233}$$

при ограничениях (6.231), (6.232).

Пусть $x_j(z'_j(\overline{y}))$ – оптимальное решение задачи (6.230)–(6.232) или (6.231)–(6.233) при заданном $z'(\overline{y})$, а $u_{ij}(z'_{ij}(\overline{y}))$, $v_j(z'_{ij}(\overline{y}))$ – значения множителей Лагранжа, соответствующие ограничениям (6.231) и (6.232), где функция Лагранжа определяется по одной из формул:

$$L_j^1(x_j, u_j, v_j) = \frac{1}{\psi_j(x_j)} \Bigg[\varphi_j(x_j) + \sum_{i=1}^{m} u_{ij} \left[g_j^i(x_j) - z'_{ij}(\overline{y}) \right] +$$

$$+ v_j(W_j(x_j) - \overline{b}'_j(\overline{y})) \Bigg],$$

$$L_j^2(x_j, u_j, v_j) = \varphi_j(x_j) + \sum_{i=1}^{m} u_{ij} \left[g_j^i(x_j) - z'_{ij}(\overline{y}) \right] + v_j(W_j(x_j) - \overline{b}'_j(\overline{y})).$$

Определим следующие функции:

$$\Phi_1^s(x(z'(\overline{y}))) = \sum_{j=1}^{p} \Phi_j^s(x_j(z'_j(\overline{y}))), \quad s = 1, 2;$$

$$\Phi_2^s(x(z'(\overline{y}))) = \max_{1 \le j \le p} \Phi_j^s(x_j(z'_j(\overline{y}))), \quad s = 1, 2,$$

которые являются выпуклыми или квазивыпуклыми. Субградиент этих функций соответственно определяется по одной из формул:

$$G_\Phi^1(z'(\overline{y})) = \left\{ g_j(z'_j(\overline{y})) = \frac{-u_j(z'_j(\overline{y}))}{\psi_j(x_j(z'_j(\overline{y})))} \right\},$$

$$G_\Phi^2(z'(\overline{y})) = \left\{ g_j(z'_j(\overline{y})) = -u_j(z'_j(\overline{y})) \right\}.$$

Тогда любая из задач (6.226)–(6.228) сводится к решению задачи минимизации одного из функционалов $\Phi_1^s(x(z'(\overline{y})))$, $s = 1, 2$ или $\Phi_2^s(x(z'(\overline{y})))$, $s = 1, 2$ при линейных ограничениях (6.229). Для решения этих задач может быть использован субградиентный метод с проектированием на линейное многообразие. На k-м шаге выбранного метода необходимо выполнить три этапа схемы декомпозиции по ресурсам.

1. При фиксированных $z'_j(\overline{y}) = z'^k_j(\overline{y})$ решить задачи (6.230)–(6.232) или (6.231)–(6.233) и найти оптимальные решения $x_j(z'_j(\overline{y}))$ и соответствующие двойственные оценки $u_j(z'_j(\overline{y}))$ ограничений (6.231).
2. Определить значения обобщенного градиента функций $\Phi_1^s(x(z'(\overline{y})))$ в точке $z'_j(\overline{y}) = z'^k_j(\overline{y})$ по соответствующей формуле.
3. Найти новые значения $z'^{k+1}_j(\overline{y})$ по формулам выбранного субградиентного метода с учетом выполнения линейных уравнений (6.229).

Следует отметить, что на последнем шаге субградиентного метода при решении задач (6.230)–(6.232) или (6.231)–(6.233) находятся и оптимальные значения $v_j(z'_j(\overline{y}))$ множителей Лагранжа.

Глава 7

Специальные задачи дробного программирования

В данной главе рассматриваются вопросы применения методов недифференцируемой оптимизации для решения некоторых специальных задач дробного программирования. Среди них относится задачи дробно-выпуклого квадратичного программирования, для решения которых предлагается использовать нелинейный параметрический метод и метод подстановке переменных, с использованием методов недифференцируемой оптимизации. Если в первом случае получаем последовательность выпуклых задач квадратичной оптимизации, то во втором – в результате применения схемы декомпозиции по переменным на каждом шаге субградиентного метода решаются линейные задачи.

Применение схем декомпозиции по ограничениям позволяет свести дробно-полиномиальную задачу к решению задач дробно-линейного программирования на каждом шаге субградиентного метода.

Для двухэтапной задачи дробно-стохастического программирования применение схемы декомпозиции по переменным позволяет их свести к субградиентным методам, на каждом шаге которого решаются линейные или дробно-линейные задачи.

Для решения дробно-линейных задач с переменными коэффициентами предлагается использовать метод декомпозиции Данцига-Вулфа, схемы декомпозиции по ограничениям и переменным.

7.1. Задачи дробно-выпуклого квадратичного программирования

Наиболее известными задачами оптимизации являются задачи квадратичного программирования с линейными или квадратичными ограничениями, для решения которых применяются различные по своей сложности алгоритмы и методы [632], [656]–[658], [734]. Один из первых таких методов относится к симплекс-методам решения квадратичных задач с линейными ограничениями [200, 501, 547]. В дальнейшем были предложены и другие подходы решения задач квадратичной выпуклой и нелинейной оптимизации, полиномиальной оптимизации с вещественными и булевскими переменными, получения квадратичных двойственных оценок для различных многоэкстремальных задач [467, 470, 471, 712, 734].

Можно отметить и другие методы и алгоритмы решения задач квадратичной оптимизации, а именно: методы, предназначенные для решения выпуклых задач [198, 212, 215, 405, 518, 527, 557], методы глобальной оптимизации [45, 403, 392] и штрафных функций [411], полиномиальные алгоритмы Кармаркара [558], методы решения квадратичных задач полуопределенного программирования [14], [572]–[574], [245] и методы недифференцируемой оптимизации различных многоэкстремальных квадратичных задач [470, 712, 734].

Одновременно с обычными квадратичными задачами в работах [5, 6, 66, 97, 229, 262, 373, 452, 456, 510] рассмотрены и дробные квадратичные задачи оптимизации с линейными и выпуклыми ограничениями, в которых приводятся необходимые и достаточные условия оптимальности, двойственные задачи, методы преобразования переменных и параметрические методы.

В данном параграфе для решения общей задачи дробно-выпуклого квадратичного программирования предлагаются методы множителей Лагранжа, параметрический нелинейный метод, метод подстановки переменных с применением схем декомпозиции по переменным и методов недифференцируемой оптимизации.

1. Задача дробно-выпуклого квадратичного программирования. Рассмотрим следующую задачу дробно-выпуклого квадратичного программирования:

$$F(x) = \frac{C(x)}{D(x)} \to \min, \tag{7.1}$$

$$K_r(x) \le 0 \quad (r = \overline{1,m}), \tag{7.2}$$

$$x \ge 0, \tag{7.3}$$

в которой: x – n-мерный вектор переменных, $x \in E^n$, $C(x)$, $D(x)$, $K_r(x)$ $(r = \overline{1,k})$ – квадратичные функции, определенные в n-мерном евклидовом пространстве, вида

$$C(x) = (x, Cx) + \alpha x + c_0,$$

$$D(x) = (x, Dx) + \beta x + d_0,$$

$$K_r(x) = (x, A_r x) + \gamma_r x + b_r \quad (r = \overline{1,m}),$$

где C, D, A_r $(r = \overline{1,m})$ – квадратичные и симметричные матрицы размерности $n \times n$; α, β, γ_r $(r = \overline{1,m})$ – n-мерные векторы, а c_0, d_0, b_r $(r = \overline{1,m})$ постоянные величины. В развернутом виде данные функции записываются следующим образом:

$$C(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_i x_j + \sum_{j=1}^{n} \alpha_j x_j + c_0,$$

$$D(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_i x_j + \sum_{j=1}^{n} \beta_j x_j + d_0,$$

$$K_r(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^r x_i x_j + \sum_{j=1}^{n} \gamma_j^r x_j + b_r \quad (r = \overline{1,m}).$$

Если предположим, что матрицы C, A_r $(r = \overline{1,m})$ являются положительно определенными, а D – отрицательно определенная матрица, то функции $C(x)$, $K_r(x)$ $(r = \overline{1,m})$ являются выпуклыми, а функция $D(x)$ – вогнута. При таких предположениях задача квадратичного программирования (7.1)–(7.3) представляет собой задачу дробно-выпуклого программирования, а функция $F(x)$ является квазивыпуклой функцией.

Для решения задачи (7.1)–(7.3) в общем случае могут быть использованы различные методы решения задачи дробно-выпуклого программирования, которые были ранее рассмотрены. Их можно разделить на следующие группы:

- методы множителей Лагранжа и необходимые и достаточные условия оптимальности;
- нелинейные параметрические методы;
- методы подстановке переменных;
- методы штрафных и барьерных функций;

- методы множителей Лагранжа и недифференцируемая оптимизация нахождения седловой точке;
- схемы декомпозиции по переменных и ограничениям и недифференцируемая оптимизация.

Применение таких методов не выводит задачи (7.1)–(7.3) из класса задач квадратичного программирования. Как правило, в данных методах на некотором этапе необходимо решать одну задачу квадратичной оптимизации.

Однако в каждом конкретном методе существуют некоторые особенности, которые позволяют разработать более эффективные методы и алгоритмы решения дробных квадратичных задач.

Рассмотрим некоторые из таких методов и алгоритмов.

2. Метод множителей Лагранжа. Для задачи (7.1)–(7.3), которая является задачей дробно-выпуклого программирования, функцию Лагранжа рассмотрим в следующем виде

$$L_1(x,u) = \frac{C(x)}{D(x)} + \sum_{r=1}^{m} u_r K_r(x).$$

В соответствии с теоремой Куна-Таккера (см. теоремы 1.20 и 1.21) рассмотрим необходимые и достаточные условия оптимальности для задачи (7.1)–(7.3) в виде системы уравнений:

$$\frac{\partial L}{\partial x_j} - v_j = 0 \quad (j = \overline{1,n}); \qquad \frac{\partial L}{\partial u_r} + y_r = 0 \quad (r = \overline{1,m});$$

$$x_j\, v_j = 0 \quad (j = \overline{1,n}); \qquad u_r\, y_r = 0 \quad (r = \overline{1,m}),$$

в которых $v = (v_1, v_2, \ldots, v_n)$ и $y = (y_1, y_2, \ldots, y_m)$ – векторы дополнительных переменных, $v_j \geq 0 \quad (j = \overline{1,n})$ и $y_r \geq 0 \quad (r = \overline{1,m})$.

Для задачи (7.1)–(7.3) они имеют вид

$$\begin{cases} D(x)\dfrac{\partial C(x)}{\partial x_j} - C(x)\dfrac{\partial D(x)}{\partial x_j} + D^2(x)\displaystyle\sum_{r=1}^{m} u_r \dfrac{\partial K_r(x)}{\partial x_j} - v_j = 0 \quad (j = \overline{1,n}), \\ K_r(x) + y_r = 0 \quad (r = \overline{1,m}), \\ x_j\, v_j = 0 \quad (j = \overline{1,n}); \quad u_r\, y_r = 0 \quad (r = \overline{1,m}). \end{cases} \tag{7.4}$$

Если функцию Лагранжа для задачи (7.1)–(7.3) определить по формуле

$$L_2(x,u) = \frac{C(x) + \sum\limits_{r=1}^{m} u_r K_r(x)}{D(x)},$$

то данные условия имеют вид следующей системе уравнений:

$$\begin{cases} \dfrac{\partial C(x)}{\partial x_j} - \dfrac{\partial D(x)}{\partial x_j} L_2(x,u) + \displaystyle\sum_{r=1}^{m} u_r \dfrac{\partial K_r(x)}{\partial x_j} - v_j = 0 \quad (j = \overline{1,n}); \\ K_r(x) + y_r = 0 \quad (r = \overline{1,m}); \\ x_j\, v_j = 0 \quad (j = \overline{1,n}); \quad u_r\, y_r = 0 \quad (r = \overline{1,m}). \end{cases} \tag{7.5}$$

Условия (7.4) или (7.5) имеют нелинейный характер и для их решения можно использовать общие методы решения нелинейных систем уравнений.

Один из способов ”линеаризации” данных условий оптимальности является применение параметрического метода для решения задач дробно-выпуклого квадратичного программирования.

3. Нелинейный параметрический метод. В параметрическом методе решение задач дробно-выпуклого программирования исходная задача (7.1)–(7.3) сводится к решению задачи выпуклого квадратичного программирования параметрического вида:

$$F(\lambda) = \min_{x \in S(x)} Z(x,\lambda) = C(x) - \lambda D(x), \tag{7.6}$$

в которой $S(x)$ представляет собой множество допустимых решений для ограничений (7.2), (7.3), а именно

$$S(x) = \{x \geq 0 : K_r(x) \leq 0 \quad (r = \overline{1,m})\}.$$

Тогда задача дробно-выпуклого программирования сводится к решению уравнения $F(\lambda) = 0$ или $C(x) - \lambda D(x) = 0$. Решение данного уравнения сводится к итеративной процедуре, в которой для каждого фиксированного значения параметра λ решается задача квадратичного выпуклого программирования (7.6).

Процедура начинается с $\lambda = \lambda_0 = 0$ и находим оптимальное решение $x^*(\lambda_0)$ задачи (7.6). Пусть после k шагов имеем решение $x^*(\lambda_k)$ задачи (7.6). Тогда $(k+1)$-й шаг процедуры состоит из следующих этапов:

- определяем очередное значение параметра λ по формуле

$$\lambda_{k+1} = F\big(x^*(\lambda_k)\big) = \frac{C\big(x^*(\lambda_k)\big)}{D\big(x^*(\lambda_k)\big)};$$

- если $\lambda_{k+1} = \lambda_k$, то λ_k является решением уравнения $F(\lambda) = 0$, а $x^*(\lambda_k)$ – решением исходной задаче дробно-выпуклого квадратичного программирования (7.1)–(7.3);

- если $\lambda_{k+1} \neq \lambda_k$, то решаем очередную задачу выпуклого квадратичного программирования (7.6) при $\lambda = \lambda_{k+1}$.

В данном случае нелинейные условия оптимальности (7.4) и (7.5) имеют одинаковую структуру, так как они определены для функции Лагранжа параметрической задачи (7.6), а именно

$$L(x, u, \lambda) = C(x) - \lambda D(x) + \sum_{r=1}^{m} u_r K_r(x).$$

Тогда данные условия оптимальности имеют вид

$$\begin{cases} \dfrac{\partial C(x)}{\partial x_j} - \lambda \dfrac{\partial D(x)}{\partial x_j} + \sum\limits_{r=1}^{m} u_r \dfrac{\partial K_r(x)}{\partial x_j} - v_j = 0 \quad (j = \overline{1, n}), \\ K_r(x) + y_r = 0 \quad (r = \overline{1, m}), \\ x_j v_j = 0 \quad (j = \overline{1, n}); \quad u_r y_r = 0 \quad (r = \overline{1, m}). \end{cases}$$

В общем случае задача (7.6) является задачей многоэкстремального квадратичного программирования. Для ее решения можно использовать различные алгоритмы и методы глобальной оптимизации [88], [194]–[196], [255]–[257], [521, 278].

Если предположить, что ограничения (7.2) представляют собой систему линейных неравенств или уравнений

$$\sum_{j=1}^{n} a_{rj} x_j \leq b_r \quad (r = \overline{1, m}),$$

то при фиксированных $\lambda = \lambda_k$, на каждом шаге параметрического метода вместо задачи (7.6) получим задачу линейного программирования (см. п. 1.8) следующего вида

$$Z(\xi, \eta) = \sum_{j=1}^{n} M \xi_j + \sum_{r=1}^{m} M \eta_r \to \min,$$

при ограничениях

$$\sum_{j=1}^{n} a_{rj} x_j + y_r + \eta_r = b_r \quad (r = \overline{1, m}),$$

$$\sum_{i=1}^{n} (c_{ij} - \lambda d_{ij}) x_i + \sum_{r=1}^{m} a_{rj} u_r - v_j + \xi_j = -(\alpha_j - \lambda_k \beta_j) \quad (j = \overline{1, n}),$$

$$x_j \geq 0, \quad v_j \geq 0, \quad \xi_j \geq 0 \quad (j = \overline{1,n}),$$
$$u_r \geq 0, \quad y_r \geq 0, \quad \eta_r \geq 0 \quad (r = \overline{1,m}),$$

где $M > 0$ – достаточно большое число.

4. Метод подстановке переменных. Рассмотрим для начало следующий частный случай задачи дробно-выпуклого квадратичного программирования (7.1)–(7.3), а именно

$$Q(x) = \frac{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} c_{ij}x_ix_j}{\left(\sum\limits_{j=1}^{n}\beta_jx_j + d_0\right)^2} \to \min, \tag{7.7}$$

$$\sum_{j=1}^{n} a_{rj}x_j \leq b_r \quad (r = \overline{1,m}), \tag{7.8}$$

$$x_j \geq 0 \quad (j = \overline{1,n}), \tag{7.9}$$

в которой $\sum\limits_{j=1}^{n}\beta_jx_j + d_0 > 0$.

Для решения данной задачи применим метод подстановке переменных ее сведения к задаче выпуклого квадратичного программирования с линейными ограничениями. Для этого введем обозначения $t = \dfrac{1}{\sum\limits_{j=1}^{n}\beta_jx_j + d_0}$ и $y_j = x_j \cdot t \quad (j = \overline{1,n})$. Тогда получим задачу:

$$P(y,t) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}y_iy_j \to \min, \tag{7.10}$$

$$\sum_{j=1}^{n} a_{rj}y_j - b_rt \leq 0 \quad (r = \overline{1,m}), \tag{7.11}$$

$$\sum_{j=1}^{n}\beta_jy_j + d_0t = 1, \tag{7.12}$$

$$t \geq 0, \quad y_j \geq 0 \quad (j = \overline{1,n}), \tag{7.13}$$

которая является задачей квадратичного программирования с линейными ограничениями.

Для полученной задачи (7.10)–(7.13) можно использовать методы и алгоритмы решения обычных задач квадратичной оптимизации (см. п. 1.8).

Тогда, если (y^*, t^*) является решением задачи (7.10)–(7.13), то $x^* = y^*/t^*$ является решением дробно-выпуклой задачи квадратичного программирования (7.7)–(7.9).

Рассмотрим теперь задачу дробно-выпуклого программирования (7.1)–(7.3) для которой предположим, что $D(x) > 0$ для любого $x \in S(x)$. Для решения данной задаче применим метод подстановке переменных с целью получения некоторой задачи не дробной оптимизации. Введем новую переменную

$$t = \frac{1}{D(x)}.$$

Если использовать переменную t, определенную по данной формуле, то функции $C(x)$, $D(x)$, $K_r(x)$ $(r = \overline{1,m})$ из задачи (7.1)–(7.3) могут быть записаны в следующем виде:

$$C(x,t) = \big(x, C(xt)\big) + \alpha xt + c_0 t,$$

$$D(x,t) = \big(x, D(xt)\big) + \beta xt + d_0 t,$$

$$K_r(x,t) = \big(x, A_r(xt)\big) + \gamma_r xt + b_r t \quad (r = \overline{1,m}).$$

Если ввести переменные $y = xt$, то задача дробно-выпуклого программирования (7.1)–(7.3) сводится к новой задаче следующего вида

$$C(x,y,t) = (x, Cy) + \alpha y + c_0 t \to \min, \tag{7.14}$$

$$D(x,y,t) = (x, Dy) + \beta y + d_0 t = 1, \tag{7.15}$$

$$K_r(x,y,t) = (x, A_r y) + \gamma_r y + b_r t \le 0 \quad (r = \overline{1,m}), \tag{7.16}$$

$$x \ge 0, \quad y \ge 0, \quad t \ge 0, \tag{7.17}$$

В новых обозначениях имеем билинейные функции

$$C(x,y,t) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_i y_j + \sum_{j=1}^{n} \alpha_j y_j + c_0 t,$$

$$D(x,y,t) = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} x_i y_j + \sum_{j=1}^{n} \beta_j y_j + d_0 t,$$

$$K_r(x, y, t) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^r x_i y_j + \sum_{j=1}^{n} \gamma_j^r y_j + b_r t \quad (r = \overline{1, m}).$$

Задача (7.14)–(7.17) представляет собой билинейную задачу квадратичного программирования, в которой имеются два типа переменных x и y, которые связаны между собой соотношением $y = x \cdot t$ и равенством (7.15).

Аналогичные задачи с двумя группами переменных рассмотрены в предыдущих главах, для решения которых применялась схемы декомпозиции по переменным и методы недифференцируемой оптимизации. Рассмотрим аналогичные процедуры для решения задачи (7.14)–(7.17), применяя схему декомпозиции по переменным x. Для этого фиксируем значения переменных $x = \overline{x}$ и рассмотрим задачу линейного программирования

$$\varphi(\overline{x}) = p(\overline{x})y + c_0 t \to \min, \tag{7.18}$$

$$\psi(\overline{x})y + d_0 t = 1, \tag{7.19}$$

$$q_r(\overline{x})y + b_r t \leq 0 \quad (r = \overline{1, m}), \tag{7.20}$$

$$y \geq 0, \quad t \geq 0, \tag{7.21}$$

в которой использованы обозначения:

$$p(\overline{x}) = (\overline{x}, C) + \alpha, \quad \psi(\overline{x}) = (\overline{x}, D) + \beta,$$

$$q_r(\overline{x}) = (\overline{x}, A_r) + \gamma_r \quad (r = \overline{1, m}).$$

Если вычислить каждый элемент векторов и матриц в отдельности по формулам

$$p_j(\overline{x}) = \sum_{i=1}^{n} c_{ij} \overline{x}_i + \alpha_j \ \ (j = \overline{1, n}), \quad \psi_j(\overline{x}) = \sum_{i=1}^{n} d_{ij} \overline{x}_i + \beta_j \ \ (j = \overline{1, n}),$$

$$q_j^r(\overline{x}) = \sum_{i=1}^{n} a_{ij}^r \overline{x}_i + \gamma_j^r \quad (r = \overline{1, m}; j = \overline{1, n}),$$

то получим следующую задачу линейного программирования

$$\sum_{j=1}^{n} p_j(\overline{x}) y_j + c_0 t \to \min,$$

$$\sum_{j=1}^{n} \psi_j(\overline{x}) y_j + d_0 t = 1,$$

$$\sum_{j=1}^{n} q_j^r(\overline{x}) y_j + b_r t \le 0 \quad (r = \overline{1,m}),$$
$$y_j \ge 0 \quad (j = \overline{1,n}), \quad t \ge 0.$$

В методе декомпозиции по переменным задача (7.18)–(7.21) является внутренней задачей, которая каждый раз решается при фиксированных (обновленных) значениях переменных x. Вычисляя каждый раз значения коэффициентов, которые формируют вектора $p(\overline{x})$, $\psi(\overline{x})$ и $q_r(\overline{x})$ $(r = \overline{1,m})$, то задача (7.18)–(7.21) представляет собой обычную задачу линейного программирования.

Внешняя задача в схеме декомпозиции по переменным имеет вид:

$$\varphi(\overline{x}) \to \min, \tag{7.22}$$
$$\overline{x} \ge 0. \tag{7.23}$$

Из определения, функция $\varphi(\overline{x})$ является частично-линейной, выпуклой и недифференцируемой. Для решения задачи (7.22), (7.23) применяем один из субградиентных методов, на $(k+1)$-м шаге которого выполняем следующие три основных этапа:

1) решаем линейную задачу (7.18)–(7.21) при фиксированных значениях переменных $x = \overline{x} = x^*(k)$ и находим ее оптимальное решение $\{y^*\big(x^*(k)\big),\, t^*\}$;
2) вычисляем значения субградиента функции $\varphi(\overline{x})$ в точке $x = \overline{x} = x^*(k)$ по формуле
$$g_\varphi(\overline{x}) = Cy + vDy + \sum_{r=1}^{m} u_r A_r y,$$
полученную в соответствии с функцией Лагранжа для задачи (7.14)–(7.17) или (7.18)–(7.21), которая имеет вид
$$L(x,y,t,v,u) = (x, Cy) + \alpha y + c_0 t + v\big((x, Dy) + \beta y + d_0 t - 1\big) + \\ + \sum_{r=1}^{m} u_r\big((x, A_r y) + \gamma_r y + b_r t\big),$$
3) находим значения новой точки $\overline{x}(k+1)$ по формуле
$$\overline{x}(k+1) = \max\{0,\, \overline{x}(k) - h_{k+1} g_\varphi(\overline{x})\},$$
где h_{k+1} – величина шага в субградиентном методе.

Теорема 7.1. *Если $x^*(k)$ является решением задачи* (7.22), (7.23), *а $\{y^*\big(x^*(k)\big),\, t^*\}$ – решением задачи* (7.18)–(7.21) *при $x = \overline{x} = x^*(k)$, то $x = y^*\big(x^*(k)\big)/t^*$ является решением задачи дробно-выпуклого квадратичного программирования* (7.1)–(7.3).

7.2. Задача дробно-полиномиальной оптимизации

В работах [467, 470, 722, 723, 727, 735] изучены задачи полиномиальной оптимизации, которые были сведены к задачам квадратичного программирования. В работе [470] была сформулирована и дробно-полиномиальная задача, без детализации схем ее решения. В данном пункте для задачи дробно-полиномиальной оптимизации используем схему ее сведения к задачи дробно-квадратичного программирования и субградиентные методы [482].

Пусть задана задача дробной оптимизации, в которой числитель, знаменатель и ограничения являются полиномами, определенные для переменных $x = (x_1, x_2, \ldots, x_n) \in E^n$ и которые для данной задачи имеют вид:

$$C(x) = \sum_{r \in R_C} c_r \prod_{j=1}^{n} x_j^{\alpha_{rj}},$$

$$D(x) = \sum_{r \in R_D} d_r \prod_{j=1}^{n} x_j^{\beta_{rj}},$$

$$P_i(x) = \sum_{r \in R_{P_i}} a_{ir} \prod_{j=1}^{n} x_j^{\gamma_{rj}^i} \quad (i = \overline{1, m}),$$

в которых: R_C – множество индексов мономов с ненулевыми коэффициентами c_r, которые формируют полином $C(x)$; R_D – множество индексов мономов с ненулевыми коэффициентами d_r, которые формируют полином $D(x)$; R_{P_i} $(i = \overline{1, m})$ – множество индексов мономов с ненулевыми коэффициентами a_{ir}, которые формируют полиномы $P_i(x)$; c_r, d_r, a_{ir} $(i = \overline{1, m})$ – ненулевые коэффициенты мономов, которые соответственно содержатся в полиномах $C(x)$, $D(x)$ и $P_i(x)$ $(i = \overline{1, m})$, а также целые положительные числа α_{rj} $(r \in R_C,\ j = \overline{1, n})$, β_{rj} $(r \in R_D,\ j = \overline{1, n})$ и γ_{rj}^i $(r \in R_{P_i};\ i = \overline{1, m};\ j = \overline{1, n})$, которые представляют собой степени соответствующих переменных x_j $(j = \overline{1, n})$, входящие в те или иные мономы, формирующие полиномы $C(x)$, $D(x)$ и $P_i(x)$ $(i = \overline{1, m})$.

В развернутом виде любой полином представляет собой сумму различных мономов любой степени. Запишем рассматриваемые полиномы более детально. Тогда для $C(x)$ имеем запись

$$\begin{aligned} C(x) = c_1 \cdot x_1^{\alpha_{11}} \cdot x_2^{\alpha_{12}} \cdot x_3^{\alpha_{13}} \cdot \ldots \cdot x_n^{\alpha_{1n}} &+ \\ + c_2 \cdot x_1^{\alpha_{21}} \cdot x_2^{\alpha_{22}} \cdot x_3^{\alpha_{23}} \cdot \ldots \cdot x_n^{\alpha_{2n}} &+ \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots& \\ + c_k \cdot x_1^{\alpha_{k1}} \cdot x_2^{\alpha_{k2}} \cdot x_3^{\alpha_{k3}} \cdot \ldots \cdot x_n^{\alpha_{kn}}&, \end{aligned}$$

где $k = ||R_C||$ – количество мономов с ненулевыми коэффициентами c_r $(r = \overline{1,k})$ для полинома $C(x)$. Степень полинома $C(x)$ определяется по формуле

$$N_C = \max_{1 \le r \le k} \sum_{j=1}^{n} \alpha_{rj}.$$

Если N_C – четное число, то $C(x)$ является полиномом четной степени, в противном случае – нечетной степени.

Аналогичное можно записывать для полинома $D(x)$ и полиномов $P_i(x)$ $(i = \overline{1,m})$. Полином $D(x)$ имеет следующую запись

$$\begin{aligned} D(x) = d_1 \cdot x_1^{\beta_{11}} \cdot x_2^{\beta_{12}} \cdot x_3^{\beta_{13}} \cdot \ldots \cdot x_n^{\beta_{1n}} &+ \\ + d_2 \cdot x_1^{\beta_{21}} \cdot x_2^{\beta_{22}} \cdot x_3^{\beta_{23}} \cdot \ldots \cdot x_n^{\beta_{2n}} &+ \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots& \\ + d_s \cdot x_1^{\beta_{s1}} \cdot x_2^{\beta_{s2}} \cdot x_3^{\beta_{s3}} \cdot \ldots \cdot x_n^{\beta_{sn}}&, \end{aligned}$$

где $s = ||R_D||$ – количество мономов с ненулевыми коэффициентами d_r $(r = \overline{1,s})$ для полинома $D(x)$. Степень полинома $D(x)$ определяется по формуле

$$N_D = \max_{1 \le r \le s} \sum_{j=1}^{n} \beta_{rj}.$$

Число N_D определяет степень полинома $D(x)$, который также может быть четной или нечетной степени.

Полиномы $P_i(x)$ $(i = \overline{1,m})$ имеют записи

$$\begin{aligned} P_i(x) = a_{i1} \cdot x_1^{\gamma^i_{11}} \cdot x_2^{\gamma^i_{12}} \cdot x_3^{\gamma^i_{13}} \cdot \ldots \cdot x_n^{\gamma^i_{1n}} &+ \\ + a_{i2} \cdot x_1^{\gamma^i_{21}} \cdot x_2^{\gamma^i_{22}} \cdot x_3^{\gamma^i_{23}} \cdot \ldots \cdot x_n^{\gamma^i_{2n}} &+ \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots& \\ + a_{il_i} \cdot x_1^{\gamma^i_{l_i 1}} \cdot x_2^{\gamma^i_{l_i 2}} \cdot x_3^{\gamma^i_{l_i 3}} \cdot \ldots \cdot x_n^{\gamma^i_{l_i n}}&, \end{aligned}$$

где $l_i = ||R_{P_i}||$ – количество мономов с ненулевыми коэффициентами a_{ir} $(r = \overline{1, l_i})$ для полиномов $P_i(x)$ $(i = \overline{1, m})$. Степени полиномов $P_i(x)$ определяются по формуле

$$N_{P_i} = \max_{1 \leq r \leq l_i} \sum_{j=1}^{n} \gamma_{rj}^i.$$

Числа N_{P_i}, которые определяют степени полиномов $P_i(x)$, также могут быть как четными, так и нечетными, и тогда полиномы $P_i(x)$ являются полиномами четной или нечетной степени.

Тогда задачи оптимизации, в которых входят такие полиномы, будут задачами полиномиальной оптимизации четной или нечетной степени.

1. Дробно-полиномиальная задача. Задачи полиномиальной оптимизации рассмотрены в различных работах [470, 722, 723, 727, 735], для решения которых рассматриваются схемы и процедуры их сведения к задачам квадратичного программирования с применением методов недифференцируемой оптимизации. В сделанных выше обозначениях рассмотрим следующую задачу оптимизации отношения двух полиномов

$$F(x) = \frac{C(x)}{D(x)} \to \min \tag{7.24}$$

при ограничениях в виде полиномов

$$P_i(x) = 0 \quad (i = \overline{1, m}). \tag{7.25}$$

Задача (7.24), (7.25) является задачей дробно-полиномиальной оптимизации. Для данной задачи определяем:

$N_C = \max\limits_{r \in R_C} \sum\limits_{j=1}^{n} \alpha_{rj}$ – степень полинома $C(x)$;

$N_D = \max\limits_{r \in R_D} \sum\limits_{j=1}^{n} \beta_{rj}$ – степень полинома $D(x)$;

$N_{P_i} = \max\limits_{r \in R_{P_i}} \sum\limits_{j=1}^{n} \gamma_{rj}^i \quad (i = \overline{1, m})$ – степень полиномов $P_i(x)$.

Предположим, что все степени N_C, N_D, N_{P_i} являются четными числами. Находим $N = \max(N_C, N_D, \max\limits_{1 \leq i \leq m} N_{P_i})$ – максимальная степень полиномов задачи (7.24), (7.25).

Тогда задача (7.24), (7.25) является задачей полиномиальной оптимизации четной степени N с дробным функционалом, которая является общей задачей математического программирования глобальной оптимизации.

Для ее решения может быть использован метод сведения к задаче дробно-квадратичного программирования. Для этого вместо каждой переменной x_j $(j = \overline{1,n})$ вводим N новых переменных $x_j^{(1)}, x_j^{(2)}, \ldots, x_j^{(N)}$ и $N-1$ дополнительных ограничений, которые связывают данные переменные между собой с помощью некоторой цепочки равенств. Переменные $x_j^{(k)}$ имеют единичную степень, для записи которых используются два индекса, а именно индекс j, который соответствует индексу переменной x_j и индекс k, который соответствует порядковому номеру повторения переменной x_j до максимальной степени N полиномов $C(x)$, $D(x)$ и $P_i(x)$ $(i = \overline{1,m})$. Таким образом, каждая переменная x_j входит в N равенств, связанные между собой пошаговыми равенствами.

Вводим следующие тождественные соотношения между новыми переменными $x_j^{(k)}$ $(j = \overline{1,n};\ k = \overline{1,N})$ и переменными x_j различной степени:

$$\begin{aligned}
&x_j \equiv x_j^{(1)},\\
&x_j^2 \equiv x_j^{(2)} = x_j^{(1)} \cdot x_j^{(1)} \Rightarrow x_j^{(2)} - x_j^{(1)} \cdot x_j^{(1)} = 0,\\
&x_j^3 \equiv x_j^{(3)} = x_j^{(2)} \cdot x_j^{(1)} \Rightarrow x_j^{(3)} - x_j^{(2)} \cdot x_j^{(1)} = 0,\\
&x_j^4 \equiv x_j^{(4)} = x_j^{(3)} \cdot x_j^{(1)} \Rightarrow x_j^{(4)} - x_j^{(3)} \cdot x_j^{(1)} = 0,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&x_j^k \equiv x_j^{(k)} = x_j^{(k-1)} \cdot x_j^{(1)} \Rightarrow x_j^{(k)} - x_j^{(k-1)} \cdot x_j^{(1)} = 0,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&x_j^N \equiv x_j^{(N)} = x_j^{(N-1)} \cdot x_j^{(1)} \Rightarrow x_j^{(N)} - x_j^{(N-1)} \cdot x_j^{(1)} = 0.
\end{aligned}$$

Через данными соотношениями любая степенная переменная x_j^k может быть представлена в виде произведением двух новых переменных единичной степени, т.е. данные соотношения имеют степень равную двум.

В общем виде они могут быть представлены следующим образом:

$$x_j^{(k)} - x_j^{(k-1)} \cdot x_j^{(1)} = 0 \quad (k = \overline{2,N};\ j = \overline{1,n}). \tag{7.26}$$

Соотношение (7.26) показывает связь между очередной переменной $x_j^{(k)}$, соответствующая переменной x_j в k-й степени, представленной через предыдущую переменную $x_j^{(k-1)}$ и первой переменной $x_j^{(1)}$, которая и является переменной x_j.

С помощью отношений (7.26) все полиномы задачи (7.24), (7.25) могут быть приведены к виду:

$$C(X) = \sum_{r \in R_C} c_r \prod_{j=1}^{n} x_j^{(\overline{\alpha}_{rj})},$$

$$D(X) = \sum_{r \in R_D} d_r \prod_{j=1}^{n} x_j^{(\overline{\beta}_{rj})},$$

$$P_i(X) = \sum_{r \in R_{P_i}} a_{ir} \prod_{j=1}^{n} x_j^{(\overline{\gamma}^i_{rj})} \quad (i = \overline{1,m}),$$

в которых вместо степенях α_{rj}, β_{rj} и γ^i_{rj} используются соответствующие этим степеням числа $\overline{\alpha}_{rj}$, $\overline{\beta}_{rj}$ и $\overline{\gamma}^i_{rj}$, которые представляют собой индексы новых переменных $x_j^{(k)}$, имеющие только единичные степени и формирующие новую матрицу переменных $X = (x_j^{(k)})_{n \times N}$.

В новых переменных все мономы представляют собой произведения различных множеств переменных $x_j^{(k)}$ первой степени. В дальнейшем, введя поочередно новые переменные с помощью соотношений

$$x_{st}^{(k_s,k_t)} = x_s^{(k_s)} \cdot x_t^{(k_t)} \quad (s = \overline{1,n};\ t = \overline{1,n}), \tag{7.27}$$

степень мономов может быть сведена до двух. Действительно, по соотношениям (7.27) переменные единичной степени $x_s^{(k_s)}$ и $x_t^{(k_t)}$ попарно группируются и вместо их произведения вводится новая переменная $x_{st}^{(k_s,k_t)}$, которая имеет уже единичную степень. В результате этой операции получаем новую четырехмерную матрицу переменных $X = \left(x_{st}^{(k_s,k_t)}\right)$, которые уменьшают ровно в два раза степень каждого монома, и соответственно, каждого полинома $C(X)$, $D(X)$ и $P_i(X)$ $(i = \overline{1,m})$. В такой матрице X не все переменные определены, а только те, которые соответствуют номерам индексов s, t, k_s и k_t. Количество переменных по формуле (7.27) также уменьшаются в два раза. Повторяя эту процедуру несколько раз получим, что каждый полином имеет степень равную двум. Таким образом, вместо полиномов максимальной четной степени N, получаем квадратичные функции с небольшим числом переменных, соответственно по две переменные для каждого монома.

В последних обозначениях полиномы исходной задачи (7.24), (7.25) имеют следующие квадратичные структуры:

$$C(x) \equiv C(X) = \sum_{r \in R_C} c_r \cdot x_{s_r} \cdot x_{t_r},$$

где индексы s_r и t_r соответствуют номерам переменных в новых обозначениях, к произведением которых сводится r-й моном полинома $C(x),\ r \in R_C$;

$$D(x) \equiv D(X) = \sum_{r \in R_D} d_r \cdot x_{s_r} \cdot x_{t_r},$$

где индексы s_r и t_r соответствуют номерам переменных в новых обозначениях, к произведением которых сводится r-й моном полинома $D(x),\ r \in R_D$;

$$P_i(x) \equiv P_i(X) = \sum_{r \in R_{P_i}} a_{ir} \cdot x_{s_{ir}} \cdot x_{t_{ir}} \quad (i = \overline{1,m}),$$

где индексы s_{ir} и t_{ir} соответствуют номерам переменных в новых обозначениях, к произведением которых сводится r-й моном полинома $P_i(x)\ \ (i = \overline{1,m});\ r \in R_{P_i}$.

Таким образом, исходная задача (7.24), (7.25) может быть сведена к некоторой задаче дробно-квадратичного программирования с новыми переменными и с новыми дополнительными ограничениями типа (7.26) и (7.27), по которым проводится обратный пересчет значений введенных переменных. Для решения такой задачи может быть использована схема декомпозиции по ограничениям с применением субградиентных методов. Ниже рассмотрим применение такого метода для решения дробно-полиномиальной задачи с одной переменной.

2. Дробно-полиномиальная задача с одной переменной. Рассмотрим частный случай задачи (7.24), (7.25), в которой все полиномы определены для одной переменной x_1. Пусть задана задача:

$$\frac{C(x_1)}{D(x_1)} = \frac{\sum\limits_{k=1}^{n} c_k x_1^k}{\sum\limits_{k=1}^{n} d_k x_1^k} \to \min, \tag{7.28}$$

$$\sum_{k=1}^{n} a_{ik} x_1^k = 0 \quad (i = \overline{1,m}), \tag{7.29}$$

где n является четным числом и представляет степень полиномов, т.е. по крайней мере один из коэффициентов c_n, d_n или $a_{in}\ \ (i = \overline{1,m})$ отличен от нуля.

Введем новые переменные $x_2, \dots, x_n$, связанные между собой равенствами:

$$x_1^2 \equiv x_2 = x_1 \cdot x_1,$$
$$x_1^3 \equiv x_3 = x_2 \cdot x_1,$$
$$\dots\dots\dots\dots\dots\dots$$
$$x_1^n \equiv x_n = x_{n-1} \cdot x_1$$

или в общем случае

$$x_1^k \equiv x_k = x_{k-1} \cdot x_1 \quad (k = \overline{2, n}). \tag{7.30}$$

Тогда задача (7.28), (7.29) имеет вид:

$$\frac{C(x)}{D(x)} = \frac{\sum\limits_{k=1}^{n} c_k x_k}{\sum\limits_{k=1}^{n} d_k x_k} \to \min, \tag{7.31}$$

$$\sum_{k=1}^{n} a_{ik} x_k = 0 \quad (i = \overline{1, m}), \tag{7.32}$$

$$x_k - x_{k-1} \cdot x_1 = 0 \quad (k = \overline{2, n}). \tag{7.33}$$

Для решения задачи (7.31)–(7.33) применяем схему декомпозиции по ограничениям с использованием методов недифференцируемой оптимизации. Для этого построим функцию Лагранжа по нелинейным ограничениям (7.33):

$$L(x, u) = \frac{\sum\limits_{k=1}^{n} c_k x_k + \sum\limits_{k=2}^{n} u_k (x_k - x_{k-1} \cdot x_1)}{\sum\limits_{k=1}^{n} d_k x_k}$$

и рассмотрим следующие две задачи

$$L^*(u) = \min_{x \in R(x)} L(x, u), \tag{7.34}$$

$$\max_u L^*(u), \tag{7.35}$$

где $u = (u_2, \dots, u_n)$ являются двойственными переменными для ограничений (7.33), а $R(x)$ – допустимое множество, определяемое ограничениями (7.32).

Для решения задачи (7.34) недифференцируемой оптимизации используем субградиентный метод.

Пусть на t-й итерации определены значения u^t двойственных переменных u. Тогда на $(t+1)$-м шаге субградиентного метода выполняем три основных этапа.

1. Определяем оптимальное решение $x^*(u^*)$ задачи (7.35) при фиксированных значениях $u = u^t$, которая в общем представляет собой задачу дробно-линейного программирования.

2. Определяем значения обобщенного градиента функции $L^*(u)$ в точке u^t по формуле

$$g_k(u^t) = x_k(u^t) - x_{k-1}(u^t) \cdot x_1(u^t).$$

3. Находим новые значения переменных u_k^{t+1} по формуле

$$u_k^{t+1} = u_k^t + \gamma_{t+1} g_k(u^t) \quad (k = 2, \ldots, n).$$

Пусть u^* является оптимальным решением задачи (7.35). Тогда значение обобщенного градиента $g(u^*)$ равна нулю. Таким образом, в точке u^* обеспечивается выполнение ограничений (7.33) для оптимального решения $x^*(u^*)$ задачи (7.34) при $u_k = u_k^*$. Так как задачи (7.31)–(7.33) и (7.28)–(7.33) эквивалентны, то решение $x^*(u^*)$ будет оптимальна для обеих задач.

7.3. Двухэтапная задача дробно-стохастического программирования

Задачи стохастического программирования условно можно разделить на две группы. К первой из них относятся задачи оптимизации, для которых решения находятся стохастическими методами, или задачи, в которых имеются вероятностные ограничения [81, 183, 184, 281, 282, 413, 651, 743]. Ко второй группе отнесем те задачи оптимизации, в которых данные задаются с помощью некоторых случайных чисел. Такие постановки приводят к двухэтапным или многоэтапным задачам стохастического программирования [73, 74, 162, 167, 246, 602, 605, 615, 728, 737, 738], для решения которых применяются декомпозиционные методы. К таким методам относятся схемы декомпозиции по ограничениям с применением субградиентных методов [728, 737, 738].

Аналогичные стохастические задачи для дробно-линейного программирования рассмотрены в работах [120]–[130], [238, 483], в которых приведены как стохастические методы, так и декомпозиционные методы решения двухэтапных задач дробно-стохастического программирования. Рассмотрим схемы декомпозиции по ограничениям и субградиентные методы для решения дробных задач стохастической оптимизации.

1. Двухэтапная задача стохастического программирования. Двухэтапная задача стохастического программирования имеет следующий вид [737, 738]:

$$\min_x [(c, x) + E(x)], \tag{7.36}$$

$$x \geq 0, \tag{7.37}$$

в которой: c, x – n_1-мерные вектора, $E(x)$ – математическое ожидание случайной функции $\varphi(x, \omega)$, определенной следующим образом:

$$\varphi(x, \omega) = \min_z (h, z) \tag{7.38}$$

$$z \geq 0, \tag{7.39}$$

$$Dz \geq b_\omega - Ax, \tag{7.40}$$

в которой: h, z – n_2 -мерные вектора, A, D – матрицы размерностей $(m \times n_1)$ и $(m \times n_2)$ соответственно, b_ω – случайный m-мерный вектор.

Когда случайный вектор b_ω принимает конечное число значений b_r $(r = \overline{1, k})$ с соответствующими вероятностями p_r $(r = \overline{1, k})$, для которых $\sum\limits_{r=1}^{k} p_r = 1$, стохастическая задача (7.36), (7.37) сводится к обычной задаче линейного программирования следующего вида [737, 738]:

$$\min_{x,z} \left[(c, x) + \sum_{r=1}^{k} p_r \cdot (h, z_r) \right], \tag{7.41}$$

$$\begin{array}{llllll} Ax + Dz_1 & & & \geq b_1, \\ \multicolumn{4}{l}{\dots\dots\dots\dots\dots\dots} \\ Ax + & Dz_r & & \geq b_r, \\ \multicolumn{4}{l}{\dots\dots\dots\dots\dots\dots} \\ Ax + & & Dz_k & \geq b_k, \end{array} \tag{7.42}$$

$$x \geq 0, \quad z_r \geq 0 \quad (r = \overline{1, k}). \tag{7.43}$$

Двойственная для задачи (7.41)–(7.43) имеет вид:

$$\max_u \sum_{r=1}^{k}(b_r, u_r), \tag{7.44}$$

$$A^T u_1 + A^T u_2 + \cdots A^T u_k \le c, \tag{7.45}$$

$$\begin{array}{llll} D^T u_1 & & & \le p_1 h, \\ & D^T u_2 & & \le p_2 h, \\ \multicolumn{4}{c}{\dots\dots\dots\dots\dots\dots\dots} \\ & & D^T u_k & \le p_k h \end{array} \tag{7.46}$$

$$u_r \ge 0 \quad (r = \overline{1,k}). \tag{7.47}$$

Как видно, задачи (7.41)–(7.43) и (7.44)–(7.47) имеют блочно-диагональную структуру, для решения которых можно применить схемы декомпозиции и субградиентные методы.

2. Двухэтапные задачи дробно-стохастического программирования. Одновременно с линейными функциями в двухэтапной задаче стохастического программирования (7.36)–(7.40) можно рассматривать и некоторые дробные функции. Рассмотрим следующие возможные линейные или дробные функции в двухэтапной задаче стохастического программирования:

$$F_1(x) = (c, x) + E_1(x),$$

$$F_2(x) = (c, x) + E_2(x),$$

$$F_3(x) = \frac{(c, x) + E_1(x)}{(d, x)},$$

$$F_4(x) = \frac{(c, x)}{(d, x)} + E_1(x),$$

$$F_5(x) = \frac{(c, x)}{(d, x)} + E_2(x),$$

в которых d – n_1-мерный вектор, а $E_1(x)$ и $E_2(x)$ математические ожидания случайных функций $\varphi_1(x, \omega)$ и $\varphi_2(x, \omega)$ соответственно.

В соответствии с этими функциями получаем двухэтапные задачи стохастического программирования типа (7.36)–(7.40), но с дробными функциями. В таких задачах целевые функции первого этапа соответствуют функциям $F_t(x)$ $(t = \overline{1,5})$, в результате которых получаем стохастические задачи оптимизации, которые представляют:

- задачи оптимизации некоторой линейной функции типа $F_1(x)$;
- задачи оптимизации некоторой дробно-линейной функции типа $F_3(x)$;
- задачи оптимизации некоторой функции, представленной в виде суммы одной линейной функции типа $F_2(x)$ и одной дробно-линейной функции типа $F_4(x)$;
- задачи оптимизации некоторой функции, представленной в виде суммы двух дробно-линейных функций типа $F_5(x)$.

В общем случае, двухэтапные задачи стохастического программирования для таких функций имеют следующие постановки:

- на первом этапе рассматривается одна из задач:

$$\min_x F_t(x), \quad \text{для} \quad t = 1,\, 2,\, 3,\, 4 \ \text{ или } \ 5, \tag{7.48}$$

$$x \geq 0, \tag{7.49}$$

- на втором этапе для определения математического ожидания $E_1(x)$ или $E_2(x)$ для одной из случайных функций $\varphi_1(x,\omega)$ или $\varphi_2(x,\omega)$ соответственно используется одна из следующих задач линейной или дробно-линейной оптимизации:

$$\varphi_1(x,\omega) = \min_z (h, z) \tag{7.50}$$

или

$$\varphi_2(x,\omega) = \min_z \frac{(h, z)}{(g, z)} \tag{7.51}$$

при ограничениях

$$z \geq 0, \tag{7.52}$$

$$Dz \geq b_\omega - Ax, \tag{7.53}$$

где g – n_2-мерный вектор.

Для каждой из функций $F_t(x)$ $(t = \overline{1,5})$ и соответственно одной из случайных функций $\varphi_1(x,\omega)$ или $\varphi_2(x,\omega)$, которые соответствуют математическим ожиданиям $E_1(x)$ или $E_2(x)$, получаем различные двухэтапные задачи стохастического программирования, которые содержат как линейные, так и дробно-линейные функции.

Рассмотрим в дальнейшем случаи, когда случайный вектор b_ω принимает некоторое конечное число значений b_r $(r = \overline{1,k})$ с соответствующими вероятностями p_r $(r = \overline{1,k})$, которые позволяют нам определить значения математических ожиданий $E_1(x)$ и $E_2(x)$ для случайных функций $\varphi_1(x,\omega)$ и $\varphi_2(x,\omega)$ в соответствии со следующими выражениями:

$$E_1\big(x(z)\big) = \sum_{r=1}^{k} p_r \cdot (h, z_r);$$

$$E_2\big(x(z)\big) = \sum_{r=1}^{k} p_r \frac{(h, z_r)}{(g, z_r)}.$$

В зависимости от использования одной из функций $F_t(x)$ $(t = \overline{1,5})$ и, соответственно, одной из случайных функций $\varphi_1(x,\omega)$ или $\varphi_2(x,\omega)$ получаем различные задачи стохастического программирования. Рассмотрим возможные случаи постановок двухэтапных задач стохастической оптимизации с линейными и дробными функциями.

Случай 1. В случае функции $F_1(x)$ и соответственно случайной функции $\varphi_1(x,\omega)$ получаем рассмотренную выше двухэтапную задачу стохастического линейного программирования типа (7.36)–(7.40), которую перепишем в следующем общем виде

$$\min_{x,z \in R(x,z)} \Big[(c,x) + \sum_{r=1}^{k} p_r \cdot (h, z_r)\Big], \tag{7.54}$$

в которой множество $R(x,z)$ содержит всевозможные значения переменных (x,z), которые удовлетворяют ограничениям (7.42) и (7.43).

В остальных случаях, для функций $F_t(x)$ $(t = \overline{2,5})$, и соответственно одной из случайных функций $\varphi_1(x,\omega)$ или $\varphi_2(x,\omega)$, получаем задачи дробно-стохастического программирования. Дробные функции появляются как на первом этапе, так и на втором этапе задач дробно-стохастического программирования.

Следует отметить, что появление дробных функций в задачах стохастического программирования усложняют такие задачи и не позволяют напрямую использовать процедуры решения, применяемые в случае линейной задачи (7.42). В таких случаях для решения двухэтапных задач дробно-стохастического программирования требуется модифицировать используемые данные процедуры и применять другие схемы и алгоритмы их решения. Рассмотрим структуру целевых функций для каждого случая оптимизации в отдельности.

Случай 2. Рассмотрим задачу (7.36)–(7.40), в которой на первом этапе используется линейная функция $F_2(x)$, а на втором этапе – случайная дробная функция $\varphi_2(x,\omega)$. В таком случае, задача (7.36), (7.37) с функцией $F_2(x)$ сводится к задаче оптимизации суммы линейной функции и k дробно-линейных функций следующего вида:

$$\min_{x,z\in R(x,z)} \left[(c,x) + \sum_{r=1}^{k} p_r \frac{(h,z_r)}{(g,z_r)}\right]. \tag{7.55}$$

Случай 3. Если в задаче (7.36)–(7.40) на первом этапе используется дробная функция $F_3(x)$ и соответственно случайная линейная функция $\varphi_1(x,\omega)$ на втором этапе, то получаем задачу дробно-линейного программирования следующего вида:

$$\min_{x,z\in R(x,z)} \frac{(c,x) + \sum\limits_{r=1}^{k} p_r \cdot (h,z_r)}{(d,x)}. \tag{7.56}$$

Случай 4. Рассмотрим случай, когда в задаче (7.36)–(7.40) на первом этапе используется дробная функция $F_4(x)$, а на втором этапе – случайная линейная функция $\varphi_1(x,\omega)$. Тогда получаем задачу оптимизации суммы одной дробно-линейной функции и k линейных функций, которая имеет следующий вид:

$$\min_{x,z\in R(x,z)} \left[\frac{(c,x)}{(d,x)} + \sum_{r=1}^{k} p_r \cdot (h,z_r)\right] \tag{7.57}$$

Случай 5. Для последнего случая таких двухэтапных задач стохастического программирования предположим, что в задаче (7.36)–(7.40) на каждом этапе используются дробные функции, а именно дробную функцию $F_5(x)$ на первом этапе и соответственно случайную дробную функцию $\varphi_2(x,\omega)$ – на втором этапе. Тогда получаем задачу оптимизации суммы из $k+1$ дробно-линейных функций, которая имеет следующий вид:

$$\min_{x,z\in R(x,z)} \left[\frac{(c,x)}{(d,x)} + \sum_{r=1}^{k} p_r \frac{(h,z_r)}{(g,z_r)}\right]. \tag{7.58}$$

Множество ограничений задачи (7.54)–(7.58), которые соответствуют случаях 1–5 имеют специальную структуру, а именно, для переменных z_r ограничения в них представляют собой отдельные диагональные блоки для каждого $r = \overline{1,k}$, однако они связаны между собой через переменные x. Следует отметить, что каждый блок содержит ограничения определенные относительно одной матрицы D, но с разными переменными z_r и одной матрицы A с одними и теми же переменными x. Для каждой из задач (7.54)–(7.58) структура ограничений, которые определяют множество $R(x,z)$ имеет блочно-диагональную структуру со связующими переменными:

$$\begin{array}{llll} Ax + Dz_1 & & & \geq b_1, \\ \multicolumn{4}{l}{\dots\dots\dots\dots\dots\dots} \\ Ax+ & Dz_r & & \geq b_r, \\ \multicolumn{4}{l}{\dots\dots\dots\dots\dots\dots} \\ Ax+ & & Dz_k & \geq b_k. \end{array}$$

Если рассмотрим двойственные задачи, то они имеют одинаковые ограничения блочно-диагональной структуры со связующими ограничениями, для которых правые части различаются для каждой задачи в отдельности, но с одинаковым целевым функционалом, а именно:

$$\max_u \sum_{r=1}^{k} (b_r, u_r),$$

$$A^T u_1 + A^T u_2 + \cdots A^T u_k \leq \alpha,$$

$$\begin{array}{llll} D^T u_1 & & & \leq \beta_1, \\ & D^T u_2 & & \leq \beta_2, \\ \multicolumn{4}{l}{\dots\dots\dots\dots\dots\dots\dots} \\ & & D^T u_k & \leq \beta_k, \end{array}$$

в котором параметры α и $\beta_r \quad (r = \overline{1,k})$ определяются следующим образом:

$\alpha = c$ или $\alpha = c - \lambda d$, а $\beta_r = p_r h$ или $\beta_r = p_r(h - \lambda_r g), \quad r = \overline{1,k}.$

С учетом введения новых параметров дробные задачи (7.55)–(7.58) могут быть записаны в следующие виды:

- для задачи (7.55) имеем задачу

$$\min_{x,z \in R(x,z)} \Big[(c,x) + \sum_{r=1}^{k} p_r (h - \lambda_r g, z_r),$$

- для задач (7.56) и (7.57) имеем задачу

$$\min_{x,z \in R(x,z)} \Big[(c - \lambda d, x) + \sum_{r=1}^{k} p_r (h, z_r) \Big],$$

- для задачи (7.58) имеем задачу

$$(c - \lambda d, x) + \sum_{r=1}^{k} p_r (h - \lambda_r g, z_r).$$

Для решения задач таких структур в предыдущих главах предложены использовать схемы декомпозиции по переменным или ограничениям с применением методов недифференцируемой оптимизации для задач линейного и дробного программирования. Для линейных функций в работах [737, 738] такие схемы декомпозиции по ограничениям использованы для решения двухэтапной задачи стохастического программирования (7.54).

3. Общие схемы декомпозиции по переменным. Рассмотрим в дальнейшем общие схемы декомпозиции по переменным для решения каждой из задач (7.54)–(7.58) в отдельности.

Для каждой из задач (7.54)–(7.58) построим соответствующие функции Лагранжа по одной из формул:

$$L_1(x, z, u) = (c, x) + \sum_{r=1}^{k} p_r \cdot (h, z_r) + \sum_{r=1}^{k} u_r (b_r - Ax - Dz_r),$$

$$L_2(x, z, u) = (c, x) + \sum_{r=1}^{k} p_r \frac{(h, z_r)}{(g, z_r)} + \sum_{r=1}^{k} u_r (b_r - Ax - Dz_r),$$

$$L_3(x, z, u) = \frac{(c, x) + \sum\limits_{r=1}^{k} p_r \cdot (h, z_r)}{(d, x)} + \sum_{r=1}^{k} u_r (b_r - Ax - Dz_r),$$

$$L_4(x, z, u) = \frac{(c, x)}{(d, x)} + \sum_{r=1}^{k} p_r \cdot (h, z_r) + \sum_{r=1}^{k} u_r (b_r - Ax - Dz_r),$$

$$L_5(x, z, u) = \frac{(c, x)}{(d, x)} + \sum_{r=1}^{k} p_r \frac{(h, z_r)}{(g, z_r)} + \sum_{r=1}^{k} u_r (b_r - Ax - Dz_r),$$

в которых u_r представляют собой множители Лагранжа для каждой группы ограничений (7.42) в отдельности $r = \overline{1, k}$.

Если для решения задачи стохастического программирования используем схему декомпозиции по переменным, тогда при фиксированных значениях переменные $x = \overline{x}$ для задач (7.54)–(7.58) получаем следующие недифференцируемые функции:

$$\Psi_1(\overline{x}) = \min_{z \in R(z(x))} \left[(c, \overline{x}) + \sum_{r=1}^{k} p_r \cdot (h, z) \right], \tag{7.59}$$

$$\Psi_2(\overline{x}) = \min_{z \in R(z(x))} \left[(c, \overline{x}) + \sum_{r=1}^{k} p_r \frac{(h, z)}{(g, z)} \right], \tag{7.60}$$

$$\Psi_3(\overline{x}) = \min_{z \in R(z(x))} \left[\frac{(c, \overline{x}) + \sum\limits_{r=1}^{k} p_r \cdot (h, z)}{(d, \overline{x})} \right], \tag{7.61}$$

$$\Psi_4(\overline{x}) = \min_{z \in R(z(x))} \left[\frac{(c, \overline{x})}{(d, \overline{x})} + \sum_{r=1}^{k} p_r \cdot (h, z) \right], \tag{7.62}$$

$$\Psi_5(\overline{x}) = \min_{z \in R(z(x))} \left[\frac{(c, \overline{x})}{(d, \overline{x})} + \sum_{r=1}^{k} p_r \frac{(h, z)}{(g, z)} \right], \tag{7.63}$$

в которых множество $R(z(x))$ представляет собой все возможные значения переменных z, удовлетворяющие ограничениям (7.42) и (7.43) при $x = \overline{x}$.

Тогда задачи недифференцируемой оптимизации имеют вид:

$$\min_{x \geq 0} \Psi_t(\overline{x}) \quad (t = \overline{1, 5}). \tag{7.64}$$

Функции $\Psi_t(\overline{x}) \quad (t = \overline{1, 5})$ в задачах (7.64) являются недифференцируемыми и для их решения используются методы субградиентного типа.

Значения субградиентов для каждой функции $\Psi_t(\overline{x}) \quad (t = \overline{1, 5})$ в точке $x = \overline{x}$ определяются соответственно по одной из следующих формул:

$$G^1_{L_1}(\overline{x}) = \left\{ c - \sum_{r=1}^{k} u_r(\overline{x}) A \right\}, \tag{7.65}$$

$$G^2_{L_2}(\overline{x}) = \left\{ c - \sum_{r=1}^{k} u_r(\overline{x}) A \right\}, \tag{7.66}$$

$$G^3_{L_3}(\overline{x}) = \left\{ \frac{c \cdot (d, \overline{x}) - d \cdot \left[(c, \overline{x}) + \sum\limits_{r=1}^{k} p_r \cdot (h, z_r) \right]}{[(d, \overline{x})]^2} - \sum_{r=1}^{k} u_r(\overline{x}) A \right\}, \quad (7.67)$$

$$G^4_{L_4}(\overline{x}) = \left\{ \frac{c \cdot (d, \overline{x}) - d \cdot (c, \overline{x})}{[d, \overline{x})]^2} - \sum_{r=1}^{k} u_r(\overline{x}) A \right\}, \quad (7.68)$$

$$G^5_{L_5}(\overline{x}) = \left\{ \frac{c \cdot (d, \overline{x}) - d \cdot (c, \overline{x})}{[d, \overline{x})]^2} - \sum_{r=1}^{k} u_r(\overline{x}) A \right\}. \quad (7.69)$$

Для решения задач (7.64) используем один из методов субградиентного типа минимизации одной из недифференцируемой функции конкретного типа $\Psi_t(\overline{x}) \quad (t = \overline{1,5})$. Пусть выполнено s шагов субградиентного метода и получены значения x^s переменных оптимизации x для соответствующей функции $\Psi_t(\overline{x})$, тогда на $(s+1)$-м шаге выполняем следующие вычисления.

I. Решаем одну из задач оптимизации (7.59)–(7.63), которая соответствует функции $\Psi_t(\overline{x})$ с фиксированными значениями переменных $x = \overline{x} = x^s$. Для каждой из них получаем следующие задачи оптимизации по переменным z.

Для случая 1 получаем задачу линейного программирования:

$$\sum_{r=1}^{k} p_r \cdot (h, z_r) \to \min, \quad (7.70)$$

$$\begin{aligned} Dz1 \qquad &\geq b_1 - Ax^s, \\ \dots\dots\dots\dots& \\ Dz_r \qquad &\geq b_r - Ax^s, \quad (7.71) \\ \dots\dots\dots\dots& \\ Dz_k &\geq b_k - Ax^s. \end{aligned}$$

$$z_r \geq 0 \quad (r = \overline{1,k}). \quad (7.72)$$

Задача (7.70)–(7.72) имеет блочно-диагональную структуру и она распадается на k линейных задач для каждого блока в отдельности, а именно для каждого $r = \overline{1,k}$ имеем прямую задачу

$$p_r \cdot (h, z_r) \to \min, \tag{7.73}$$

$$Dz_r \geq b_r - Ax^s, \tag{7.74}$$

$$z_r \geq 0 \tag{7.75}$$

и соответствующую ей двойственную задачу

$$(b_r - Ax^s,\, u_r) \to \max, \tag{7.76}$$

$$D^T u_r \leq p_r h, \tag{7.77}$$

$$u_r \geq 0. \tag{7.78}$$

Находим оптимальное решение $z_r^*(x^s)$ линейной задачи (7.73)–(7.75) и соответствующее ей двойственное решение $u_r^*(x^s)$ (множители Лагранжа) задачи (7.76)–(7.78) для каждой группы ограничений в отдельности $r = \overline{1, k}$.

Для случая 2 получаем задачу дробно-линейного программирования:

$$\sum_{r=1}^{k} p_r \frac{(h, z_r)}{(g, z_r)} \to \min \tag{7.79}$$

при ограничениях (7.71), (7.72).

Так как функционал (7.79) сепарабельный, то задача (7.79), (7.71), (7.72) распадается на отдельные блоки для каждого $r = \overline{1, k}$ и получаем k задач дробно-линейного программирования вида:

$$\frac{(h, z_r)}{(g, z_r)} \to \min, \tag{7.80}$$

$$Dz_r \geq b_r - Ax^s, \tag{7.81}$$

$$z_r \geq 0. \tag{7.82}$$

Для решения таких задач применяем один из приведенных в предыдущих главах методов решения задач дробного программирования. Если использовать параметрический метод, то при фиксированном параметре λ_r решаем прямые и двойственные задачи линейной оптимизации для каждого $r = \overline{1, k}$ в отдельности:

- прямая задача

$$(h - \lambda_r g,\, z_r) \to \min, \tag{7.83}$$

$$Dz_r \geq b_r - Ax^s, \tag{7.84}$$

$$z_r \geq 0; \tag{7.85}$$

- двойственная задача

$$(b_r - Ax^s,\, u_r) \to \max, \tag{7.86}$$

$$D^T u_r \le h - \lambda_r g, \tag{7.87}$$

$$u_r \ge 0. \tag{7.88}$$

Определяем значение параметра $\lambda_r = \lambda_r^*$ задачи параметрического программирования, в которой $z_r^*(x^s)$ является решением линейной задачи (7.83)–(7.85), для которого находим оптимальное решение $z^* = \{z_r^*(x^s) \quad (r = \overline{1,k})\}$ дробно-линейной задаче (7.79), (7.71), (7.72), а $u_r^*(x^s)$ – оптимальное решение двойственной задаче (7.86)–(7.88) для каждой группы ограничений в отдельности $r = \overline{1,k}$. В результате получаем значения параметров $\{\lambda_r^*\}$, для которых z_r^* является оптимальным решением дробно-линейной задачи (7.79), (7.71), (7.72).

Для случая 3 получаем задачу линейного программирования:

$$\frac{1}{(d, x^s)} \sum_{r=1}^{k} p_r \cdot (h, z_r) \to \min \tag{7.89}$$

при ограничениях (7.71), (7.72).

Линейная задача (7.89), (7.71), (7.72) имеет блочно-диагональную структуру и, как и для случая 1, распадается на k прямых и двойственных линейных задач для каждого блока в отдельности, для которых определяются оптимальные решения $z_r^*(x^s)$ и $u_r^*(x^s) \quad (r = \overline{1,k})$.

Для случая 4 получаем такие же задачи (прямую и двойственную) линейного программирования как и для случая 1 и 3, которые распадаются по каждому блоку. В результате для каждого блока в отдельности определяются оптимальные решения $z_r^*(x^s)$ и $u_r^*(x^s) \quad (r = \overline{1,k})$.

Для случая 5 получаем аналогичную задачу дробно-линейного программирования как и для случая 2. Находим значение параметра $\lambda_r = \lambda_r^*$ задачи параметрического программирования (7.83)–(7.85), для которого получаем оптимальное решение $z^* = \{z_r^*(x^s) \quad (r = \overline{1,k})\}$ задачи дробно-линейного программирования (7.79), (7.71), (7.72), где $z_r^*(x^s)$ решение прямой линейной задачи (7.83)–(7.85), а $u_r^*(x^s)$ – оптимальное решение двойственной задачи (7.86)–(7.88) для каждой группе ограничений в отдельности $r = \overline{1,k}$.

II. Определяем в точке $x = \overline{x} = x^s$ значения одного из субградиентов $G_{L_t}^t(\overline{x}) \quad (t = \overline{1,5})$, которые соответствуют функциям оптимизации $\Psi_t(\overline{x}) \quad (t = \overline{1,5})$ по одной из формул (7.65)–(7.69), в которых используются решения $z^* = \{z_r^*(x^s) \quad (r = \overline{1,k})\}$ и $u^* = \{u_r^*(x^s) \quad (r = \overline{1,k})\}$, полученные в результате решения соответствующей задачи (7.60).

III. Имея значения одного из субградиентов $G^t_{L_t}(\overline{x}) \quad (t = \overline{1,5})$, выполняем соответствующие вычисления при переходе к другой точке x^{s+1} по формуле

$$x^{s+1} = \max\{0,\, x^s - \mu_{s+1} G^t_{L_t}(\overline{x})\},$$

где μ_{s+1} величина шага в субградиентном методе.

В результате получаем, что точка $x = x^*$, которое дает $\Psi_t(x^*) = \min\limits_{x \geq 0} \Psi_t(\overline{x})$, оптимальное решение задачи первого этапа двухэтапной задачи стохастического программирования, а значения $z = z^*$, которые соответствуют оптимальным решениям линейных задач (7.83)–(7.85) для $x = x^s$, представляют собой мультипликатором оптимальных решений второго этапа при различных значениях $b_r \quad (r = \overline{1,k})$.

Таким образом, из изложенного видно, что схема решения задач (7.64), или первоначальной двухэтапной задачи дробно-стохастического программирования (7.36), (7.37) является одинаковой для каждой из функций $F_t(x) \quad (t = \overline{1,5})$. На каждой итерации субградиентного метода решаются одни и те же прямые (7.83)–(7.85) или двойственные (7.86)–(7.88) задачи линейного программирования. Данные схемы различаются только формулами вычисления субградиентов функций $\Psi_t(\overline{x}) \quad (t = \overline{1,5})$.

Такая схема решения линейных или дробных двухэтапных задач стохастического программирования требует для ее реализации на ЭВМ наличия программного обеспечения для недифференцируемой оптимизации и решения задач линейного программирования.

7.4. Задачи дробно-линейного программирования с переменными коэффициентами

Рассмотрим задачу дробно-линейного программирования, в которой неизвестны не только переменные x_j, но и коэффициенты столбцов a_j матрицы ограничений, коэффициенты c_j числителя и d_j знаменателя дробно-линейного функционала. Таким образом, сформулируем следующую задачу дробно-линейного программирования: найти неотрицательные значения переменных x_j и значения векторов $(a_j,\, c_j,\, d_j)$, обеспечивающие минимум целевой функции

$$F(A,c,d,x) = \frac{F_1(c,x)}{F_2(d,x)} = \frac{\sum\limits_{j=1}^{n} c_j x_j}{\sum\limits_{j=1}^{n} d_j x_j} \to \min \tag{7.90}$$

при ограничениях

$$\sum_{j=1}^{n} a_j x_j \leq b, \tag{7.91}$$

$$x_j \geq 0 \quad (j = \overline{1,n}), \tag{7.92}$$

$$(a_j,\, c_j,\, d_j) \in S_j \quad (j = \overline{1,n}), \tag{7.93}$$

где a_j, b – векторы размерности m; $A = (a_1, a_2, \ldots \ldots, a_n)$; $c = (c_1, c_2, \ldots, c_n)$; $d = (d_1, d_2, \ldots, d_n)$; $x = (x_1, x_2, \ldots, x_n)$; S_j предполагаются выпуклыми ограниченными многогранными множествами в $(m+2)$-мерном пространстве.

Задача (7.90)–(7.93) является задачей нелинейного дробного программирования. В общем случае она является многоэкстремальной и для ее решения могут быть использованы общие методы математического программирования. Однако для решения задачи (7.90)–(7.93) могут быть разработаны декомпозиционные алгоритмы, основанные на принципе разложения Данцига-Вулфа и схемах декомпозиции по переменным и ограничениям.

1. Декомпозиционный алгоритм Данцига-Вулфа. Рассмотрим для начала декомпозиционный алгоритм решения задачи (7.90)–(7.93), основанный на принципе разложения Данцига-Вулфа. Так как множество S_j является выпуклым и ограниченным многогранником, то любой вектор $(a_j, c_j, d_j) \in S_j$ может быть представлен в виде линейной выпуклой комбинации крайних точек многогранника S_j.

Пусть (a_j^k, c_j^k, d_j^k) $(k = \overline{1, r_j})$ – крайние точки многогранника S_j. Тогда

$$(a_j, c_j, d_j) = \sum_{k=1}^{r_j} \lambda_j^k (a_j^k, c_j^k, d_j^k) \quad (j = \overline{1,n}), \tag{7.94}$$

$$\sum_{k=1}^{r_j} \lambda_j^k = 1 \quad (j = \overline{1,n}), \quad \lambda_j^k \geq 0 \quad (j = \overline{1,n};\ k = \overline{1, r_j}).$$

Подставив (7.94) в (7.90) и (7.91) получим задачу:

$$\frac{\sum\limits_{j=1}^{n}\sum\limits_{k=1}^{r_j}\lambda_j^k c_j^k x_j}{\sum\limits_{j=1}^{n}\sum\limits_{k=1}^{r_j}\lambda_j^k d_j^k x_j} \to \min,$$

$$\sum_{j=1}^{n}\sum_{k=1}^{r_j}\lambda_j^k a_j^k x_j \le b,$$

$$\sum_{k=1}^{r_j}\lambda_j^k = 1 \quad (j=\overline{1,n}),$$

$$x_j \ge 0 \quad (j=\overline{1,n}),$$

$$\lambda_j^k \ge 0 \quad (j=\overline{1,n};\ k=\overline{1,r_j}).$$

Введем новые переменные:

$$y_j^k = \lambda_j^k x_j \quad (j=\overline{1,n};\ k=\overline{1,r_j}). \tag{7.95}$$

Тогда эту задачу можно записать в следующем виде:

$$\frac{F_1(y)}{F_2(y)} = \frac{\sum\limits_{j=1}^{n}\sum\limits_{k=1}^{r_j} c_j^k y_j^k}{\sum\limits_{j=1}^{n}\sum\limits_{k=1}^{r_j} d_j^k y_j^k} \to \min, \tag{7.96}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{r_j} a_j^k y_j^k \le b, \tag{7.97}$$

$$y_j^k \ge 0 \quad (j=\overline{1,n};\ k=\overline{1,r_j}). \tag{7.98}$$

Полученная задача является задачей дробно-линейного программирования и для ее решения используем модифицированный симплекс-метод. Следует заметить, что при решении задачи (7.96)–(7.98) нет необходимости заранее знать все крайние точки многогранников S_j.

Пусть B – текущий базис задачи (7.96)–(7.98), u и v – векторы двойственных переменных соответственно для числителя и знаменателя функционала (7.96), которые определяются по формулам

$$u = c_{_B} B^{-1}, \quad v = d_{_B} B^{-1},$$

а $F_1(y)$ и $F_2(y)$ – значения числителя и знаменателя в текущем базисе. Тогда признаком оптимальности текущего базиса задачи (7.96)–(7.98) является выражение

$$\Delta_j^k = F_2(y)(c_j^k - ua_j^k) - F_1(y)(d_j^k - va_j^k) \geq 0,$$

$$j = \overline{1, n};\ k = \overline{1, r_j}.$$

В базис вводится вектор с наименьшим значением выражения Δ_j^k. Для этого решаем n задач линейного программирования:

$$\big(F_1(y)v - F_2(y)u\big)a_j - F_2(y)c_j - F_1(y)d_j \to \min, \tag{7.99}$$

$$(a_j,\, c_j,\, d_j) \in S_j. \tag{7.100}$$

Пусть решение j-й задачи (7.99), (7.100) $(a_j^{t_j},\, c_j^{t_j},\, d_j^{t_j})$ достигается в t_j-й крайней точке многогранника S_j и пусть $\Delta_j^{t_j}$ – оптимальное значение функционала (7.99). Тогда находим

$$\Delta_l^{t_l} = \min_j \Delta_j^{t_j}.$$

Если $\Delta_l^{t_l} < 0$, то в базис вводится вектор $(a_l,\, c_l,\, d_l)$ с координатами $(a_l^{t_l},\, c_l^{t_l},\, d_l^{t_l})$.

Если $\Delta_l^{t_l} \geq 0$, то текущий базис является оптимальным.

Пусть $\{\overline{y}_j^k\}$ – оптимальное решение задачи (7.96)–(7.98). Тогда учитывая (7.94) и (7.95), находим оптимальное решение задачи (7.90)–(7.93) по формулам

$$\overline{x}_j = \sum_{k=1}^{r_j} \overline{y}_j^k \quad (j = \overline{1, n}),$$

$$(\overline{a}_j,\, \overline{c}_j,\, \overline{d}_j) = \sum_{k=1}^{r_j} \overline{\lambda}_j^k (a_j^k,\, c_j^k,\, d_j^k) \quad (j = \overline{1, n}),$$

$$\overline{\lambda}_j^k = \overline{y}_j^k / \overline{x}_j \quad (j = \overline{1, n};\ k = \overline{1, r_j}).$$

Таким образом задача (7.90)–(7.93) сводится к решению задачи дробно-линейного программирования (7.96)–(7.98). На каждой итерации модифицированного симплекс-метода ее решения необходимо решать n задач линейного программирования на выпуклых множествах

S_j. В общей сложности метод решения задачи (7.90)–(7.93), основанный на принципе разложения Данцига-Вулфа, представляет собой двухуровневый алгоритм. На первом уровне решается координирующая задача дробно-линейного программирования (7.96)–(7.98), а на втором – n задач линейного программирования (7.99), (7.100).

2. Двухуровневый алгоритм декомпозиции по переменным и ограничениям. Рассмотрим теперь двухуровневый алгоритм решения задачи (7.90)–(7.93), основанный на схемах декомпозиции по переменным и ограничениям с использованием субградиентных методов. Предположим, что $a_j \geq 0 \quad (j = \overline{1,n})$. Тогда для задачи (7.90)–(7.93) можно применить теорию Куна-Таккера и множителей Лагранжа, тем самым использовать схемы декомпозиции по переменным и ограничениям и свести ее к нахождению седловой точки функции Лагранжа субградиентными методами.

Так как переменные x_j входят только в ограничения (7.91), то целесообразно первоначально применить схему декомпозиции по переменным. Тогда при фиксированных значениях переменных x_j получим задачу минимизации дробно-линейного функционала на множестве переменных $(a_j,\, c_j,\, d_j)$, удовлетворяющих ограничениям (7.91) и блочно-диагональным ограничениям (7.93). Для решения такой задачи применяем схему декомпозиции по ограничениям. Используя параметрический метод, полученную задачу дробно-линейного программирования определения значений обобщенного градиента решаем для каждого блока в отдельности.

На внешнем уровне итеративного алгоритма решения задачи (7.90)–(7.93) применяем схему декомпозиции по переменным, для чего фиксируем переменные $x_j = \overline{x}_j$. Тогда получим задачу минимизации квазивыпуклой функции

$$\Phi(x) = \min_{(A,c,d)\in R(x)} F(A,c,d,x)$$

на множестве переменных $x \geq 0$, где $R(x)$ – множество значений переменных (A, c, d), удовлетворяющих ограничениям (7.91) при фиксированных $x = \overline{x}$. Для минимизации функции $\Phi(x)$ используем субградиентные методы, на t-м шаге которых для достигнутой точке $x = x^t$ необходимо выполнить следующие три основных этапа.

1. При фиксированных $x_j = x_j^t$ решить задачу дробно-линейного программирования

$$\frac{F_1(c, x^t)}{F_2(d, x^t)} = \frac{\sum\limits_{j=1}^{n} x_j^t c_j}{\sum\limits_{j=1}^{n} x_j^t d_j} \to \min, \tag{7.101}$$

$$\sum_{j=1}^{n} x_j^t a_j \le b, \tag{7.102}$$

$$(a_j,\, c_j,\, d_j) \in S_j \quad (j = \overline{1, n}) \tag{7.103}$$

и найти оптимальное решение $\{a_j^*(x^t),\, c_j^*(x^t),\, d_j^*(x^t)\}$, значения числителя $F_1(c, x^t)$ и знаменателя $F_2(d, x^t)$, а также двойственные оценки $u^*(x^t)$ ограничениям (7.102).

2. Определить значения обобщенных градиентов функции $\Phi(x)$ в точке $x = x^t$ по формуле

$$g_j^\Phi(x^t) = \bigl(c_j^*(x^t) F_2(d, x^t) - d_j^*(x^t) F_1(c, x^t)\bigr) / [F_2(d, x^t)]^2 +$$
$$+ \bigl(u^*(x^t),\, a_j^*(x^t)\bigr) \quad (j = \overline{1, n}).$$

3. Найти новые значения

$$x_j^{t+1} = \max\bigl\{0,\, x_j^t - h_{t+1} g_j^\Phi(x^t)\bigr\} \quad (j = \overline{1, n}),$$

где h_{t+1} – величина шага.

Для решения задачи (7.101)–(7.103) применяем схему декомпозиции по ограничениям. Построим функцию Лагранжа на множестве ограничений (7.102) и рассмотрим задачу

$$L^*(u) = \min_{(A,c,d) \in S(x)} \frac{\sum\limits_{j=1}^{n} \overline{x}_j c_j + \Bigl(u,\, \sum\limits_{j=1}^{n} \overline{x}_j a_j - b\Bigr)}{\sum\limits_{j=1}^{n} \overline{x}_j d_j}, \tag{7.104}$$

где $\overline{x}_j = x_j^t \;\; (j = \overline{1, n})$, $u = (u_1, u_2, \ldots, u_m)$ – множители Лагранжа для ограничений (7.102), а $S(x) = S_1 \times S_2 \times \cdots \times S_n$.

На втором уровне итеративного алгоритма решаем задачу безусловной оптимизации функции $L^*(u)$, которая является вогнутой и кусочно-линейной на множестве переменных $u \ge 0$. Тогда на r-м шаге субградиентного метода, в сочетании с параметрическим методом решения задачи дробно-линейного программирования, необходимо выполнить следующие четыре основных этапа.

1. Находим значение параметра λ^r по формуле

$$\lambda^r = \frac{\sum\limits_{j=1}^{n} x_j^t c_j^*(u^{r-1}) + \left(u^{r-1}, \sum\limits_{j=1}^{n} x_j^t a_j^*(u^{r-1}) - b\right)}{\sum\limits_{j=1}^{n} x_j^t d_j^*(u^{r-1})},$$

где $\{a_j^*(u^{r-1}),\ c_j^*(u^{r-1}),\ d_j^*(u^{r-1})\}$ решение задачи (7.104), найденное на предыдущем шаге субградиентного метода.

2. Решить задачу дробно-линейного программирования (7.104) параметрическим методом при фиксированном $\lambda = \lambda^r$. Тогда для каждого блока решаем задачу линейного программирования

$$\min_{(a_j, c_j, d_j) \in S_j} \big(c_j + (u^r,\, a_j) - \lambda^r d_j\big), \tag{7.105}$$

для которой находим решение $(a_j^*(u^r),\ c_j^*(u^r),\ d_j^*(u^r))$.

3. Определить значения обобщенного градиента функции $L^*(u)$ в точке $u = u^r$ по формуле

$$G(u^r) = \left\{ \left(\sum_{j=1}^{n} x_j^t a_j^*(u^r) - b \right) \Big/ \sum_{j=1}^{n} x_j^t d_j^*(u^r) \right\},$$

где $\big(a_j^*(u^r), c_j^*(u^r), d_j^*(u^r)\big)$, $\quad j = \overline{1, n}$ – оптимальные решения задач (7.105) при фиксированных $u = u^r$ и $\lambda = \lambda^r$.

4. Найти новые значения

$$u^{r+1} = \max\big\{0,\ u^r + h_{r+1} G(u^r)\big\},$$

где h_{r+1} – величина шага.

Таким образом, решить задачу (7.90)–(7.93) можно с помощью двухуровнего итеративного алгоритма, на внешнем уровне которого решается задача минимизации функции $\Phi(x)$, а на внутреннем уровне – задача максимизации функции $L^*(u)$. Для решения этих задач используются субградиентные методы, что обеспечивает сходимость итеративного алгоритма. Эффективность решения задачи (7.90)–(7.93) этим алгоритмом достигается за счет решения задач (7.105) для каждого блока в отдельности.

3. Задача дробно-линейного программирования с нелинейными коэффициентами. Рассмотрим задачу дробно-линейного программирования, в которой коэффициенты c_j, d_j и a_j являются нелинейными функциями от некоторой группы переменных $y = (y_1, y_2, \ldots, y_k) \in$ $\in R^k$. Эта задача имеет вид

$$\frac{F_1(x,y)}{F_2(x,y)} = \frac{\sum\limits_{j=1}^{n} c_j(y)x_j}{\sum\limits_{j=1}^{n} d_j(y)x_j} \to \min, \tag{7.106}$$

$$\sum_{j=1}^{n} a_j(y)x_j \le b, \tag{7.107}$$

$$x_j \ge 0, \tag{7.108}$$

$$y \in Y \subseteq R^k. \tag{7.109}$$

Предположим, что $c_j(y) \ge 0 \quad (j = \overline{1,n}), \quad d_j(y) > 0 \quad (j = \overline{1,n})$ и вектор-функции $a_j(y) \quad (j = \overline{1,n})$ определены для любых значений переменных $y \in Y$, где $Y \subseteq R^k$ – заданный компакт.

В общем случае задача (7.106)–(7.109) является многоэкстремальной, трудно разрешимой, и в зависимости от сложности функций c_j, d_j и a_j применяются различные методы ее решения. Мы рассмотрим подход решения задачи (7.106)–(7.109), основанный на схеме декомпозиции по переменным.

Зафиксируем переменные $y = \overline{y} \in Y$ и рассмотрим задачу дробно-линейного программирования

$$\frac{F_1(x,\overline{y})}{F_2(x,\overline{y})} = \frac{\sum\limits_{j=1}^{n} c_j(\overline{y})x_j}{\sum\limits_{j=1}^{n} d_j(\overline{y})x_j} \to \min, \tag{7.110}$$

$$\sum_{j=1}^{n} a_j(\overline{y})x_j \le b, \tag{7.111}$$

$$x_j \ge 0. \tag{7.112}$$

Определим функцию

$$\Phi(\overline{y}) = \min_{x \in S(\overline{y})} \frac{F_1(x,\overline{y})}{F_2(x,\overline{y})}, \tag{7.113}$$

где $S(\overline{y})$ – множество допустимых решений задачи (7.110)–(7.112) при фиксированных значениях переменных $\overline{y} \in Y$. При сделанных предположениях функция $\Phi(y)$ является квазивыпуклой и рассматривается задача

$$\min_{y \in Y} \Phi(y), \tag{7.114}$$

в которой функция $\Phi(y)$ определяется по формуле (7.113). Функция $\Phi(y)$ является недифференцируемой и для ее минимизации используется метод обобщенного градиентного спуска.

Если для задачи (7.106)–(7.109) построить функцию Лагранжа по формуле

$$L(x, y, u) = \frac{\sum\limits_{j=1}^{n} c_j(y)x_j + \Big(u, \sum\limits_{j=1}^{n} a_j(y)x_j - b\Big)}{\sum\limits_{j=1}^{n} d_j(y)x_j},$$

где u – вектор множителей Лагранжа ограничений (7.107). Тогда обобщенный градиент функции $\Phi(\overline{y})$ в точке $y = \overline{y} \in Y$ определяется по формуле

$$g_\Phi(\overline{y}) = g_L^y\big(\overline{y},\, x(\overline{y})\big), \tag{7.115}$$

где $x(\overline{y})$ – решение задачи (7.110)–(7.112) в точке $y = \overline{y}$; $g_L^y\big(\overline{y},\, x(\overline{y})\big)$ – проекция такого обобщенного градиента функции $L(z, u) = L(x, y, u)$ на пространство R_y^k, у которого проекция на подпространство R_x^n равна нулю (обобщенный градиент берется в точке $z = \big(\overline{y},\, x(\overline{y})\big)$).

Таким образом решение задачи (7.106)–(7.109) сводится к решению задачи (7.114) одним из методов обобщенного градиента. Тогда на t-м шаге алгоритма метода обобщенного градиентного спуска необходимо выполнить следующие три основных этапа.

1. Решить задачу дробно-линейного программирования (7.110)–(7.112) при фиксированных $y = y^t$ и найти оптимальное решение $x^*(y^t)$ и двойственные оценки $u^*(y^t)$.

2. Найти значения обобщенного градиента функции $\Phi(y)$ в точке $y = y^t$ по формуле (7.115).

3. Найти новые значения переменных y^{t+1} в соответствии с выбранным методом обобщенного градиента и структуры ограничений, задающие компакт Y.

В общей сложности, алгоритм решения задачи (7.106)–(7.109) сводится к задаче недифференцируемой оптимизации (7.114) одним из субградиентных методов и решением на каждом его шаге задачи дробно-линейного программирования для определения значений субградиента функции $\Phi(y)$.

Замечание. *Если для решения задачи дробно-линейного программирования* (7.110)–(7.112) *применить параметрический метод, то при этом достаточно выполнить только одну его итерацию.*

Для этого фиксируем значение параметра

$$\lambda = \lambda^{t+1} = F\big(x^*(y^t, \lambda^t), y^t\big)$$

и решаем задачу линейного программирования

$$\min_{x \in S(y^{t+1})} \big(F_1(x, y^{t+1}) - \lambda^{t+1} F_2(x, y^{t+1})\big).$$

Полученное решение $x^*(y^{t+1}, \lambda^{t+1})$ *и двойственные оценки* $u^*(y^{t+1}, \lambda^{t+1})$ *используются для вычисления значений обобщенного градиента по формуле* (7.115)*, полученной в соответствии с функцией Лагранжа* $L(x, y, u)$.

Глава 8

Задачи дробно-линейного программирования транспортного типа

Большинство практических задач оптимизации являются задачами производственно-транспортного типа. Именно для решения таких задач первоначально была опробована эффективность методов недифференцируемой оптимизации. Одновременно с линейными транспортными задачами на практике встречаются различные постановки таких задач с дробно-линейными функционалами.

Если задачу дробно-линейного программирования можно свести к линейной задачи, то в случае транспортной задачи, такое сведение выводит нас из этого класса оптимизационных задач. Поэтому для решения дробно-линейных транспортных задач необходимо использовать другие алгоритмы и методы, которые позволяют учитывать специфику транспортных ограничений.

В данной главе рассматриваются три группы транспортных задач с дробно-линейными функционалами: обычные транспортные задачи, производственно-транспортные задачи, обобщенные транспортные задачи.

Для их решения приводятся алгоритмы метода потенциалов, покоординатного спуска, штрафных функций, вектора спада, параметрического метода, декомпозиции Данцига-Вулфа, а также схемы декомпозиции по ограничениям и переменным, основанные на методах недифференцируемой оптимизации.

8.1. Дробно-линейные транспортные задачи

Пусть задана дробно-линейная транспортная задача

$$F(x) = \frac{C(x)}{D(x)} = \frac{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} c_{ij}x_{ij}}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} d_{ij}x_{ij}} \to \min, \tag{8.1}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}), \tag{8.2}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,n}), \tag{8.3}$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}), \tag{8.4}$$

где m и n – соответственно количество поставщиков и потребителей; a_i и b_j – соответственно объемы поставки i-го поставщика и объемы потребления j-го потребителя; x_{ij} – объем перевозки однородного груза от i-го поставщика j-му потребителю; c_{ij} – характеристика транспортного процесса, имеющая "отрицательное" влияние на работу транспорта, при перевозке грузов от i-го поставщика j-му потребителю (затраты, расстояние, время и т.п.); d_{ij} – характеристика транспортного процесса, имеющая "положительное" влияние на работу транспорта, при перевозке грузов от i-го поставщика j-му потребителю (доходы, производительность, транспортная работа и т.п.).

Предположим, что $D(x) > 0$ для любых допустимых планов задачи (8.1)–(8.4), а также имеем равенство (баланс между объемами поставки и потребления)

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j.$$

Рассмотрим двойственную к ней задачу

$$\frac{\sum\limits_{i=1}^{m} a_i u_i^1 - \sum\limits_{j=1}^{n} b_j v_j^1}{\sum\limits_{i=1}^{m} a_i u_i^2 - \sum\limits_{j=1}^{n} b_j v_j^2} \to \max, \tag{8.5}$$

$$
\begin{gathered}
(c_{ij} - u_i^1 + v_j^1)\left(\sum_{i=1}^{m} a_i u_i^2 - \sum_{j=1}^{n} b_j v_j^2\right) - \\
-(d_{ij} - u_i^2 + v_j^2)\left(\sum_{i=1}^{m} a_i u_i^1 - \sum_{j=1}^{n} b_j v_j^1\right) \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}),
\end{gathered} \tag{8.6}
$$

$$
\sum_{i=1}^{m} a_i u_i^2 - \sum_{j=1}^{n} b_j v_j^2 \geq 0. \tag{8.7}
$$

Для задач (8.1)–(8.4) и (8.5)–(8.7) справедливы основные теоремы двойственности [601, 740, 741].

Ниже рассмотрим алгоритмы решения дробно-линейных транспортных задач.

1. Алгоритм метода потенциалов. Применим метод потенциалов для решения задачи (8.1)–(8.4). По своей структуре алгоритм метода потенциалов решения дробно-линейной транспортной задачи соответствует аналогичному алгоритму линейной задачи. Отличие состоит в критерии оптимальности и в системе потенциалов. Для дробно-линейной задачи выделяются две группы потенциалов, одна – для числителя, другая – для знаменателя.

Тогда алгоритм метода потенциалов для решения дробно-линейной транспортной задачи заключается в следующем.

1. Находим первоначальный опорный план задачи (8.1)–(8.4) методом "северо-западного угла" или минимального отношения c_{ij}/d_{ij}, если все $d_{ij} \neq 0$.

2. Определяем значения потенциалов u^1, u^2, v^1 и v^2, решая следующие системы уравнений

$$
\{u_i^1 - v_j^1 = c_{ij}\} \text{ и } \{u_i^2 - v_j^2 = d_{ij}\}
$$

для всех i и j, для которых $x_{ij} \neq 0$.

3. Определяем значения числителя $C(x)$ и знаменателя $D(x)$ функционала (8.1).

4. Находим $\Delta_{ij}^1 = c_{ij} - u_i^1 + v_j^1$ и $\Delta_{ij}^2 = d_{ij} - u_i^2 + v_j^2$ для всех i, j, для которых $x_{ij} \neq 0$.

5. Вычисляем $\Delta_{ij} = D(x)\Delta_{ij}^1 - C(x)\Delta_{ij}^2$ и находим $\Delta_{rt} = \min\limits_{ij} \Delta_{ij}$.

6. Если $\Delta_{rt} \geq 0$, то план x является оптимальным для задачи (8.1)–(8.4).

7. Если $\Delta_{rt} < 0$, то переходим к новому опорному плану x' задачи (8.1)–(8.4) по известным правилам линейного случая транспортной задачи и возвращаемся ко второму шагу алгоритма.

Полученное решение x по данному алгоритму является оптимальным для исходной дробно-линейной транспортной задачи (8.1)–(8.4), а соответствующие потенциалы u^1, u^2, v^1 и v^2 – решением двойственной дробно-линейной задачи (8.5)–(8.7).

2. Алгоритм параметрического метода. Применим параметрический метод для решения дробно-линейной транспортной задачи. Обозначим через $M(x)$ множество значений переменных x, удовлетворяющих ограничениям (8.2)–(8.4), и рассмотрим следующую задачу параметрического программирования транспортного типа:

$$\min_{x \in M(x)} \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} - \lambda d_{ij}) x_{ij}. \tag{8.8}$$

Как и для задачи дробно-линейного программирования, алгоритм параметрического метода решения задачи (8.1)–(8.4) заключается в следующем.

0-й шаг. Берем $\lambda = \mu_0 = 0$ и находим оптимальное решение x_0^* задачи (8.8), которая является линейной транспортной задачи.

k-й шаг. Определяем значение параметра $\mu_k = F(x_{k-1}^*)$ и находим оптимальное решение x_k^* линейной транспортной задачи (8.8) при $\lambda = \mu_k$. Если $F(x_k^*) = F(x_{k-1}^*)$, то план x_k^* является оптимальным решением исходной дробно-линейной транспортной задачи (8.1)–(8.4).

Замечание 1. *На каждом k-м шаге параметрического метода при решении линейной транспортной задачи* (8.8) *с фиксированном значение параметра $\lambda = \mu_k$ в качестве первоначального опорного плана используем полученное на предыдущем шаге оптимальное x_{k-1}^*, т.е. на каждом k-м шаге параметрического метода дооптимизируем предыдущее оптимальное решение x_{k-1}^* с несущественными изменениями значениями коэффициентов линейной целевой функцией.*

Замечание 2. *Если задача* (8.1)–(8.4) *решается неоднократно в некотором итерационном процессе с незначительными изменениями коэффициентов c_{ij} и d_{ij}, то можно решать только одну линейную транспортную задачу типа* (8.8) *при фиксированном значении параметра λ. Пусть на предыдущем шаге итерационного метода получено оптимальное решение x_{k-1}^* задачи* (8.8) *при фиксированном значении параметра $\lambda = \lambda_{k-1}$. Тогда на k-ом шаге итерационного метода, после незначительных изменениях коэффициентов c_{ij} и d_{ij} решаем только одну задачу типа* (8.8) *при $\lambda = \lambda_k = F(x_{k-1}^*)$.*

3. Алгоритмы метода покоординатного спуска. Рассмотрим случай дробно-линейной транспортной задачи, когда объемы поставки или потребления неограничены, т.е. в задаче (8.1)– (8.4) отсутствуют ограничения (8.2) или (8.3). Если предположить, что отсутствуют ограничения (8.3), то имеем следующую дробно-линейную задачу:

$$F(x) = \frac{C(x)}{D(x)} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}x_{ij}} \to \min, \tag{8.9}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}), \tag{8.10}$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}). \tag{8.11}$$

Пусть $R(x)$ – множество допустимых решений задачи (8.9)–(8.11). Так как ограничения (8.10) имеют специальную структуру, то решение задачи (8.9)–(8.11) сводится к нахождению индексов $\{j_1, j_2, \ldots, j_m\}$ таких, что план x, определенный по формуле

$$x_{ij} = \begin{cases} a_i, \text{для} & i = \overline{1,m}; \quad j = j_i, \\ 0, \text{ для} & i = \overline{1,m}; \quad j \neq j_i \end{cases}$$

является оптимальным.

При разработке вычислительного алгоритма решения задачи (8.9)–(8.11) используется тот факт, что для любого опорного плана в каждом равенстве (8.10) только одна переменная отлична от нуля и равна a_i и что соседние крайние точки многогранника $R(x)$ различаются только двумя координатами.

Для определения первоначального опорного плана задачи (8.9)–(8.11) используем метод минимального отношения в строке: для каждого i находим $j = l$, для которого

$$c_{il}/d_{il} = \min_j (c_{ij}/d_{ij}), \quad d_{ij} \neq 0,$$

и полагаем

$$x_{ij} = \begin{cases} a_i, \text{для} & i = \overline{1,m};\ j = l, \\ 0, \text{ для} & i = \overline{1,m};\ j \neq l. \end{cases}$$

Определим процесс перехода от одного опорного плана задачи (8.9)–(8.11) к другому. Для того чтобы перейти к новому опорному плану x', достаточно найти такие индексы k, l и t, для которых $x_{kl} = a_k$, а $x_{kt} = 0$, и положить $x'_{kt} = a_k$, а $x'_{kl} = 0$. Остальные компоненты плана x' совпадают с компонентами плана x. Новый план x' является опорным для задачи (8.9)–(8.11). Это означает переход от одной крайней точки к другой (соседней), которые различаются только двумя координатами.

Значение функционала (8.9) в новой точке определяется по формуле

$$\frac{C(x')}{D(x')} = \frac{C(x) + a_k(c_{kt} - c_{kl})}{D(x) + a_k(d_{kt} - d_{kl})}.$$

Обозначим $\Delta^1_{kt} = (c_{kt} - c_{kl}),\ \Delta^2_{kt} = (d_{kt} - d_{kl})$ и определим разность

$$\frac{C(x')}{D(x')} - \frac{C(x)}{D(x)} = \frac{a_k}{D(x')D(x)}\Big(D(x)\Delta^1_{kt} - C(x)\Delta^2_{kt}\Big). \tag{8.12}$$

Так как $D(x) > 0$, $D(x') > 0$ и $a_i > 0$, то знак разности (8.12) зависит от знака выражения

$$\Delta_{kt} = D(x)\Delta^1_{kt} - C(x)\Delta^2_{kt}. \tag{8.13}$$

Сформулируем следующий признак оптимальности опорного плана x задачи (8.9)–(8.11).

Теорема 8.1. *Если $\Delta_{ij} \geq 0$ для всех i и j, то опорный план x является оптимальным для задачи* (8.9)–(8.11).

Рассмотрим следующий итерационный алгоритм покоординатного спуска решения задачи (8.9)–(8.11).

1. Определяем первоначальный опорный план x методом минимального отношения в строке.

2. Находим значения $C(x)$ и $D(x)$.

3. Вычисляем значения выражения Δ_{ij} по формуле (8.13) для всех i и j, для которых $x_{ij} = 0$.

4. Находим $\Delta_{kt} = \min\limits_{ij} \Delta_{ij}$.

5. Если $\Delta_{kt} \geq 0$, то план x является оптимальным.

6. Если $\Delta_{kt} < 0$, то переходим к новому плану x' :

$$x'_{ij} = \begin{cases} a_k, \text{ для } & i = k;\ j = t, \\ 0, \quad \text{для } & i = k;\ j = l, \\ x_{ij}, \text{для всех } & i \neq k;\ j \neq l;\ j \neq t, \end{cases}$$

где l – индекс, для которого $x_{kl} = a_k$.

7. Находим новые значения числителя и знаменателя по формулам

$$C(x') = C(x) + a_k \Delta^1_{kt}, \quad D(x') = D(x) + a_k \Delta^2_{kt}$$

и переходим к третьему пункту алгоритма.

Изложенный выше алгоритм позволяет найти оптимальное решение задачи (8.9)–(8.11) за конечное число итераций. Однако иногда целесообразно найти некоторое приближенное решение за одну итерацию алгоритма, который состоит в следующем.

1. Определяем первоначальный опорный план x методом минимального отношения в строке.

2. Ставим $i = 1$.

3. Находим значения $C(x)$ и $D(x)$.

4. Вычисляем значения выражения Δ_{ij} по формуле (8.13) для всех $j = \overline{1, n}$, и $j \neq l$, где l – индекс, для которого $x_{ij} = a_i$.

5. Находим $\Delta_{kt} = \min\limits_{ij} \Delta_{ij}$.

6. Если $\Delta_{it} \geq 0$, то переходим к девятому пункту алгоритма.

7. Если $\Delta_{kt} < 0$, то переходим к новому плану x' :

$$x'_{ij} = \begin{cases} a_k, \text{ для } & j = t, \\ 0, \quad \text{для } & j = l, \\ x_{ij}, \text{для всех } & j \neq l;\ j \neq t. \end{cases}$$

8. Находим новые значения числителя и знаменателя по формулам

$$C(x') = C(x) + a_i \Delta^1_{it}, \quad D(x') = D(x) + a_i \Delta^2_{it}.$$

9. Ставим $i = i + 1$, и если $i \leq m$, то перейдем к четвертому пункту, в противном случае – конец работы алгоритма.

Кроме этих алгоритмов, для решения задачи (8.9)–(8.11) можно использовать алгоритм параметрического метода, который в данном случае состоит в следующем.

1. Находим оптимальное решение x^* задачи

$$\min_{x \in R(x)} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij},$$

т.е. решаем линейную задачу

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min,$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}),$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n})$$

следующим образом: для каждого i находим $j = l$, для которого $c_{il} = \min\limits_j c_{ij}$, и полагаем

$$x'_{ij} = \begin{cases} a_i, \text{для} \quad i = \overline{1,m};\ j = l, \\ 0, \ \text{для} \quad i = \overline{1,m};\ j \neq l. \end{cases}$$

2. Находим оптимальное решение x' линейной задачи

$$\min_{x \in R(x)} \sum_{i=1}^{m}\sum_{j=1}^{n} \Big(c_{ij} - F(x^*)d_{ij}\Big)x_{ij}$$

по формуле

$$x'_{ij} = \begin{cases} a_i, \text{для} \quad i = \overline{1,m};\ j = l; \\ 0, \ \text{для}\, i = \overline{1,m};\ j \neq l, \end{cases}$$

где l – значения индексов j $(j = \overline{1,n})$, для которых

$$\Big(c_{il} - F(x^*)d_{il}\Big) = \min_j \Big(c_{ij} - F(x^*)d_{ij}\Big).$$

3. Если $F(x^*) = F(x')$, то оптимальный план x^* линейной задачи является оптимальным решением и для дробно-линейной задачи (8.1)–(8.4).

4. Если $F(x^*) \neq F(x')$, то ставим $x^* = x'$ и переходим ко второму пункту алгоритма.

Замечание. *В некоторых вычислительных процедурах, когда задача* (8.9)–(8.11) *решается циклически с несущественными изменениями значений коэффициентов c_{ij} и d_{ij}, то достаточно выполнить только один раз пункт 2 алгоритма параметрического метода, где x^* представляет собой решение, полученное на предыдущей итерации цикла, а x' новое приближенное решение задачи* (8.9)–(8.11). *Таким образом получаем одноитерационный алгоритм параметрического метода.*

4. Алгоритм субградиентного метода. Рассмотрим дробно-линейную транспортную задачу:

$$F(x) = \frac{C(x)}{D(x)} = \frac{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} c_{ij}x_{ij}}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} d_{ij}x_{ij}} \to \min, \tag{8.14}$$

$$\sum_{j=1}^{n} x_{ij} \le a_i \quad (i = \overline{1,m}), \tag{8.15}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,n}), \tag{8.16}$$

$$x_{ij} \ge 0 \quad (i = \overline{1,m};\ j = \overline{1,n}), \tag{8.17}$$

в которой предполагается, что объем поставки превышает объем потребления, а количество потребителей намного больше, чем количество поставщиков. Тогда для ее решения можно использовать алгоритмы, основанные на схемах декомпозиции по ограничениям и субградиентных методов, или же декомпозиционный метод Данцига-Вулфа.

Так как $D(x) > 0$, то решение задачи (8.14)–(8.17) сводится к максимизации кусочно-линейной вогнутой функции $L^*(u)$ при ограничениях $u \ge 0$. Функция $L^*(u)$ имеет вид

$$L^*(u) = \min_{x \in R(x)} L(x, u), \tag{8.18}$$

в которой

$$L(x,u) = \frac{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} c_{ij}x_{ij} + \sum\limits_{i=1}^{m} u_i\left(\sum\limits_{j=1}^{n} x_{ij} - a_i\right)}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} d_{ij}x_{ij}},$$

а $R(x)$ – ограниченное выпуклое многогранное множество значений переменных x, удовлетворяющих ограничениям (8.16), (8.17), а $u = (u_1, u_2, \ldots, u_m)$ – множители Лагранжа для ограничений (8.15), $u \ge 0$.

Рассмотрим следующий алгоритм схемы декомпозиции по ограничениям для решения дробно-линейной транспортной задачи.

1. Решаем задачу максимизации функции $L^*(u)$ при ограничениях $u \geq 0$ одним из методов недифференцируемой оптимизации. Тогда на $(t+1)$-м шаге субградиентного метода необходимо выполнить основные три этапа схемы декомпозиции по ограничениям.

а) Решить задачу дробно-линейного программирования (8.18), которая соответствует задачи (8.9)–(8.11). При фиксированных значениях $u = u^t$ для ее решения используем один из алгоритмов параметрического метода или покоординатного спуска и находим решение x_t^*. Следует заметить, что при повторном решении задачи (8.18) на $(t+1)$-м шаге субградиентного метода изменяются по отношению к предыдущему t-му шагу только значения коэффициентов числителя $c_{ij}^{t+1} = c_{ij} + u_i^{t+1}$, т.е. они изменяются на величины $h_{t+1} \cdot g_i(u^t)$, которые, в соответствии с субградиентным методом, стремятся к нулю. Поэтому в данном случае для решения задачи (8.18) можно использовать одноитерационный алгоритм параметрического метода, т.е. решить только одну задачу типа

$$\sum_{i=1}^{m}\sum_{j=1}^{n} q_{ij}x_{ij} \to \min,$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,n}),$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}),$$

в которой $q_{ij} = c_{ij} + u_i^t - \lambda_t d_{ij}$, а $\lambda_t = L(x_{t-1}^*, u^t)$, решение которой определяется по формуле:

$$x_{ij}^* = \begin{cases} b_j, \text{ для } j = \overline{1,n}, & i = k; \\ 0, \text{ для } j = \overline{1,n}, & i \neq k, \end{cases}$$

где k значение индексов i $\ (i = \overline{1,m})$ для которых

$$q_{kj} = \min_i \{q_{ij} = c_{ij} + u_i^t - \lambda_t d_{ij}\}.$$

б) Вычислить значения обобщенного градиента функции $L^*(u)$ в точке $u = u^t$ по формуле

$$g_i(u^t) = \left(\sum_{i \in M_i} b_j - a_i\right) \Big/ D(x_t^*) \quad (i = \overline{1,m}),$$

где M_i – множество значений индексов j, для которых $x^*_{ij} = b_j$ в оптимальном решением задачи (8.18). Используя множества M_i оптимальное решение x^*_t представим следующим образом:

$$x^*_t = \{x^*_{ij}\colon x^*_{ij} = b_j,\ j \in M_i,\ x^*_{ij} = 0,\ j \notin M_i,\ i = \overline{1,m}\}.$$

в) Найти новые значения

$$u_i^{t+1} = \max\left\{0, u_i^t + h_{t+1} \cdot g_i(u^t)\right\} \quad (i = \overline{1,m}),$$

где h_{t+1} – величина шага.

2. Пусть u^* – оптимальное решение задачи максимизации функции $L^*(u)$, а x^* – оптимальное решение задачи (8.18) при фиксированных $u = u^*$. Тогда находим значения

$$\Delta^1_{ij} = c_{kj} + u^*_k - c_{ij} - u^*_i; \quad \Delta^2_{ij} = d_{kj} - d_{ij}$$

для всех i и j, где k – индекс для которого $j \in M_k$.

3. Находим множество J тех значений индексов j, для которых имеет место выражение

$$\Delta_{ij} = |\Delta^1_{ij} - F(x^*)\Delta^2_{ij}| \le \varepsilon$$

хотя бы для одного значения индекса $i \ne k$, где $\varepsilon > 0$ – достаточно малое число.

4. Находим множество I тех значений индексов i, для которых коэффициенты a'_i, определенные по формуле

$$a'_i = a_i - \sum_{j \notin J} x^*_{ij},$$

положительны.

5. По алгоритму метода потенциалов или параметрического метода решаем дробно-линейную транспортную задачу:

$$\frac{\sum\limits_{i\in I}\sum\limits_{j\in J} c_{ij}x_{ij} + c_0}{\sum\limits_{i\in I}\sum\limits_{j\in J} d_{ij}x_{ij} + d_0} \to \min;$$

$$\sum_{j\in J} x_{ij} \le a'_i, \quad i \in I;$$

$$\sum_{i \in I} x_{ij} = b_j, \quad j \in J;$$

$$x_{ij} \geq 0, \quad i \in I; \ j \in J;$$

в которой

$$c_0 = \sum_{i \notin I} \sum_{j \notin J} c_{ij} x_{ij}^*, \quad d_0 = \sum_{i \notin I} \sum_{j \notin J} d_{ij} x_{ij}^*$$

и она имеет существенно уменьшенные размеры относительно первоначальной задачи (8.14)–(8.17), т.е. имеем $||I|| \approx ||J|| \approx m$.

5. Алгоритм декомпозиции Данцига-Вулфа. Другой декомпозиционный алгоритм задачи (8.14)–(8.17) может быть построен на базе принципа разложения Данцига-Вулфа. Если через $\{x_{ij}^k\}$ $(k = \overline{1, r})$, обозначить координаты всех крайних точек многогранника $R(x)$, то любая точка $x \in R(x)$ может быть представлена в виде:

$$x_{ij} = \sum_{k=1}^{r} \lambda_k x_{ij}^k \quad (i = \overline{1, m}; \ j = \overline{1, n}),$$

$$\sum_{k=1}^{r} \lambda_k = 1, \quad \lambda_k \geq 0 \quad (k = \overline{1, r}).$$

Тогда задача (8.14)–(8.17) сводится к следующей координирующей задаче:

$$\frac{P(x, \lambda)}{Q(x, \lambda)} = \frac{\sum\limits_{k=1}^{r} p_k \lambda_k}{\sum\limits_{k=1}^{r} q_k \lambda_k} \to \min, \tag{8.19}$$

$$\sum_{k=1}^{r} t_{ik} \lambda_k \leq a_i \quad (i = \overline{1, m}), \tag{8.20}$$

$$\lambda_k \geq 0 \quad (k = \overline{1, r}), \tag{8.21}$$

где

$$p_k = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}^k \quad (k = \overline{1, r}),$$

$$q_k = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij}^k \quad (k = \overline{1, r}),$$

$$t_{ik} = \sum_{j=1}^{n} x_{ij}^{k} \quad (i = \overline{1,m};\ k = \overline{1,r}).$$

Задача (8.19)–(8.21) является задачей дробно-линейного программирования относительно переменных λ_k $(k = \overline{1,r})$, для решения которой используется модифицированный симплекс-метод.

Пусть B – текущий базис, B^{-1} – соответствующая обратная матрица, λ_B – значения переменных λ_k в данном базисе, $P(x_B, \lambda_B)$ и $Q(x_B, \lambda_B)$ – соответственно значения числителя и знаменателя в текущем базисе, а x_B – крайние точки $\{x_{ij}^{k}\}$, соответствующие переменным $\lambda_B = \{\lambda_k\}$. Тогда:

1) находим значения двойственных оценок ограничений (8.20) соответственно для числителя u и знаменателя v по формулам

$$u = p_B B^{-1} \quad \text{и} \quad v = q_B B^{-1},$$

где p_B и q_B – векторы коэффициентов числителя и знаменателя, соответствующие текущему базису B;

2) находим l-ю крайнюю точку $\{x_{ij}^{l}\}$ многогранника $R(x)$, в которой достигается минимальное значение функционала

$$F(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} \gamma_{ij} x_{ij},$$

при ограничениях

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,n}),$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}),$$

где $\gamma_{ij} = Q(x_B, \lambda_B)(c_{ij} + u_i) - P(x_B, \lambda_B)(d_{ij} + v_i)$. Координаты такой точки, которое соответствует оптимальному решению данной задачи определяются по формуле

$$x_{ij}^{l} = \begin{cases} b_j, \text{для} \quad j = \overline{1,n}; \quad i = s; \\ 0, \ \text{для} \quad j = \overline{1,n}; \quad i = \overline{1,m}; \quad i \neq s, \end{cases}$$

где s – значения индекса i, для которого $\gamma_{sj} = \min\limits_{i} \gamma_{ij}$;

3) если $F(x^l) \geq 0$, то текущий базис является оптимальным;

4) пусть $F(x^l) < 0$, тогда вектор l вводится в базис, для чего находятся значения коэффициентов p_l, q_l и t_{il} по соответствующим формулам.

Следует заметить, что перед вводом вектора l в базис, его координаты p_l, q_l и t_{il} необходимо умножить на обратную матрицу B^{-1}.

Пусть $\{\lambda_k^*\}$ – оптимальное решение задачи (8.19)–(8.21), тогда оптимальное решение для исходной задачи (8.14)–(8.17) определяется по формуле

$$x_{ij}^* = \sum_{k=1}^{m} \lambda_k^* x_{ij}^k \quad (i = \overline{1,m};\ j = \overline{1,n}),$$

где $\{x_{ij}^k\}$ – координаты крайних точек многогранника $R(x)$, соответствующие базисным переменных λ_k оптимального базиса B.

Таким образом решение дробно-линейной транспортной задачи (8.14)–(8.17) сводится к задаче дробно-линейного программирования (8.19)–(8.21) с уменьшенным количеством ограничений, специальным правилом выбора вектора для ввода в базис и новым критерием оптимальности.

8.2. Обобщенная дробно-линейная транспортная задача

Пусть задана обобщенная (распределительная) дробно-линейная транспортная задача

$$F(x) = \frac{C(x)}{D(x)} = \frac{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} c_{ij}x_{ij}}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} d_{ij}x_{ij}} \to \min, \tag{8.22}$$

$$\sum_{i=1}^{m} f_{ij}x_{ij} \le b_j \quad (j = \overline{1,n}), \tag{8.23}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}), \tag{8.24}$$

$$x_{ij} \ge 0 \quad (i = \overline{1,m};\ j = \overline{1,n}), \tag{8.25}$$

где a_i, b_j и f_{ij} – положительные числа. Обобщение дробно-линейной транспортной задачи (8.1)–(8.4) проводится в смысле, что коэффициенты перед переменных x_{ij} в ограничениях (8.3) отличны от единицы и равны числами f_{ij}, которые как правило являются целыми и положительными.

Предположим, что $D(x) > 0$ для любых допустимых планов задачи (8.22)–(8.25).

Рассмотрим двойственную к ней задачу

$$\frac{\sum_{i=1}^{m} a_i u_i^1 - \sum_{j=1}^{n} b_j v_j^1}{\sum_{i=1}^{m} a_i u_i^2 - \sum_{j=1}^{n} b_j v_j^2} \to \max, \tag{8.26}$$

$$\begin{gathered}(c_{ij} - u_i^1 + v_j^1 f_{ij})\left(\sum_{i=1}^{m} a_i u_i^2 - \sum_{j=1}^{n} b_j v_j^2\right) - \\ -(d_{ij} - u_i^2 + v_j^2 f_{ij})\left(\sum_{i=1}^{m} a_i u_i^1 - \sum_{j=1}^{n} b_j v_j^1\right) \geq 0 \quad (i = \overline{1, m};\ j = \overline{1, n}),\end{gathered} \tag{8.27}$$

$$v_j^1 \geq 0;\ v_j^2 \geq 0 \quad (j = \overline{1, n}), \tag{8.28}$$

$$\sum_{i=1}^{m} a_i u_i^2 - \sum_{j=1}^{n} b_j v_j^2 \geq 0. \tag{8.29}$$

Для задач (8.22)–(8.25) и (8.26)–(8.29) справедливы основные теоремы двойственности.

Ниже рассмотрим алгоритмы решения обобщенных дробно-линейных транспортных задач (8.22)–(8.25).

1. Алгоритм метода потенциалов. Применим метод потенциалов для решения задачи (8.22)–(8.25). По своей структуре алгоритм метода потенциалов решения обобщенной задачи с дробно-линейным функционалом аналогичен алгоритмам линейных и дробно-линейных транспортных задач [493, 494, 601] и состоит в следующем.

1. Находим первоначальный опорный план задачи (8.22)–(8.25) методом "северо-западного угла" или минимального отношения c_{ij}/d_{ij}, если все $d_{ij} \neq 0$.

2. Определяем значения потенциалов u^1, u^2, v^1 и v^2, решая следующие системы уравнений

$$\{u_i^1 - v_j^1 f_{ij} = c_{ij}\} \quad \text{и} \quad \{u_i^2 - v_j^2 f_{ij} = d_{ij}\}$$

для всех i и j, для которых $x_{ij} \neq 0$.

3. Определяем значение числителя $C(x)$ и знаменателя $D(x)$ функционала (8.22).

4. Находим $\Delta^1_{ij} = c_{ij} - u^1_i + v^1_j f_{ij}$ и $\Delta^2_{ij} d_{ij} - u^2_i + v^2_j f_{ij}$ для всех i, j, для которых $x_{ij} \neq 0$.

5. Вычисляем

$$\Delta_{ij} = D(x)\Delta^1_{ij} - C(x)\Delta^2_{ij}$$

и находим $\Delta_{rt} = \min_{ij} \Delta_{ij}$.

6. Если $\Delta_{rt} \geq 0$, то план x является оптимальным.

7. Если $\Delta_{rt} < 0$, то переходим к новому опорному плану x' задачи (8.22)–(8.25) по известным правилам линейного случая [601] и возвращаемся ко второму шагу алгоритма. Полученное решение x по данному алгоритму является оптимальным для исходной задачи (8.22)–(8.25), а соответствующие потенциалы u^1, u^2, v^1 и v^2 – решением двойственной задачи (8.26)–(8.29).

2. Алгоритм параметрического метода. Применим параметрический метод для решения обобщенной дробно-линейной транспортной задачи. Обозначим через $M(x)$ множество значений переменных x, удовлетворяющих ограничениям (8.23)–(8.25), и рассмотрим следующую задачу параметрического программирования

$$\min_{x \in M(x)} \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} - \lambda d_{ij}) x_{ij}, \tag{8.30}$$

которая является параметрической линейной распределительной задачей транспортного типа.

Как и для задачи дробно-линейного программирования, а также для дробно-линейной транспортной задачи (8.1)–(8.4) алгоритм параметрического метода решения обобщенной дробно-линейной транспортной задачи (8.22)–(8.25) заключается в следующем.

0-й шаг. Берем $\lambda = \mu_0 = 0$ и находим оптимальное решение x^*_0 задачи (8.30), которая является обычной распределительной задачей с линейным функционалом.

k-й шаг. Определяем значение параметра $\mu_k = F(x^*_{k-1})$ и находим оптимальное решение x^*_k задачи (8.30) при $\lambda = \mu_k$. Если $F(x^*_k) = F(x^*_{k-1})$, то план x^*_k является оптимальным решением и для задачи (8.22)–(8.25).

Замечание. *При повторном решении задачи* (8.22)–(8.25) *с незначительными изменениями коэффициентов числителя можно использовать одноитерационный алгоритм параметрического метода, т.е.*

используя предыдущее решение x^* решаем только одну задачу (8.30) (*так называемый одноитерационный алгоритм параметрического метода*).

В случае, когда n намного больше m, для решения линейных распределительных задач эффективно используют декомпозиционные алгоритмы [335, 543, 602, 635, 639, 662, 715, 733].

Рассмотрим два алгоритма декомпозиции, основанные на схеме разложении по ограничениям и принципа Данцига-Вулфа.

3. Алгоритм субградиентного метода. Для задачи (8.22)–(8.25) на множестве ограничений (8.23) построим функцию Лагранжа

$$L(x,v) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{j=1}^{n} v_j \left(\sum_{i=1}^{m} f_{ij}x_{ij} - b_j \right)}{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}x_{ij}}$$

и рассмотрим задачу

$$L^*(v) = \min_{x \in R(x)} L(x,v), \tag{8.31}$$

в которой $R(x)$ – ограниченное выпуклое многогранное множество значений переменных x, удовлетворяющих ограничениям (8.24), (8.25), а $v = (v_1, v_2, \ldots, v_n)$ – множители Лагранжа для ограничений (8.23), $v \geq 0$.

Для решения задачи (8.22)–(8.25) применяем схему декомпозиции по ограничениям, которая сводится к следующему.

1. Решаем задачу максимизации функции $L^*(v)$ при ограничениях $v \geq 0$. Так как функция $L^*(v)$ является кусочно-линейной и вогнутой, то для ее максимизации используем одним из субградиентным методов недифференцируемой оптимизации. Тогда на $(t+1)$-м шаге субградиентного метода необходимо выполнить основные три этапа вычислений по схеме декомпозиции по ограничениям.

а) Решить задачу дробно-линейного программирования (8.31), которая соответствует задачи (8.9)–(8.11) при фиксированных значениях $v = v^t$ по одному из алгоритмов параметрического метода или покоординатного спуска, т.е. найти решение (оптимальное или приближенное) $x_t^*(v)$ задачи (8.31).

Замечание. *Так как коэффициенты* c_{ij} *в задаче типа* (8.9)–(8.11)*, которая соответствует задачи* (8.31) *изменяются соответственно на величины* $h_{t+1} \cdot g_j(v^t) \cdot f_{ij}$*, которые стремятся к нулю, то для решения задачи* (8.31) *можно использовать одноитерационный алгоритм параметрического метода, т.е. решить только одну задачу типа*

$$\sum_{i=1}^{m}\sum_{j=1}^{n} q_{ij}x_{ij} \to \min,$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}),$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}),$$

в которой $q_{ij} = c_{ij} + v_j^t f_{ij} - \lambda_t d_{ij}$, *а* $\lambda_t = L(x_{t-1}^*, v^t)$.

б) Вычислить значения обобщенного градиента функции $L^*(v)$ в точке $v = v^t$ по формуле

$$g_j(v^t) = \frac{\sum\limits_{i=1}^{m} f_{ij}x_{ij}^*(v) - b_j}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} d_{ij}x_{ij}^*(v)} \quad (j = \overline{1,n}),$$

где $\{x_{ij}^*(v)\}$ – решение задачи (8.31) при $v = v^t$.

в) Найти новые значения

$$v_j^{t+1} = \max\left\{0, v_j^t + h_{t+1} \cdot g_j(v^t)\right\} \quad (j = \overline{1,n}),$$

где h_{t+1} – величина шага.

2. Пусть v^* – оптимальное решение задачи максимизации функции $L^*(v)$, а x^* – оптимальное решение задачи (8.31) при фиксированных $v = v^*$. Тогда находим значения

$$\Delta_{ij}^1 = c_{ik} + v_k^* f_{ik} - c_{ij} - v_j^* f_{ij};\quad \Delta_{ij}^2 = d_{ik} - d_{ij}$$

для всех i и j для которых $x_{ij}^*(v) = 0$, где k – значение индекса j, для которого $x_{ik}^*(v) = a_i$.

3. Находим множество I тех значений индексов i, для которых имеет место выражение

$$\Delta_{ij} = |\Delta_{ij}^1 - F(x^*)\Delta_{ij}^2| \le \varepsilon$$

хотя бы для одного значения индекса $j \ne k$, где $\varepsilon > 0$ – достаточно малое число.

4. Находим множество J тех значений индексов j, для которых коэффициенты b_j', определенные по формуле

$$b_j' = b_j - \sum_{i \notin I} f_{ij} x_{ij}^*,$$

положительны.

5. По алгоритму метода потенциалов или параметрического метода решаем распределительную дробно-линейную транспортную задачу:

$$\frac{\sum\limits_{i \in I} \sum\limits_{j \in J} c_{ij} x_{ij} + c_0}{\sum\limits_{i \in I} \sum\limits_{j \in J} d_{ij} x_{ij} + d_0} \to \min;$$

$$\sum_{i \in I} f_{ij} x_{ij} \le b_j', \quad j \in J;$$

$$\sum_{j \in J} x_{ij} = a_i, \quad i \in I;$$

$$x_{ij} \ge 0, \quad i \in I;\ j \in J,$$

в которой $||I|| \approx ||J|| \approx m,$ а

$$c_0 = \sum_{i \notin I} \sum_{j \notin J} c_{ij} x_{ij}^*, \quad d_0 = \sum_{i \notin I} \sum_{j \notin J} d_{ij} x_{ij}^*.$$

4. Алгоритм декомпозиции Данцига-Вулфа. Обозначим через $\{x_{ij}^k\}$ $(k = \overline{1,r})$, координаты всех крайних точек многогранника $R(x)$. Тогда любая точка $x \in R(x)$ может быть представлена в виде

$$x_{ij} = \sum_{k=1}^{r} \lambda_k x_{ij}^k \quad (i = \overline{1,m};\ j = \overline{1,n}),$$

$$\sum_{k=1}^{r} \lambda_k = 1, \quad \lambda_k \ge 0 \quad (k = \overline{1,r}).$$

Тогда задача (8.22)–(8.25) сводится к следующей координирующей задаче:

$$\frac{P(x,\lambda)}{Q(x,\lambda)} = \frac{\sum\limits_{k=1}^{r} p_k\lambda_k}{\sum\limits_{k=1}^{r} q_k\lambda_k} \to \min, \tag{8.32}$$

$$\sum_{k=1}^{r} t_{jk}\lambda_k \le b_j \quad (j=\overline{1,n}), \tag{8.33}$$

$$\sum_{k=1}^{r} \lambda_k = 1, \tag{8.34}$$

$$\lambda_k \ge 0 \quad (k=\overline{1,r}), \tag{8.35}$$

где

$$p_k = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}^k \quad (k=\overline{1,r}),$$

$$q_k = \sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}x_{ij}^k \quad (k=\overline{1,r}),$$

$$t_{jk} = \sum_{i=1}^{m} f_{ij}x_{ij}^k \quad (j=\overline{1,n};\ k=\overline{1,r}).$$

Задача (8.32)–(8.35) является задачей дробно-линейного программирования, для решения которой используем модифицированный симплекс-метод.

Рассмотрим алгоритм декомпозиции Данцига-Вулфа для решения обобщенной дробно-линейной транспортной задачи (8.22)–(8.25).

Пусть B – текущий базис, B^{-1} – соответствующая обратная матрица, λ_B – значения переменных λ_k в данном базисе, $P(x_B,\lambda_B)$ и $Q(x_B,\lambda_B)$ – соответственно значения числителя и знаменателя в текущем базисе, а x_B – крайние точки $\{x_{ij}^k\}$, соответствующие базисным переменным $\lambda_B = \{\lambda_k\}$ Тогда:

1) находим значения двойственных оценок ограничений (8.33) и (8.34) соответственно для числителя $u = (u_1, u_2, \ldots, u_n, \overline{u})$ и знаменателя $v = (v_1, v_2, \ldots, v_n, \overline{v})$ по формулам

$$u = p_B B^{-1} \;\text{ и }\; v = q_B B^{-1},$$

где p_B и q_B – векторы коэффициентов числителя и знаменателя для переменных, соответствующие текущему базису B;

2) находим l-ю крайнюю точку $x^l = \{x^l_{ij}\}$ многогранника $R(x)$, в которой достигается минимальное значение функционала

$$G(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} g_{ij} x_{ij} + g_0$$

при ограничениях

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}),$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}),$$

где

$$g_{ij} = Q(x_B, \lambda_B)(c_{ij} - u_j f_{ij}) - P(x_B, \lambda_B)(d_{ij} - v_j f_{ij});$$

$$g_0 = Q(x_B, \lambda_B)\overline{u} - P(x_B, \lambda_B)\overline{v}.$$

Порядковый номер l, соответствует индексу переменной λ_l, которую следует ввести в базис. Координаты такой точки находятся по формуле

$$x^l_{ij} = \begin{cases} a_i, \text{для} \quad i = \overline{1,m};\ j = s; \\ 0, \ \text{для} \quad i = \overline{1,m};\ j = \overline{1,n};\ j \neq s, \end{cases}$$

где s – значения индекса j, для которого $g_{is} = \min\limits_{j} g_{ij}$;

3) если $G(x^l) \geq 0$, то текущий базис является оптимальным.

4) пусть $G(x^l) < 0$, тогда вектор l вводится в базис, т.е. переменная λ_l станет базисной, для чего находятся значения коэффициентов p_l, q_l и t_{jl} по соответствующим формулам, которые формируют вектор-столбец, соответствующий переменной λ_l для ввода в базис.

Пусть $\{\lambda^*_k, \quad k = \overline{1, n+1}\}$ – оптимальное решение задачи (8.32)–(8.35), тогда оптимальное решение для исходной задачи (8.22)–(8.25) определяется по формуле

$$x^*_{ij} = \sum_{k=1}^{n+1} \lambda^*_k x^k_{ij} \quad (i = \overline{1,m};\ j = \overline{1,n}),$$

где $\{x^k_{ij}, \quad k = \overline{1, n+1}\}$ – координаты крайних точек многогранника $R(x)$, соответствующие базисным переменным $\{\lambda^*_k, \quad k = \overline{1, n+1}\}$ оптимального базиса B.

Таким образом решение обобщенной дробно-линейной транспортной задачи (8.22)–(8.25) сводится к задаче дробно-линейного программирования (8.32)–(8.35) с уменьшенным количеством ограничений, специальным правилом выбора вектора для ввода в базис и новым критерием оптимальности.

8.3. Производственно-транспортные задачи с дробно-линейными функционалами

Рассмотрим одну из постановок задач производственно-транспортного типа с дробно-линейным функционалом:

$$F(x,y)=\frac{C(x,y)}{D(x,y)}=\frac{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n}c_{ij}x_{ij}+\sum\limits_{j=1}^{n}c_jy_j}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n}d_{ij}x_{ij}+\sum\limits_{j=1}^{n}d_jy_j}\to\min, \tag{8.36}$$

$$\sum_{j=1}^{n}x_{ij}=a_i \quad (i=\overline{1,m}), \tag{8.37}$$

$$\sum_{i=1}^{m}x_{ij}=y_j \quad (j=\overline{1,n}), \tag{8.38}$$

$$\alpha_j\le y_j\le\beta_j \quad (j=\overline{1,n}), \tag{8.39}$$

$$x_{ij}=\begin{cases}a_i\\0\ \ (i=\overline{1,m};\ j=\overline{1,n}),\end{cases} \tag{8.40}$$

где m и n – соответственно количество пунктов потребления и производства продукции; a_i – объемы потребления продукции в i-м пункте; y_j – искомый объем производства продукции в j-м пункте; x_{ij} – искомый объем перевозки продукции i-му потребителю от j-го производителя; c_{ij} – характеристика транспортного процесса, имеющая "отрицательное" влияние на работу транспорта, при перевозке продукции i-му потребителю от j-го производителя; d_{ij} – характеристика транспортного процесса, имеющая "положительное" влияние на работу транспорта, при перевозке продукции i-му потребителю от j-го производителя; c_j – характеристика, имеющая "отрицательное" влияние на процесс производства продукции в j-м пункте; d_j – характеристика, имеющая "положительное" влияние на процесс производства продукции в j-м пункте;

α_j – минимальный возможный объем производства продукции в j-м пункте; β_j – максимальный возможный объем производства продукции в j-м пункте; c_0 и d_0 – постоянные величины.

В отличие от других постановок производственно-транспортных задач [639], задача (8.36)–(8.40) имеет дискретное ограничение (8.40), что приводит ее к классу целочисленных задач. Задачу (8.36)–(8.40) можно свести к задаче дробно-линейного программирования с булевыми переменными. Для решения такого рода задач можно использовать алгоритмы метода ветвей и границ.

Однако, учитывая специфику структуры ограничений, для решения задачи (8.36)–(8.40) можно привести более эффективные алгоритмы, удобные для решения практических задач с большим числом потребителей, т.е. ограничений (8.37).

1. Алгоритмы методов штрафных функций и вектора спада. Опишем случай задачи (8.36)–(8.40), когда любая частная сумма чисел a_i отлична от α_j и β_j. Рассмотрим приближенный алгоритм с использованием штрафных функций:

$$\varphi_j(y_j) = \begin{cases} c_j y_j, & \text{если } y_j \in [\alpha_j, \beta_j]; \\ c_j y_j + \gamma_j(\alpha_j, \beta_j)(y_j - \alpha_j)(y_j - \beta_j), & \text{если } y_j \notin [\alpha_j, \beta_j]. \end{cases}$$

Функции $\varphi_j(y_j)$ обладают следующими свойствами:

- на отрезке $[\alpha_j, \beta_j]$ они линейны;
- если все коэффициенты $\gamma_j(\alpha_j, \beta_j)$ положительны, то за пределами отрезков $[\alpha_j, \beta_j]$ они монотонно возрастают для $y_j > \beta_j$ и монотонно убывают для $y_j < \alpha_j$;
- функция $\varphi(y) = \sum_{j=1}^{n} \varphi_j(y_j)$ является непрерывной и кусочно-линейной строго выпуклой функцией.

Согласно введенным функциям $\varphi_j(y_j)$ задача (8.36)–(8.40) сводится к следующей задаче:

$$\Phi(x, y) = \frac{P(x, y)}{Q(x, y)} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{j=1}^{n} \varphi_j(y_j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} q_{ij}x_{ij}} \to \min, \tag{8.41}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1, m}), \tag{8.42}$$

$$x_{ij} = \begin{cases} a_i \\ 0 \ \ (i = \overline{1, m};\ j = \overline{1, n}), \end{cases} \tag{8.43}$$

где $q_{ij} = d_{ij} + d_j$, а переменные y_j определяются по формуле (8.38).

Задача (8.41)–(8.43) является задачей целочисленного дробного программирования, и ее оптимальное решение может быть найдено точными методами дискретного программирования. Однако практическая реализация таких методов для рассматриваемой задачи связана с большими вычислительными трудностями, поскольку она относится к классу NP-полных задач. Поэтому для ее решения целесообразнее использовать приближенные алгоритмы. Далее предлагается один из таких алгоритмов решения задачи (8.41)–(8.43), основанный на методе вектора спада [661].

Остановимся подробнее на специфике применения метода вектора спада для приближенного решения задачи (8.41)–(8.43). С этой целью на множестве G целочисленных решений задачи (8.41)–(8.43) определяем метрику

$$\rho(x, z) = \sum_{i=1}^{m} \sum_{j=1}^{n} sign(x_{ij} - z_{ij}),\ x, z \in G,$$

а для любой точки $x \in G$ введем понятие окрестности

$$O_G(x, r) = \{z \in G\colon \rho(x, z) \le r\}$$

относительно заданного радиуса r.

Тогда с помощью одного из алгоритмов метода вектора спада можно найти точку локального минимума функции $\Phi(x, y)$, т.е. точку минимума функции $\Phi(x, y)$ в окрестности радиуса r.

Заметим, что если ввести обозначение

$$R(x) = \Big\{x_{ij}\colon \sum_{j=1}^{n} x_{ij} = a_i \ \ (i = \overline{1, m}); \quad x_{ij} \ge 0 \ \ (i = \overline{1, m};\ j = \overline{1, n})\Big\},$$

то множество G целочисленных допустимых решений задачи (8.41)–(8.43) совпадает со множеством крайних точек многогранника $R(x)$, и существует взаимно-однозначное соответствие между элементами множества G и крайними точками многогранника $R(x)$.

Тогда если взять $r = 2$, то приближенный алгоритм метода вектора спада решения задачи (8.41)–(8.43) относительно окрестности радиуса r, по существу, означает следующее. Если находимся в некоторой крайней точке (x, y) многогранника $R(x)$, то в качестве окрестности $O_G(x, r)$

этой точки берется множество всех соседних с ней крайних точек. Для всех таких точек определяется значение координат вектора спада и выбирается направление, или точнее говоря крайняя точка, в которой значение функционала $\Phi(x, y)$ уменьшается. Если такой точки нет, то в текущей крайней точке (x, y) функционал $\Phi(x, y)$ достигает локального минимума. Таким образом, процесс решения задачи (8.41)–(8.43) по алгоритму метода вектора спада состоит в последовательном переходе от одной крайней точки многогранника $R(x)$ к другой (соседней), в которой значение функционала $\Phi(x, y)$ уменьшается.

Рассмотрим теперь вопросы перехода от одной крайней точки к другой (соседней) и определения значений координат вектора спада функции $\Phi(x, y)$ в заданной окрестности.

Для перехода от одного целочисленного решения задачи (8.41)–(8.43) к другому (т.е. от одной крайней точки многогранника $R(x)$ к другой, соседней) достаточно найти такие индексы k, l и t, для которых $x_{kl} = a_k$, а $x_{kt} = 0$. Тогда координаты соседней точки определяются по следующим формулам:

$$x'_{ij} = \begin{cases} a_k \text{ для } i = k;\ j = t, \\ 0 \quad \text{для } i = k;\ j = l, \\ x_{ij} \text{ для всех } i \neq k;\ j \neq l;\ j \neq t, \end{cases} \tag{8.44}$$

$$y'_j = \begin{cases} y_t + a_k \text{ для } j = t, \\ y_l - a_k \text{ для } j = l, \\ y_j \qquad\quad \text{для всех } j \neq l;\ j \neq t. \end{cases} \tag{8.45}$$

Значение функционала (8.41) в новой точке крайней точке (x, y) определяется по формуле

$$\frac{P(x', y')}{Q(x', y')} = \frac{P(x, y) + \Delta^1_{kt}(y_t)}{Q(x, y) + \Delta^2_{kt}},$$

где

$$\Delta^1_{kt}(y_t) = a_k(c_{kt} - c_{kl}) + \varphi_t(y_t + a_k) - \varphi_t(y_t) + \varphi_l(y_l - a_k) - \varphi_l(y_l);$$

$\Delta^2_{kt} = a_k(d_{kt} - d_{kl})$.

Находим разность

$$\frac{P(x', y')}{Q(x', y')} - \frac{P(x, y)}{Q(x, y)} = \frac{Q(x, y)\Delta^1_{kt} - P(x, y)\Delta^2_{kt}}{Q(x', y')Q(x, y)}. \tag{8.46}$$

Так как $Q(x, y) > 0$ и $Q(x', y') > 0$, то знак разности (8.46) зависит от знака выражения

$$\Delta_{kt}(y_t) = Q(x, y)\Delta^1_{kt} - P(x, y)\Delta^2_{kt}. \tag{8.47}$$

На множестве G определим следующую векторную функцию $\Delta(x, y) = \{\Delta_{ij}(y_j)\}$, для которой значения координат определяются по формуле (8.47). Тогда, если окрестность $O_G(x^*, r)$ с центром в точке (x^*, y^*) с заданным радиусом r определяется как множество всех соседних крайних точек многогранника $R(x)$, то в качестве вектора спада функции $\Phi(x, y)$ выбирается функция $\Delta(x, y)$.

Функция $\Delta(x, y)$ удовлетворяет всем условиям определения вектора спада [661] и она может быть использована для построения приближенного алгоритма нахождения локального минимума (относительно заданной окрестности) задачи (8.41)–(8.43).

Тогда можно сформулировать следующий признак достижимости локального минимума (относительно заданной окрестности) задачи (8.41)–(8.43).

Теорема 8.2. *Если $\Delta_{ij}(y_j) > 0$ для всех i и j, то соответствующая крайняя точка (x, y) многогранника $R(x)$ является точкой локального минимума задачи* (8.41)–(8.43).

На основе вышеизложенного можно предложить следующий приближенный алгоритм решения задачи (8.36)–(8.40), основанный на методах штрафных функций, вектора спада и покоординатного спуска.

1. Решаем задачу $\min\limits_{x \in R(x)} \Phi(x, y)$ по одному из алгоритмов параметрического метода или метода покоординатного спуска. Пусть (x, y) – оптимальное решение. Если для всех $j = \overline{1, n}$ $y_j \in [\alpha_j, \beta_j]$, то план (x, y) является оптимальным для задачи (8.36)–(8.40).

2. Вычисляем значения функций $\varphi(y_j)$ для плана (x, y) и находим значения $P(x, y)$ и $Q(x, y)$.

3. Вычисляем значения выражения $\Delta_{ij}(y_j)$ по формуле (8.47) для всех i и j, для которых $x_{ij} = 0$.

4. Находим $\Delta_{kt}(y_t) = \min\limits_{ij} \Delta_{ij}(y_j)$.

5. Если $\Delta_{kt}(y_t) \geq 0$, то план (x, y) дает локальный минимум задачи (8.41)–(8.43).

6. Если $\Delta_{kt}(y_t) < 0$, то находим новый план (x', y') по формулам (8.44), (8.45) и переходим ко второму пункту алгоритма.

7. Если все переменные $y_j \in [\alpha_j, \beta_j]$, то план (x, y) является допустимым для задачи (8.36)–(8.40), но не всегда оптимальным. В таком случае он берется в качестве некоторого приближенного решения задачи (8.36)–(8.40).

3. Алгоритм субградиентного метода. Приведем теперь приближенный алгоритм решения задачи (8.36)–(8.40), основанный на схеме декомпозиции по ограничениям и одного из субградиентных методов. Для этого задачу (8.36)–(8.40) перепишем в следующем виде:

$$\frac{P(x,y)}{Q(x,y)} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} q_{ij}x_{ij}} \to \min, \tag{8.48}$$

$$\sum_{i=1}^{m} x_{ij} \le \beta_j \quad (j = \overline{1,n}), \tag{8.49}$$

$$\sum_{i=1}^{m} x_{ij} \ge \alpha_j \quad (j = \overline{1,n}), \tag{8.50}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}), \tag{8.51}$$

$$x_{ij} = \begin{cases} a_i \\ 0 \quad (i = \overline{1,m};\ j = \overline{1,n}), \end{cases} \tag{8.52}$$

здесь предполагается, что количество пунктов потребления намного больше количества пунктов производства. Для данного алгоритма нет необходимости предполагать, что любая частная сумма чисел a_i отлична от α_j и β_j. Коэффициенты p_{ij} и q_{ij} определяются по формулам $p_{ij} = c_{ij} + c_j$ и $q_{ij} = d_{ij} + d_j$ для всех $i = \overline{1,m},\ j = \overline{1,n}$.

Так как $Q(x) > 0$, то решение задачи (8.48)–(8.52) сводится к максимизации функции $L^*(u,v)$ при ограничениях $u \ge 0$ и $v \ge 0$. Функция $L^*(u,v)$ имеет вид

$$L^*(u,v) = \min_{x \in R(x)} L(x,u,v), \tag{8.53}$$

где

$$L(x,u,v) = \frac{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n}(p_{ij}+u_j-v_j)x_{ij} + \sum\limits_{j=1}^{n}(\alpha_j v_j - \beta_j u_j)}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} q_{ij}x_{ij}}$$

– функция Лагранжа для задачи (8.48)–(8.50); $R(x)$ – ограниченное выпуклое многогранное множество значений переменных x, удовлетворяющих ограничениям (8.51), (8.52); $u = (u_1, u_2, \ldots, u_n)$ и $v = = (v_1, v_2, \ldots, v_n)$ – множители Лагранжа для ограничений (8.49), (8.50); $u \geq 0$ и $v \geq 0$. Обобщенный градиент функции $L^*(u,v)$ в точке (u,v) определяется по формулам

$$g_j(u) = \left(\sum_{i=1}^{m} x^*_{ij}(u,v) - \beta_j\right) \Big/ Q(x^*(u,v)) \quad (j = \overline{1,n}), \tag{8.54}$$

$$g_j(v) = \left(\alpha_j - \sum_{i=1}^{m} x^*_{ij}(u,v)\right) \Big/ Q(x^*(u,v)) \quad (j = \overline{1,n}), \tag{8.55}$$

где $x^*(u,v) = \{x^*_{ij}(u,v)\}$ – оптимальное решение задачи (8.53) при фиксированных значениях переменных u и v.

Для решения задачи (8.48)–(8.52) применяем схему декомпозиции по ограничениям, которая заключается в следующем.

1. Решаем задачу максимизации функции $L^*(u,v)$ при ограничениях $u \geq 0$ и $v \geq 0$. Тогда на t-м шаге субградиентного метода необходимо выполнить основные три этапа вычислений:

- решить задачу дробно-линейного программирования (8.53) при фиксированных значениях $u = u^t$ и $v = v^t$ по одному из алгоритмов параметрического метода или метода покоординатного спуска;
- вычислить значения обобщенного градиента функции $L^*(u,v)$ в точке (u^t, v^t) по формулам (8.54) и (8.55) с учетом полученного решения $x^*(u^t, v^t)$ задачи (8.53);
- найти новые значения

$$u_j^{t+1} = \max\left\{0,\ u_j^t + h_{t+1}\cdot g_j(u^t)\right\} \quad (j = \overline{1,n}),$$

$$v_j^{t+1} = \max\left\{0,\ v_j^t + h_{t+1}\cdot g_j(v^t)\right\} \quad (j = \overline{1,n}),$$

где h_{t+1} – величина шага.

2. Пусть (u^*, v^*) – оптимальное решение задачи максимизации функции $L^*(u, v)$. Тогда если выполняются условия (8.49) и (8.50), то оптимальный план $x^*(u, v)$ задачи (8.53), найденный при фиксированных $u = u^t$ и $v = v^t$, будет оптимальным и для задачи (8.48)–(8.52). В противном случае $x^*(u, v)$ будет приближенным решением задачи (8.48)–(8.52).

8.4. Обобщенная линейная и дробно-линейная транспортная задача

Рассмотрим следующую задачу:

$$F(x) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}x_{ij}} + \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} \to \min, \tag{8.56}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1, m}), \tag{8.57}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1, n}), \tag{8.58}$$

$$x_{ij} \geq 0 \quad (i = \overline{1, m};\ j = \overline{1, n}), \tag{8.59}$$

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j. \tag{8.60}$$

Задача (8.56)–(8.60) является обобщением задач линейного и дробно-линейного программирования [355, 431, 631, 700] но с транспортными ограничениями, т.е. является обобщением линейной и дробно-линейной транспортной задачи.

Обозначим через $M(x)$ множество значений переменных x, удовлетворяющих ограничениям (8.57)–(8.60). Введем обозначения

$$C(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}, \quad D(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}x_{ij}, \quad P(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij},$$

и предположим, что $D(x) > 0$ для всех $x \in M(x)$.

Известно [431, 700], что если $P(x) \leq 0$, то функционал $F(x)$ имеет несколько локальных минимумов, которые достигаются в крайних точках многогранника $M(x)$. Если же $P(x) > 0$, то функционал $F(x)$ имеет единственный минимум, который, вообще говоря, необязательно достигается в крайней точке многогранника $M(x)$.

Рассмотрим случай, когда $P(x) \leq 0$. Частные производные функции $F(x)$ в точке x определяются по формуле

$$\frac{\partial F(x)}{\partial x_{ij}} = \frac{c_{ij}D(x) - d_{ij}C(x)}{[D(x)]^2} + p_{ij} \quad (i = \overline{1,m};\ j = \overline{1,n}).$$

Пусть имеем некоторую крайнюю точку $x' \in M(x)$. Тогда если частная производная по направлению ребра многогранника $M(x)$, ведущего в некоторую соседнюю точку x'', отрицательна, то значение функционала $F(x'') < F(x')$. Если же производная по всем направлениям неотрицательна, то в этой точке функционал $F(x)$ достигает своего минимума. Рассмотрим эффективный способ вычисления производной по направлению ребра многогранника $R(x)$.

Пусть x – некоторый опорный план задачи (8.56)–(8.60). Введем переменные $u = \{u_i^1, u_i^2, u_i^3\}$ и $v = \{v_j^1, v_j^2, v_j^3\}$, для которых выполняются следующие равенства:

$$\left.\begin{aligned} c_{ij} &= u_i^1 + v_j^1, \\ d_{ij} &= u_i^2 + v_j^2, \\ p_{ij} &= u_i^3 + v_j^3 \end{aligned}\right\} \tag{8.61}$$

для всех i и j, для которых $x_{ij} \neq 0$, и

$$\begin{aligned} c'_{ij} &= c_{ij} - u_i^1 - v_j^1, \\ d'_{ij} &= d_{ij} - u_i^2 - v_j^2, \\ p'_{ij} &= p_{ij} - u_i^3 - v_j^3 \end{aligned}$$

для всех i и j, для которых $x_{ij} = 0$. Тогда

$$C(x) = \sum_{i,j \in S} c'_{ij} x_{ij} + L_1(u^1, v^1),$$

$$D(x) = \sum_{i,j \in S} d'_{ij} x_{ij} + L_2(u^2, v^2),$$

$$P(x) = \sum_{i,j \in S} p'_{ij} x_{ij} + L_3(u^3, v^3),$$

где

$$L_k(u^k, v^k) = \sum_{i=1}^{m} a_i u_i^k + \sum_{j=1}^{n} b_j v_j^k, \quad k = 1,\, 2,\, 3, \tag{8.62}$$

а S – множество индексов небазисных переменных опорного плана x.

Таким образом, частные производные функционала $F(x)$ по направлениям ребер многогранника $M(x)$, идущих в соседние точки опорного плана x, определяются по формулам

$$\frac{\partial F(x)}{\partial x_{ij}} = \frac{c'_{ij} L_2(u^2, v^2) - d'_{ij} L_1(u^1, v^1)}{[L_2(u^2, v^2)]^2} + p'_{ij}, \quad i, j \in S$$

или

$$\Delta_{ij} = \frac{L_2(u^2, v^2)(c_{ij} - u_i^1 - v_j^1) - L_1(u^1, v^1)(d_{ij} - u_i^2 - v_j^2)}{[L_2(u^2, v^2)]^2} + \\ + p_{ij} - u_i^3 - v_j^3, \quad i, j \in S, \tag{8.63}$$

где $\{u_i^1, u_i^2, u_i^3\}$ и $\{v_j^1, v_j^2, v_j^3\}$ – некоторые решения систем линейных уравнений (системы потенциалов) (8.61). Тогда критерием оптимальности опорного плана x задачи (8.56)–(8.60) является

Теорема 8.3. *Для того чтобы опорный план $x \in M(x)$ был решением задачи* (8.56)–(8.60), *достаточно существования таких значений переменных $\{u_i^1, u_i^2, u_i^3\}$ и $\{v_j^1, v_j^2, v_j^3\}$, удовлетворяющих равенствам* (8.61), *для которых $\Delta_{ij} \geq 0$ для всех i и j.*

1. Алгоритм метода потенциалов. Приведем метод потенциалов решения задачи (8.56)–(8.60), являющийся обобщением метода потенциалов решения задач линейного и дробно-линейного программирования транспортного типа. Так как рассматривается случай, когда $P(x) \leq 0$, то решение задачи (8.56)–(8.60) достигается в крайней точке многогранника $M(x)$.

Алгоритм метода потенциалов решения задачи (8.56)–(8.60) состоит в следующем.

1. Строим первоначальный опорный план задачи (8.56)–(8.60) методом "северо-западного угла" или минимального значения $c_{ij}/d_{ij} + p_{ij}$.

2. Находим значения потенциалов u и v решая системы уравнений (8.61).

3. Вычисляем значения $L_k(u^k, v^k)$, $\;k = 1,\, 2,\, 3$, по формуле (8.60).

4. Определяем значения выражения Δ_{rt} по формуле (8.63) для всех i, j, для которых $x_{ij} = 0$.

5. Находим $\Delta_{rt} = \min_{ij} \Delta_{ij}$.

6. Если $\Delta_{rt} \geq 0$, то опорный план x является решением задачи (8.56)–(8.60).

7. Если $\Delta_{rt} < 0$, то переходим к новому опорному плану x' задачи (8.56)–(8.60) по известным правилам линейного случая [601] и возвращаемся ко второму шагу алгоритма.

2. Функция Лагранжа и условия оптимальности. Так как $D(x) > 0$ для любого $x \in M(x)$, то для задачи (8.56)–(8.60) функцию Лагранжа можно построить следующим образом:

$$L(x,u,v) = \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} + \\ + \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij} + u_i + v_j)x_{ij} - \sum_{i=1}^{m} a_i u_i - \sum_{j=1}^{n} b_j v_j}{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}x_{ij}}, \tag{8.64}$$

где $u = (u_1, u_2, \ldots, u_m)$ и $v = (v_1, v_2, \ldots, v_n)$ являются множителями Лагранжа соответственно для ограничений (8.57) и (8.58).

Теорема 8.4. *Для того чтобы x^* было оптимальным решением задачи* (8.56)–(8.60)*, необходимо и достаточно существования таких значений переменных u^* и v^*, чтобы (x^*, u^*, v^*) была седловой точкой функции Лагранжа $L(x,u,v)$ (определенной по формуле* (8.64)*), при $x \geq 0$.*

Таким образом решение задачи (8.56)–(8.60) может быть сведено к нахождению седловой точки функции Лагранжа, т.е. к решению одной из следующих задач:

$$L(x^*, u^*, v^*) = \max_{u,v} \min_{x \geq 0} L(x,u,v) = \min_{x \geq 0} \max_{u,v} L(x,u,v).$$

3. Алгоритм параметрического метода. С помощью параметрического метода решение задачи (8.56)–(8.60) сводится к задаче параметрического программирования

$$\min_{x \in M(x)} \sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij} - \lambda d_{ij} + p_{ij})x_{ij}. \tag{8.65}$$

Алгоритм параметрического метода решения задачи (8.56)–(8.60) заключается в следующем.

0-й шаг. Берем $\lambda = \mu_0 = 0$ и находим оптимальное решение x_0^* задачи (8.65).

k-й шаг. Определяем значение параметра $\mu_k = \dfrac{C(x_{k-1}^*)}{D(x_{k-1}^*)}$ и находим оптимальное решение x_k^* линейной транспортной задачи (8.65) при $\lambda = \mu_k$. Если $F(x_k^*) = F(x_{k-1}^*)$, то план x_k^* является оптимальным решением задачи (8.56)–(8.60).

4. Алгоритмы метода покоординатного спуска. Рассмотрим частный случай задачи (8.56)–(8.60), когда ограничения (8.57) отсутствуют

$$F(x) = \frac{C(x)}{D(x)} + P(x) \to \min, \tag{8.66}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,n}), \tag{8.67}$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}), \tag{8.68}$$

при предположениях, что $D(x) > 0$ и $P(x) \leq 0$ для любого допустимого плана задачи (8.66)–(8.68).

Опорный план задачи (8.66)–(8.68) определяется по формуле: для каждого j положим

$$x_{ij} = \begin{cases} b_j, \text{для некоторого } & i = i_j, \\ 0, \ \text{для всех } i \neq j. \end{cases}$$

Для любого опорного плана определяем значения

$$\Delta_{ij} = \frac{D(x)\Delta_{ij}^1 - C(x)\Delta_{ij}^2}{D(x)\big(D(x) + b_j\Delta_{ij}^2\big)} + \Delta_{ij}^3 \tag{8.69}$$

для всех i и j, где

$$\Delta_{ij}^1 = c_{ij} - c_{i_j j}; \quad \Delta_{ij}^2 = d_{ij} - d_{i_j j}; \quad \Delta_{ij}^3 = p_{ij} - p_{i_j j}.$$

Тогда сформулируем следующий признак оптимальности опорного плана x задачи (8.66)–(8.68).

Теорема 8.5. *Если $\Delta_{ij} \geq 0$ для всех i и j, то опорный план x является оптимальным для задачи* (8.66)–(8.68).

Алгоритм метода покоординатного спуска для решения задачи (8.66)–(8.68) состоит в следующем.

1. Определяем первоначальный опорный план x методом минимального значения коэффициента p_{ij}, отношения c_{ij}/d_{ij} или выражения $p_{ij} + c_{ij}/d_{ij}$ в столбце.

2. Находим значения $C(x), D(x)$ и $P(x)$.

3. Вычисляем значения выражения Δ_{ij} по формуле (8.69) для всех i и j, для которых $x_{ij} = 0$.

4. Находим $\Delta_{kt} = \min\limits_{ij} \Delta_{ij}$.

5. Если $\Delta_{kt} \geq 0$, то план x является оптимальным решением задачи (8.66)–(8.68).

6. Если $\Delta_{kt} < 0$, то переходим к новому плану x' :

$$x'_{ij} = \begin{cases} b_t, & \text{для} \quad i = k;\ j = t, \\ 0, & \text{для} \quad i = k;\ j = l, \\ x_{ij}, & \text{для всех} \quad i \neq k;\ i \neq l;\ j \neq t, \end{cases}$$

где l – индекс, для которого $x_{lt} = b_t$.

7. Находим новые значения

$$C(x') = C(x) + b_k\Delta^1_{kt}, \quad D(x') = D(x) + b_k\Delta^2_{kt}, \quad P(x') = P(x) + b_k\Delta^3_{kt}$$

и переходим к третьему пункту алгоритма.

Поскольку $P(x) \leq 0$, то полученное решение задачи (8.66)–(8.68) по данному алгоритму может быть как локальным, так и глобальным минимумом.

Для решения задачи (8.66)–(8.68) можно использовать и некоторый приближенный одноитерационный алгоритм.

1. Определим первоначальный опорный план x задачи (8.66)–(8.68) по описанному выше правилу.

2. Положим $j = 1$.

3. Найдем значения $C(x), D(x)$ и $P(x)$.

4. Вычислим значения выражения

$$\Delta_{ij} = \frac{D(x)(c_{ij} - c_{kj}) - C(x)(d_{ij} - d_{kj})}{D(x)\big(D(x) + b_j(d_{ij} - d_{kj})\big)} + p_{ij} - p_{kj}$$

для всех $i \neq k$, где k индекс, для которого $x_{kj} = b_j$.

5. Найдем $\Delta_{tj} = \min\limits_{j} \Delta_{ij}$.

6. Если $\Delta_{tj} \geq 0$, то перейдем к девятому пункту алгоритма.

7. Если $\Delta_{tj} < 0$, то перейдем к новому плану x' :

$$x'_{ij} = \begin{cases} b_j, & \text{для} \quad i = t, \\ 0, & \text{для} \quad i = k, \\ x_{ij}, & \text{для всех} \quad i \neq k;\ i \neq t. \end{cases}$$

8. Найдем новые значения

$$C(x') = C(x) + b_j(c_{ij} - c_{kj}),$$

$$D(x') = D(x) + b_j(d_{ij} - d_{kj}),$$

$$P(x') = P(x) + b_j(p_{ij} - p_{kj}).$$

9. Положим $j = j+1$, и если $j \leq n$, то перейдем к четвертому пункту, в противном случае – конец работы алгоритма.

Кроме этих алгоритмов, для решения задачи (8.66)–(8.68) можно использовать алгоритм параметрического метода, который состоит в следующем.

1. Находим оптимальное решение x' задачи

$$\min_{x \in R(x)} \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} + p_{ij}) x_{ij}$$

следующим образом: для каждого j находим $i = k$, для которого

$$(c_{kj} + p_{kj}) = \min_i (c_{ij} + p_{ij}),$$

и полагаем

$$x^*_{ij} = \begin{cases} b_j, \text{для} & j = \overline{1,n};\ i = k, \\ 0, \ \text{для} & j = \overline{1,n};\ i \neq k. \end{cases}$$

2. Находим оптимальное решение x^* задачи

$$\min_{x \in R(x)} \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} - \mu d_{ij} + p_{ij}) x_{ij},$$

в которой $\mu = \dfrac{C(x')}{D(x')}$ по формуле

$$x^*_{ij} = \begin{cases} b_j, \text{для} & j = \overline{1,n};\ i = k; \\ 0, \ \text{для}\, j = \overline{1,n};\ i \neq k, \end{cases}$$

где k значения индексов i $(i = \overline{1,m})$, для которых

$$(c_{kj} - \mu\, d_{kj} + p_{kj}) = \min_i (c_{ij} - \mu\, d_{ij} + p_{ij}).$$

3. Если $F(x^*) = F(x')$, то план x^* является оптимальным решением и для задачи (8.66)–(8.68).

4. Если $F(x^*) \neq F(x')$, то $x' = x^*$ и переходим ко второму пункту алгоритма.

5. Алгоритм субградиентного метода. Рассмотрим теперь алгоритм решения задачи (8.56)–(8.59) для случая, когда $\sum_{i=1}^{m} a_i \geq \sum_{j=1}^{n} b_j$ и количество потребителей n намного больше чем количество поставщиков m. Тогда задачу (8.56)–(8.59) запишем в виде:

$$F(x) = \frac{C(x)}{D(x)} + P(x) \to \min, \tag{8.70}$$

$$\sum_{j=1}^{n} x_{ij} \leq a_i \quad (i = \overline{1,m}), \tag{8.71}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,n}), \tag{8.72}$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}), \tag{8.73}$$

при предположениях, что $C(x) > 0$ и $P(x) \leq 0$ для любого допустимого плана задачи (8.70)–(8.73).

Построим функцию Лагранжа для задачи (8.70)–(8.73) по формуле

$$L(x,u) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij} - u_i)x_{ij} - \sum_{i=1}^{m} a_i u_i}{D(x)} + P(x),$$

где $u = (u_1, u_2, \ldots, u_m)$ – множители Лагранжа для ограничений (8.71).

Тогда (8.70)–(8.73) сводится к решению задачи

$$\max_{u \geq 0} L^*(u), \tag{8.74}$$

где

$$L^*(u) = \min_{x \in R(x)} L(x,u). \tag{8.75}$$

Здесь $R(x)$ – ограниченное выпуклое многогранное множество значений переменных x, удовлетворяющих ограничениям (8.72), (8.73)

Рассмотрим следующий алгоритм схемы декомпозиции по ограничениям для решения задачи (8.70)–(8.73).

1. Решаем задачу максимизации функции $L^*(u)$ при ограничениях $u \geq 0$. Тогда на t-м шаге субградиентного метода необходимо выполнить основные три этапа вычислений:

- решить задачу дробно-линейного программирования (8.75) при фиксированных значениях $u = u^t$ по одному из алгоритмов параметрического метода или метода покоординатного спуска;
- вычислить значения обобщенного градиента функции $L^*(u)$ в точке $u = u^t$ по формуле

$$g_i(u^t) = \left(\sum_{i \in M_i} b_j - a_i \right) / D(x^*) \quad (i = \overline{1, m}),$$

  где M_i – множество значений индексов j, для которых $x_{ij} = b_j$ в оптимальном решении задачи (8.75). Используя множества M_i оптимальное решение x^* представим следующим образом:

$$x^* = \{x^*_{ij} \colon x^*_{ij} = b_j,\ j \in M_i; \quad x^*_{ij} = 0,\ j \notin M_i,\ i = \overline{1, m}\};$$

- найти новые значения

$$u_i^{t+1} = \max\left\{0, u_i^t + h_{t+1} \cdot g_i(u^t)\right\} \quad (i = \overline{1, m}),$$

  где h_{t+1} – величина шага.

2. Пусть u^* – оптимальное решение задачи (8.74), а x^* – оптимальное решение задачи (8.75) при фиксированных $u = u^*$. Тогда для каждого j и для всех $i \neq k$ находим значения

$$\Delta_{ij} = \frac{D(x)(c_{kj} + u_k^* - c_{ij} - u_i^*) - C(x)(d_{kj} - d_{ij}))}{D(x)\big(D(x) + b_j(d_{kj} - d_{ij})\big)} + p_{ij} - p_{kj},$$

где k – индекс, для которого $j \in M_k$.

3. Находим множество J тех значений индексов j, для которых имеет место выражение $|\Delta_{ij}| < \varepsilon$ хотя бы для одного значения индекса $i \neq k$, где $\varepsilon > 0$ – достаточно малое число.

4. Находим множество I тех значений индексов i, для которых коэффициенты a'_i, определенные по формуле

$$a'_i = a_i - \sum_{j \notin J} x^*_{ij},$$

положительны.

5. По алгоритму метода потенциалов или параметрического метода решаем задачу:

$$\frac{\sum\limits_{i\in I}\sum\limits_{j\in J} c_{ij}x_{ij} + c_0'}{\sum\limits_{i\in I}\sum\limits_{j\in J} d_{ij}x_{ij} + d_0'} + \sum_{i\in I}\sum_{j\in J} p_{ij}x_{ij} + p_0' \to \min;$$

$$\sum_{j\in J} x_{ij} \le a_i', \quad i \in I;$$

$$\sum_{i\in I} x_{ij} = b_j, \quad j \in J;$$

$$x_{ij} \ge 0, \quad i \in I;\ j \in J,$$

где

$$c_0' = \sum_{i\notin I}\sum_{j\notin J} c_{ij}x_{ij}^*, \quad d_0' = \sum_{i\notin I}\sum_{j\notin J} d_{ij}x_{ij}^*, \quad p_0' = \sum_{i\notin I}\sum_{j\notin J} p_{ij}x_{ij}^*.$$

8.5. Дробно-линейная транспортная задача с дополнительными ограничениями

Рассмотрим следующую задачу

$$F(x) = \frac{C(x)}{D(x)} = \frac{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} c_{ij}x_{ij}}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} d_{ij}x_{ij}} \to \min, \tag{8.76}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^r x_{ij} \le \beta_r \quad (r = \overline{1,k}), \tag{8.77}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}), \tag{8.78}$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = \overline{1,n}), \tag{8.79}$$

$$x_{ij} \ge 0 \quad (i = \overline{1,m};\ j = \overline{1,n}). \tag{8.80}$$

Задача (8.76)–(8.80) представляет собой дробно-линейную транспортную задачу с дополнительными ограничениями (8.77). Несмотря на то, что в задаче (8.76)–(8.80) присутствуют транспортные ограничения (8.78)–(8.80), наличие ограничений (8.77) выводит ее из класса транспортных задач и для ее решения не могут быть использованы эффективные алгоритмы типа метода потенциалов. Поэтому задачу (8.76)–(8.80) рассматривают как общую задачу дробно-линейного программирования и для ее решения применяют известные методы (симплекс-метод, модифицированный симплекс-метод, параметрический метод и т.п.).

Однако, для решения задачи (8.76)–(8.80) могут быть построены и другие алгоритмы. Рассмотрим алгоритм схемы декомпозиции по ограничениям с применением субградиентных методов и алгоритм декомпозиции Данцига-Вулфа.

1. Схемы декомпозиции по ограничениям. Построим функцию Лагранжа задачи (8.76)–(8.80) на множестве ограничений (8.77) по формуле

$$L(x,y)=\frac{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n}c_{ij}x_{ij}+\sum\limits_{r=1}^{k}y_r\left(\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n}f_{ij}^{r}x_{ij}-\beta_r\right)}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n}d_{ij}x_{ij}},$$

где $y_r=(y_1,\,y_2,\ldots,y_k)$ – множители Лагранжа для ограничений (8.77), $y\geq 0$. Обозначим через $M(x)$ множество значений переменных x, удовлетворяющих ограничениям (8.78)–(8.80), т.е. $M(x)$ является выпуклым, ограниченным транспортным многогранником.

Рассмотрим задачи

$$\max_{y\geq 0}L^{*}(y), \tag{8.81}$$

где

$$L^{*}(y)=\min_{x\in M(x)}L(x,y). \tag{8.82}$$

Тогда решение задачи (8.76)–(8.80) по схеме декомпозиции по ограничениям сводится к решению задачи (8.81) субградиентным методом.

1. Решаем задачу (8.81) субградиентным методом. Тогда на $(t+1)$-м шаге необходимо:

а) решить задачу дробно-линейного программирования транспортного типа (8.82) при фиксированных значениях $v=v^t$ по одному из алгоритмов параметрического метода или метода потенциалов при фиксированных значениях $y=y^t$;

б) вычислить значения обобщенного градиента функции $L^*(y)$ в точке $y = y^t$ по формуле

$$g_i(y_r^t) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}^r x_{ij}^*(y^t) - \beta_r}{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} x_{ij}^*(y^t)} \quad (r = \overline{1,k}),$$

где $x_{ij}^*(y^t)$ – решение задачи (8.82) при фиксированных $y = y^t$;

в) найти новые значения

$$y_r^{t+1} = \max\left\{0,\, y_r^t + h_{t+1} \cdot g_r(y_r^t)\right\} \quad (r = \overline{1,k}),$$

где h_{t+1} – величина шага.

Следует отметить, что так как задача (8.82) решается неоднократно в итерационном процессе субградиентного метода, то в алгоритме параметрического метода достаточно решить только одну задачу типа (8.8). Пусть на предыдущем шаге итерационного метода найдено решение x_{t-1}^* транспортной задачи (8.8) при фиксированных λ. Тогда на t-м шаге решаем одну транспортную задачу (8.82) при $\lambda = \lambda_k = L(x_{t-1}^*, y^t)$.

2. Алгоритм декомпозиции Данцига-Вулфа. Обозначим через $\{x_{ij}^l\}$ $(l = \overline{1,s})$, координаты всех крайних точек многогранника $M(x)$. Тогда любая точка $x \in M(x)$ может быть представлена в виде

$$x_{ij} = \sum_{l=1}^{s} \lambda_l x_{ij}^l \quad (i = \overline{1,m};\ j = \overline{1,n}),$$

$$\sum_{l=1}^{s} \lambda_l = 1, \quad \lambda_l \geq 0 \quad (l = \overline{1,s}).$$

Тогда задача (8.76)–(8.80) сводится к следующей координирующей задаче:

$$\frac{P(x,\lambda)}{Q(x,\lambda)} = \frac{\sum_{l=1}^{s} p_l \lambda_l}{\sum_{l=1}^{s} q_l \lambda_l} \to \min, \tag{8.83}$$

$$\sum_{l=1}^{s} t_l^r \lambda_l \leq b_r \quad (r = \overline{1,k}), \tag{8.84}$$

$$\sum_{l=1}^{s} \lambda_l = 1, \tag{8.85}$$

$$\lambda_l \geq 0 \quad (l = \overline{1, s}), \tag{8.86}$$

где

$$p_l = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}^l \quad (l = \overline{1, s}),$$

$$q_l = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij}^l \quad (l = \overline{1, s}),$$

$$t_l^r = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}^r x_{ij}^l \quad (l = \overline{1, s};\ r = \overline{1, k}).$$

Задача (8.83)–(8.86) является задачей дробно-линейного программирования, для решения которой используем модифицированный симплекс-метод.

Рассмотрим алгоритм декомпозиции Данцига-Вулфа для решения исходной задачи (8.22)–(8.25).

Пусть B – текущий базис, B^{-1} – соответствующая обратная матрица, λ_B – значения переменных λ_l в данном базисе, $P(x_B, \lambda_B)$ и $Q(x_B, \lambda_B)$ – соответственно значения числителя и знаменателя в текущем базисе, а x_B – крайние точки $\{x_{ij}^l\}$, соответствующие базисным переменным $\lambda_B = \{\lambda_l\}$ Тогда:

1) находим значения двойственных оценок ограничений (8.84) и (8.85) соответственно для числителя $u = (u_1, u_2, \ldots, u_k, \overline{u})$ и знаменателя $v = (v_1, v_2, \ldots, v_k, \overline{v})$ по формулам

$$u = p_B B^{-1} \ \text{ и } \ v = q_B B^{-1},$$

где p_B и q_B – векторы коэффициентов числителя и знаменателя при базисных переменных λ_B;

2) находим очередную крайнюю точку $\{x_{ij}^l\}$ многогранника $M(x)$, в которой достигается минимальное значение функционала

$$G(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} g_{ij} x_{ij} + g_0$$

где $g_{ij} = Q(x_B, \lambda_B)\Big(c_{ij} - \sum\limits_{r=1}^{k} u_r f_{ij}^r\Big) - P(x_B, \lambda_B)\Big(d_{ij} - \sum\limits_{r=1}^{k} v_r f_{ij}^r\Big)$; $g_0 = Q(x_B, \lambda_B)\overline{u} - P(x_B, \lambda_B)\overline{v}$, при исходных ограничениях (8.78)–(8.80), т.е. решаем обычную линейную транспортную задачу одним из известных методов.

Пусть $x^l = \{x_{ij}^l\}$ – оптимальное решение данной задачи транспортной задачи.

3) если $G(x^l) \geq 0$, то текущий базис является оптимальным.

4) пусть $G(x^l) < 0$. Тогда вектор l вводится в базис, для чего находятся значения коэффициентов p_l, q_l и t_{jl} по соответствующим формулам.

Пусть $\{\lambda_k^*\}$ – оптимальное решение задачи (8.83)–(8.86), тогда оптимальное решение для исходной задачи (8.76)–(8.80) определяется по формуле

$$x_{ij}^* = \sum_{l=1}^{k+1} \lambda_l^* x_{ij}^l \quad (i = \overline{1,m};\ j = \overline{1,n}),$$

где $\{x_{ij}^k\}$ – координаты крайних точек многогранника $M(x)$, соответствующие базисным переменных λ_l^* оптимального базиса B.

Таким образом решение дробно-линейной транспортной задачи с дополнительными ограничениями (8.76)–(8.80) сводится к задачи дробно-линейного программирования (8.83)–(8.86) с уменьшенным количеством ограничений, специальным правилом выбора вектора для ввода в базис и новым критерием оптимальности.

8.6. Производственно-транспортная задача дробного программирования с неизвестными мощностями

Рассмотрим следующую производственно-транспортную задачу с дробным функционалом

$$F(x,y,z) = \frac{F_1(x,y,z)}{F_2(x,y,z)} = \frac{C(x) + \varphi(y) + P(z)}{D(x) + \psi(y) + Q(z)} \to \min, \tag{8.87}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}), \tag{8.88}$$

$$\sum_{i=1}^{m} f_{ij} x_{ij} = y_j \quad (j = \overline{1,n}), \tag{8.89}$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}), \tag{8.90}$$

$$\sum_{l=1}^{k} z_{jl} = y_j \quad (j = \overline{1,n}), \tag{8.91}$$

$$\sum_{j=1}^{n} z_{jl} = b_l \quad (l = \overline{1,k}), \tag{8.92}$$

$$z_{jl} \geq 0 \quad (j = \overline{1,n};\ l = \overline{1,k}), \tag{8.93}$$

в которой

$$C(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}, \quad D(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}x_{ij},$$

$$P(z) = \sum_{j=1}^{n}\sum_{l=1}^{k} p_{jl}z_{jl}, \quad Q(z) = \sum_{j=1}^{n}\sum_{l=1}^{k} q_{jl}z_{jl},$$

$$\varphi(y) = \sum_{j=1}^{n} \varphi_j(y_j), \quad \psi(y) = \sum_{j=1}^{n} \psi_j(y_j),$$

а функции $\varphi_j(y_j)$ и $-\psi_j(y_j)$ являются выпуклыми, $\{f_{ij}\}$, $\{c_{ij}\}$, $\{d_{ij}\}$, $\{p_{jl}\}$, $\{q_{jl}\}$, $\{a_i\}$, $\{b_l\}$ – заданные числа, $\{x_{ij}\}$, $\{z_{jl}\}$, $\{y_j\}$ – переменными (неизвестными) величинами.

Задача (8.87)–(8.93) является задачей дробного программирования, в которой имеются линейные ограничения, дробно-выпуклый функционал и три группы переменных, определяющие план перевозки сырья $\{x_{ij}\}$, мощности производства $\{y_j\}$ и план перевозки готовой продукции $\{t_{jl}\}$.

Для решения задачи (8.87)–(8.93) можно использовать схемы декомпозиции по переменным и субградиентные методы.

Пусть переменные y_j фиксированы и принимают значение $\overline{y}_j$. Тогда получим задачу

$$\Phi(\overline{y}) = \min_{x \in M(x,z,\overline{y})} F(x, z, \overline{y}), \tag{8.94}$$

в которой $M(x, z, \overline{y})$ – множество допустимых решений (x, y), удовлетворяющие ограничениям (8.89)–(8.93) при фиксированном $y = \overline{y}$. Тогда решение задачи (8.87)–(8.93) сводится к задаче

$$\min_{y \geq 0} \Phi(y). \tag{8.95}$$

Функция $\Phi(y)$ является кусочно-линейной и не везде дифференцируемой, для минимизации которой используем субградиентные методы.

Тогда имеем двухуровневую схему решения задачи (8.87)–(8.93), на внешнем уровне которой решается задача (8.95), а на внутреннем уровне – задача (8.94).

Пусть для решения задачи (8.95) применяется один из субградиентным методов, на t-м шаге, которого имеем значения y^t и решение (x^t, z^t) задачи (8.94) при фиксированных $y = y^t$. Тогда на очередном, $(t+1)$-м шаге, необходимо выполнить следующие три основных этапа.

1. Решить при $\overline{y} = y^t$ задачу дробной оптимизации (8.94). Пусть $(x^*(y^t), z^*(y^t))$ – решение задачи (8.94), которая является задачей дробно-линейного программирования следующего вида

$$W(x,z) = \frac{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} c_{ij}x_{ij} + \sum\limits_{j=1}^{n}\sum\limits_{l=1}^{k} p_{jl}z_{jl} + \varphi(y^t)}{\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} d_{ij}x_{ij} + \sum\limits_{j=1}^{n}\sum\limits_{l=1}^{k} q_{jl}z_{jl} + \psi(y^t)}, \tag{8.96}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}), \tag{8.97}$$

$$\sum_{i=1}^{m} f_{ij}x_{ij} = y_j^t \quad (j = \overline{1,m}), \tag{8.98}$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}), \tag{8.99}$$

$$\sum_{l=1}^{k} z_{jl} = y_j^t \quad (j = \overline{1,n}), \tag{8.100}$$

$$\sum_{j=1}^{n} z_{jl} = b_l \quad (l = \overline{1,k}), \tag{8.101}$$

$$z_{jl} \geq 0 \quad (j = \overline{1,n};\ l = \overline{1,k}). \tag{8.102}$$

Дробно-линейная задача (8.96)–(8.102) имеет ограничения разделенными на две независимые группы переменных, связанных между собой значениями $\{y_j^t\}$, которые переопределяются на каждом шаге субградиентного метода. Если для ее решения применить параметрический метод (в частности одноитерационный), то получим две отдельные задачи. Первая из них это распределительно-транспортная задача для определения плана перевозке и переработке сырья, а вторая – транспортная задача определения плана перевозки готовой продукции.

Если в параметрическом методе решения дробных задач выбрать $\lambda = \mu_t = W(x^t, z^t)$, то получим следующие две задачи:

– распределительная задача линейного программирования транспортного типа

$$\sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij} - \mu_t d_{ij})x_{ij} \to \min, \tag{8.103}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = \overline{1,m}), \tag{8.104}$$

$$\sum_{i=1}^{m} f_{ij}x_{ij} = y_j^t \quad (j = \overline{1,n}), \tag{8.105}$$

$$x_{ij} \geq 0 \quad (i = \overline{1,m};\ j = \overline{1,n}); \tag{8.106}$$

– обычная линейная транспортная задача

$$\sum_{j=1}^{n}\sum_{l=1}^{k}(p_{jl} - \mu_t q_{jl})z_{jl} \to \min, \tag{8.107}$$

$$\sum_{l=1}^{k} z_{jl} = y_j^t \quad (j = \overline{1,n}), \tag{8.108}$$

$$\sum_{j=1}^{n} z_{jl} = b_l \quad (l = \overline{1,k}), \tag{8.109}$$

$$z_{jl} \geq 0 \quad (j = \overline{1,n};\ l = \overline{1,k}). \tag{8.110}$$

Пусть $x(y^t)$ – решение задачи (8.103)–(8.106), а $\{u_i^t\}$ и $\{v_j^t\}$ значения потенциалов (двойственные оценки) для ограничений (8.104) и (8.105) соответственно. Также допустим, что $z(y^t)$ является решением задачи (8.107)–(8.110), а $\{\alpha_j^t\}$ и $\{\beta_l^t\}$ – значения потенциалов (двойственные оценки) для ограничений (8.108) и (8.109) соответственно.

Следует отметить, что при фиксированных значениях y_j^t транспортная задача (8.107)–(8.110) может не иметь допустимые решения, так как может быть, что $\sum\limits_{j=1}^{n} y_j^t \neq \sum\limits_{l=1}^{k} b_l$, т.е. транспортная задача не является сбалансированной (закрытой). Поэтому, для выхода из этой ситуации вводится дополнительный (искусственный) поставщик с номером $n+1$, для которого $y_{n+1} = \left| \sum\limits_{l=1}^{k} b_l - \sum\limits_{j=1}^{n} y_j^t \right|$, а $p_{n+1,l} = M$ и $q_{n+1,l} = 0 \ \ (l = \overline{1,k})$, где M – достаточно большое положительное число.

Задача (8.103)–(8.106), также может быть неразрешимой. Поэтому для этой задачи вводим дополнительный потребитель с номером $n+1$ для которого полагаем объем потребления $a_{n+1} = \sum\limits_{i=1}^{m} a_i$, а $c_{i,n+1} = 0$ и $d_{i,n+1} = 0 \quad (i = \overline{1,m})$.

2. Найти значения субградиента функции $\Phi(y)$ в точке $y = y^t$. Для этого используем функцию Лагранжа, которая соответствует параметрической задачи, используемой в методе решения дробной задачи (8.87)–(8.93) с учетом значений переменных x и y, полученных на $(t+1)$-м шаге субградиентного метода.

Рассмотрим функцию Лагранжа задачи (8.87)–(8.93) в параметрическом методе, которая при $\lambda = \mu_t = W(x^t, z^t)$ имеет следующий вид

$$L(x,y,z,u,v,\alpha,\beta) = \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} - \mu_t d_{ij} + u_i^t + f_{ij} v_j^t) x_{ij} -$$

$$- \sum_{i=1}^{m} a_i u_i^t - \sum_{j=1}^{n} y_j^t \cdot v_j^t + \sum_{j=1}^{n} \sum_{l=1}^{k} (p_{jl} - \mu_t q_{jl} + \alpha_j^t + \beta_l^t) z_{jl} -$$

$$- \sum_{j=1}^{n} y_j^t \alpha_j^t - \sum_{l=1}^{k} b_l \beta_l^t + \sum_{j=1}^{n} \bigl(\varphi_j(y_j^t) - \mu_t \psi_j(y_j^t) \bigr).$$

Тогда обобщенный градиент функции $\Phi(y)$ в точке $y = y^t$ определяется по формуле

$$g_j(y_j^t) = g_j^{\varphi}(y_j^t) - \mu_k g_j^{\psi}(y_j^t) - v_j^t - \alpha_j^t \quad (j = \overline{1,n}),$$

где $g_j^{\varphi}(y_j^t)$ и $g_j^{\psi}(y_j^t)$ – соответственные градиенты (субградиенты) функций $\varphi_j(y_j)$ и $\psi_j(y_j)$, вычисленные в точке $y_j = y_j^t$.

3. Определить новые значения для переменных y_j по формуле

$$y_j^{t+1} = \max \bigl\{ 0,\, y_j^t - h_{t+1} \cdot g_j(y_j^t) \bigr\} \quad (j = \overline{1,n}),$$

где h_{t+1} – величина шага в субградиентном методе.

Таким образом, для решения производственной транспортной задачи (8.87)–(8.93) можно использовать схему декомпозиции по переменным с применением субградиентного метода, на каждом шаге которого решаются две линейные задачи транспортного типа.

Литература

[1] *Abadie J.M., Williams A.C.* Dual and parametric methods in decomposition. - Recent Adcances in Mathematical Programming. - McGraw-Hill, Inc., New York, 1963.

[2] *Abrham J., Luthra S.* Comparison of duality models in fractional linear programming. - Mathematical Method of Operations Research, 1977, vol. 21, nr. 3, p. 125 - 130.

[3] *Aggarwal S.P.* Analysis of the solution to a linear fractional functional programming. - Metrika, 1970, vol. 16, nr. 1, p. 9 - 26.

[4] *Aggarwal S.P.* Parametric linear fractional functional programming. - Metrika, 1968, vol. 12, nr. 2 - 3, p. 106 - 114.

[5] *Aggarwal S.P.* Quadratic fractional functionals programming. - Cahiers Centre Etudes Recherche Oper., 1973, vol. 15, nr. 2, p. 157 - 165.

[6] *Aggarwal S.P.* Variation in parameters of a quadratic fractional functionals programming. - Rev. Belge Statist. Informat. Rech. Operat., 1972, vol. 11, nr. 4, p. 3 - 12.

[7] *Aggarwal S.P., Saxena P.C.* Duality theorems for fractional functional programming with quadratic constraint. - Ekonom.-Mat. Obzor, 1974, vol. 10, nr. 1, p. 86 - 92.

[8] *Aggarwal S.P., Saxena P.C.* Duality theorems for non-linear fractional programs. - Z. Angew. Math. Mech., 1975, vol. 55, p. 523 - 525.

[9] *Aggarwal S.P., Verma R.K.* A suboptimization method for the sum of linear and linear fractional interval programming problems. - Acta Cienc. Indica Math., 1981, vol. 7, nr. 1, p. 14 - 24.

[10] *Aggarwal S.P., Verma R.K.* On the solutions of the sum of linear and linear fractional interval programming problems. - Pure Appl. Math. Sci., 1983, vol. 17, nr. 1 - 2, p. 73 - 81.

[11] *Aggarwal U., Swarup K., Garg K.C.* Goal programming problem with linear fractional objective functions. - Cahiers Centre Etudes Recherche Oper., 1984, vol. 26, nr. 1 - 2, p. 33 - 41.

[12] *Ahmad I., Husain Z.* Duality in nondifferentiable minimax fractional programming with generalized convexity. - Appl. Math. Comp., 2006, vol. 176, nr. 2, p. 545 - 551.

[13] *Ahmad I., Husain Z.* Optimality conditions and duality in nondifferentiable minimax fractional programming with generalized convexity. - Journal of Optimization Theory and Applications, 2006, vol. 129, nr. 2, p. 255 - 275.

[14] *Alizadeh F.* Interior point methods in semidefinite programming with applications to combinatorial optimization. - SIAM Journal on Optimization, 1995, vol. 5, nr. 1, p. 13 - 51.

[15] *Almogy Y., Levin O.* A class of fractional programming problems. - Operations Research, 1971, vol. 19, nr.1, p. 57 - 67.

[16] *Anand P.* Decomposition procedure for linear fractional programs with upper bounds. - ZAMM, 1973, vol. 53, p. 635 - 636.

[17] *Anand P.* Dual and parametric methods in decomposition for linear fractional program. - Studia Scientarium Mathematicarum Hungarica, 1971, nr. 6, p. 267 - 275.

[18] *Andrei N.* Metode de punct interior in programarea convexa. - Bucuresti, Editura MATRIXROM, 2000.

[19] *Andrei N.* Optimizare fara restrictii. Metode de directii conjugate. - Bucuresti, Editura MATRIXROM, 2000.

[20] *Andrei N.* Programarea matematica avansata. Teorie, Metode Computationale, Aplicatii. - Bucuresti, Editura Tehnica, 1999.

[21] *Andrei N.* Programarea matematica. Metode de punct interior. - Bucuresti, Editura Tehnica, 1999.

[22] *Anstreicher K.M.* A combined phase I-phase 2 projective algorithm for linear programming. - Mathematical Programming, 1989, vol. 43, p. 209 - 223.

[23] *Anstreicher K.M.* A combined phase I-phase 2 scaled potential algorithm for linear programming. - Mathematical Programming, 1991, vol. 52, p. 429 - 498.

[24] *Anstreicher K.M.* A monotonic projective algorithm for fractional linear programming. - Algorithmica, 1986, vol. 1, nr. 4, p. 483 - 498.

[25] *Anstreicher K.M.* A worst-case step in Karmarkar's algorithm. - Mathematics of Operations Research, 1989, vol. 14, p. 294 - 302.

[26] *Anstreicher K.M., Watteyne P.* A family of search directions for Karmarkar's algorithm. - Operations Research, 1993, vol. 41, p. 759 - 767.

[27] *Antczak T.* Generalized fractional minimax programming with B-(p,r)-invexity. - Computers Mathematics with Applications, 2008, vol. 56, nr. 6, p. 1505 - 1525.

[28] *Arsham H., Kahn A.B.* A complete algorithm for linear financial planning. - J. Oper. Res. Soc., 1990, vol. 20, p. 11 - 23.

[29] *Atkinson D.S., Vaidya P.M.* A cutting plane algorithm for convex programming that uses analytic centers. - Mathematical Programming, 1995, vol. 69, p.1 - 43.

[30] *Avriel M., Diewert W.E., Schaible S., Zang I.* Generalized concavity. - Plenum Press: New York, 1988.

[31] *Bachem A., Grotschel M., Korte B. (Eds)* Mathematical programming. The state of art. - Berlin: Springer-Verlag, 1983.

[32] *Bajalinov E.B.* Linear-fractional programming: theory, methods, applications and software. - Dordrecht: Kluwer Academic Publishers, 2003.

[33] *Balinski M.L., Wolfe P. (Eds)* Nondifferentiable optimization. - Mathematical Programming Study, vol. 3. - Amsterdam: North - Holland, 1975.

[34] *Bancompte M., Martinez-Legaz J.E.* Fractional programming by lower subdifferentiability techniques. - Journal of Optimization Theory and Applications, 1991, vol. 68, p. 95 - 116.

[35] *Barnes E.R.* A variation on Karmarkar's algorithm for linear programming. - Mathematical Programming, 1986, vol. 36, nr. 2, p. 174 - 182.

[36] *Barros A.I.* Discrete and fractional programming techniques for location models. - Kluwer Academic Publishers, Dordrecht/Boston/London, 1998.

[37] *Barros A.I., Dekker R., Frenk J.B.G., Weeren S.* Optimizing a general optimal replacement model by fractional programming techniques. - Journal of Global Optimization, 1997, vol. 10, nr. 4, p. 405 - 423.

[38] *Barros A.I., Frenk J.B.G.* Generalized fractional programming and cutting plane algorithms. - Journal of Optimization Theory and Applications, 1995, vol. 87, nr. 1, p. 103 - 120.

[39] *Barros A.I., Frenk J.B.G., Gromicho J.* Fractional location problems. - Location Science, 1997, vol. 5, nr. 1, p. 47 - 58.

[40] *Barros A.I., Frenk J.B.G., Schaible S., Zhang S.* A new algorithm for generalized fractional programs. - Mathematical Programming, 1996, vol. 72, nr. 2, p. 147 - 175.

[41] *Barros A.I., Frenk J.B.G., Schaible S., Zhang S.* Using duality to solve generalized fractional programming problems. - Journal of Global Optimization, 1996, vol. 8, nr. 2, p. 139 - 170.

[42] *Bayer D.A., Lagarias J.C.* Karmarkar's algorithm and Newton's method. - Mathematical Programming, 1991, vol. 50, p. 291 - 330.

[43] *Bazaraa M.S., Jarvis J.J., Sherali H.D.* Linear programming and network flows. - John Wiley and Sons, New York, NY, 1990.

[44] *Bazaraa M.S., Sherali H.D., Shetty C.M.* Nonlinear programming. Theory and algorithms. - John Wiley and Sons, New York, NY, 1993.

[45] *Beck A., Ben-Tal A., Teboulle M.* Finding a global optimal solution fot a quadratically constrained fractional quadratic problem with applications to the regularized total least squares. - SIAM Journal on Matrix Analysis and Applications, 2006, vol. 28, p. 425 - 445.

[46] *Bector C.R.* Duality in linear fractional programming. - Utilitas Math., 1974, vol. 4, p. 155 - 168.

[47] *Bector C.R.* Duality in nonlinear fractional programming. - Zeitschrift fur Operations Research, 1973, vol. 17, nr. 5, p. 183 - 193.

[48] *Bector C.R.* Nonlinear fractional functional programmming with nonlinear constraints. - ZAMM, 1968, vol. 48, p. 284 - 286.

[49] *Bector C.R.* On convexity, pseudo convexity and quasi-convexity of composite functions. - Cahiers Centre Etudes Recherche Oper., 1973, vol. 15, nr. 4, p. 411 - 428.

[50] *Bector C.R.* Programming problems with convex fractional functions. - Operations Research, 1968, vol. 16, nr. 2, p. 383 - 391.

[51] *Bector C.R., Cambini A.* Fractional programming - some recent results. - Lecture Notes in Economics and Mathematical Systems. - Berlin: Springer-Verlag, 1990, vol. 345, p. 86 - 98.

[52] *Bector C.R., Chandra S., Bector M.K.* Generalized fractional programming duality: a parametric approach. - Journal of Optimization Theory and Applications, 1989, vol. 60, nr. 2, p. 243 - 260.

[53] *Bector C.R., Chandra S., Husain I.* Generalized continuous fractional programming duality: a parametric approach. - Util. Math. 1992, vol. 42, p. 39 - 60.

[54] *Bector C.R., Chandra S., Husain I.* Optimality conditions and duality in subdifferentiable multiobjective fractional programming. - Journal of Optimization Theory and Applications, 1993, vol. 79, nr. 1, p. 105 -125.

[55] *Bector C.R., Chandra S., Kumar V.* Fenchel duality in generalized fractional programming. - Generalized convexity and generalized monotonicity. - Kluwer Academic Publishers, Dordrecht, 1998, p. 373 - 386.

[56] *Bector C.R., Chandra S., Singh C.* Duality on multiobjective fractional programming. - Lecture Notes in Economics and Mathematical Systems, - Berlin: Springer-Verlag, 1990, vol. 345, p. 232 - 241.

[57] *Bector C.R., Kumar U.* Duality for multiple-objective fractional programs. - Cahiers Centre Etudes Recherche Oper., 1984, vol. 26, nr. 3 - 4, p. 201 - 207.

[58] *Bector C.R., Suneja S.K.* Duality in nondifferentiable generalized fractional programming. - Asia Pacific Journal of Operations Research, 1988, vol. 5, nr. 2, p. 134 - 139.

[59] *Bector M., Husain I., Chandra S., Bector C.R.* A duality model for a generalized minimax program. - Naval Res. Logist., 1988, vol. 35, nr. 5, p. 493 - 501.

[60] *Bellman R.E., Dreyfus S.* Applied dynamic programming. - Princeton University Press, Princeton, NJ, 1962.

[61] *Benders J.F.* Partitioning procedures for solving mixed variables programming problems. - Numerische Mathematik, 1962, vol. 4, p. 238 - 252.

[62] *Benadada Y., Crouzeix J.-P., Ferland J.A.* An interval-type algorithm for generalized fractional programming. - Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, 1990, vol. 345, p. 106 - 120.

[63] *Benadada Y., Ferland J.A.* Partial linearization for generalized fractional programming. - Zeitschrift fur Operations Research, 1988, vol.32, nr. 2, p. 101 - 106.

[64] *Benson H.P.* A simplicial branch and bound duality-bounds algorithm for the linear sum-of-ratios problem. -Eur. J. Oper. Res., 2007, vol. 182, nr. 2, p. 597-611.

[65] *Benson H.P.* An outcome space branch and bound - outer approximation algorithm for convex multiplicative programming. - Journal of Global Optimization, 1999, vol. 15, p. 315 - 342.

[66] *Benson H.P.* Fractional programming with convex quadratic forms and functions. - European Journal of Operational Research, 2006, vol. 173, p. 351 - 369.

[67] *Benson H.P.* Global optimization algorithm for the nonlinear sum of ratios problem. - Journal of Optimization Theory and Applications, 2002, vol. 112, nr.1, p. 1 - 29.

[68] *Benson H.P.* Global optimization of nonlinear sums of ratios. - Journal of Mathematical Analysis and Applications, 2001, vol. 263, nr.1, p. 301 - 315.

[69] *Benson H.P.* On the global optimization of sums of nonlinear fractional functions over a convex set. - J. Optim. Theory Appl., 2004, vol. 121, nr. 1, p. 19 - 39.

[70] *Benson H.P.* Solving sum of ratios fractional programs via concave minimization. - J. Optim. Theory Appl., 2007, vol. 135, nr. 1, p. 1 - 17.

[71] *Benson H.P.* Using concave envelopes to globally solve the nonlinear sum of ratios problem. - Journal of Global Optimization, 2002, vol. 22, nr. 1 - 4, p. 343 - 364.

[72] *Benson H.P., Boger G.M.* Multiplicative programming problems: analysis and efficient point search heuristic. - Journal of Optimization Theory and Aplications, 1997, vol. 94, p. 487 - 510.

[73] *Berkelaar A., Dert C., Oldenkamp B., Zhang S.* A primal-dual decomposition-based interior point approach to two-stage stochastic linear programming. - Operations Research, 2002, vol. 50, p. 904 - 915.

[74] *Berkelaar A., Gromicho J., Kouwenberg A., Zhang S.* A primal-dual decomposition algorithm for multistage stochastic convex programmimng. - Mathematical Programming, 2005, vol. 104, p. 153 - 177.

[75] *Bernard J.C., Ferland J. A.* Convergence of interval-type algorithms for generalized fractional programming. - Mathematical Programming, 1989, vol. 43, nr. 3, p. 349 - 363.

[76] *Bertsekas D.P.* Constrained optimization and Lagrange multiplier methods. - Academic Press, New York, NY, 1982.

[77] *Bertsekas D.P.* Necessary and sufficient conditions for a penalty method to be exact. - Math. Programming, 1975, vol. 9, p. 87 - 99.

[78] *Bertsekas D.P.* Nonlinear programming. - Athena Scientific, Belmont, Massachusets, 1995.

[79] *Bertsimas D., Orlin J.B.* A technique for speeding up the solution of the Lagrangean dual. - Mathematical Programming, 1994, vol. 63, p. 23 - 45.

[80] *Bhatt S.K.* Equivalence of various linearization algorithms for linear fractional programming. - Mathematical Methods of Operations Research, 1989, vol. 33, nr. 1, p. 39 - 43.

[81] *Birge J.R., Louveaux F.* Introduction to stochastic programming. - Berlin: Springer-Verlag, 1997.

[82] *Bitran G.R.* Duality for nonlinear multiple-criteria optimization problems. - Journal of Optimization Theory and Applications, 1981, vol. 35, p. 367 - 401.

[83] *Bitran G.R.* Experiments with linear fractional problems. - Naval Res. Logist. Quart., 1979, vol. 26, nr. 4, p. 689 - 693.

[84] *Bitran G.R., Magnanti T.L.* Duality and sensitivity analysis for fractional programs. - Operations Research, 1976, vol. 24, nr. 4, p. 675 - 699.

[85] *Bitran G.R., Novaes A.J.* Linear programming with a fractional objective function. - Operations Research, 1973, vol. 21, nr. 1, p. 22 - 29.

[86] *Bland R.G., Goldfarb D., Todd M.J.* The ellipsoid method: A survey. - Operations Research, 1981, vol. 29, nr. 6, p. 1039 - 1091.

[87] *Boliac R., Lozovanu D., Solomon D.* Optimal paths in network games with p plazers. - Discrete Applied Mathematics, 2000, vol. 99, nr. 13, p. 339 - 348.

[88] *Bomze I., Csendes T., Horst R., Pardalos P.M.* Development in global optimization. - Kluwer Academic Publishers, 1997.

[89] *Boncompte M., Martinez-Legaz J.E.* Fractional programming by lower subdifferentiability techniques. - Journal of Optimization Theory and Applications, 1991, vol. 68, nr. 1, p. 95 - 116.

[90] *Borde J., Crouzeix J.P.* Convergence of a Dinkelbach-type algorithm in generalized fractional programming. - Mathematical Methods of Operations Research, 1987, vol. 31, nr. 1, p. 31 - 54.

[91] *Bot R.I., Chares R., Wanka G.* Duality for multiobjective fractional programming problems. - Nonlinear Analysis Forum, 2006, vol. 11, nr. 2, p. 185 - 201.

[92] *Boyd S., Vandenberghe L.* Convex optimization. - Cambridge University Press, 2004.

[93] *Brickman L.* Mathematical introduction to linear programming and game theory. - Berlin: Springer-Verlag, 1989

[94] *Bykadorov I.A.* On quasiconvexity in fractional programming. - Generalized convexity and generalized monotonicity. - Lecture Notes in Economics and Mathematical Systems. - Berlin: Springer-Verlag, 1994, vol. 405, p. 281 - 293.

[95] *Caballero R., Hernandez M.* The controlled estimation method in the multiobjective linear fractional problem. - Computers & Operations Research, 2004, vol. 31, nr. 11, p. 1821-1832

[96] *Cabot V.* Maximizing the sum of certain quasi-concave functions using generalized Benders decomposition. - Naval Res. Logist. Quart., 1978, vol. 25, nr. 3, p. 473 - 481.

[97] *Cambini A., Carosi L., Martein L.* On the supremum in quadratic fractional programming. - Generalized convexity and generalized monotonicity. - Lecture Notes in Economics and Mathematical Systems, vol. 502. - Berlin: Springer-Verlag, 2001, p. 129 - 143.

[98] *Cambini A., Castagnoli E., Martein L., Mazzoleni P., Schaible S. (Eds.)* Generalized Convexity and Fractional Programming with Economic Applications. - Proceedings of the "Third Conference on Generalized Convexity held at University of Pisa, Pisa (Italy), May 30 - June 1, 1988. - Lecture Notes in Economics and Mathematical Systems, vol.345. - Springer-Verlag, Berlin, 1990.

[99] *Cambini A., Martein L.* Equivalence in linear fractional programming. - Optmization, 1992, vol. 23, nr. 1, p. 41 - 51.

[100] *Cambini A., Martein L.* Linear fractional and bicriteria linear fractional programs. - Lecture Notes in Economics and Mathematical Systems, vol. 345. - Springer-Verlag, 1990, p.155 - 166.

[101] *Cambini A., Martein L., Schaible S.* On maximizing a sum of ratios. - Journal of Information and Optimization Sciences, 1989, vol. 10, nr. 1, p. 65 - 79.

[102] *Cambini A., Martein L., Schaible S.* On the pseudoconvexity of the sum of two linear fractional functions. - Generalized Convexity, Generalized Monotonicity and Applications. - Nonconvex Optimization and Its Applications, vol. 77. - Springer-Verlag, 2005, p. 161 - 172.

[103] *Cambini A., Martein L., Stancu-Minasian I.M.* A survey of bicriteria fractional problems. - Advanced Modeling and Optimization, 1999, vol. 1, nr. 1, p. 9 - 46.

[104] *Cambini R., Carosi L.* On generalized linearity of quadratic fractional functions. - Journal of Global Optimization, 2004, vol. 30, nr. 2, p. 235 - 251

[105] *Cambini R., Carosi L., Schaible S.* Duality in fractional programming problems with set constraints. -Generalized Convexity, Generalized Monotonicity and Applications. - Nonconvex Optimization and Its Applications, vol. 77. - Springer-Verlag, 2005, p. 147 - 160.

[106] *Carosi L., Martein L.* Some classes of pseudoconvex fractional functions via the Charnes-Cooper transformation. - Generalized Convexity and Related Topics. - Lecture Notes in Economics and Mathematical Systems, vol. 583. - Springer, 2006., p. 177 - 188.

[107] *Chadha S.S.* A decomposition principle for fractional programming. - Opsearch, 1967, vol. 4, nr. 3, p. 123 - 132.

[108] *Chadha S.S.* A dual fractional program. - ZAMM, 1971, vol. 51, p. 560 -561.

[109] *Chadha S.S.* Duality teorems for a generalized linear and linear-fractional program. - Ekonomiko-matematicky obzor, 1972, vol. 8, nr. 4, p. 410 - 416.

[110] *Chadha S.S., Chadha V.* Linear fractional programming and duality. - Central European Journal of Operations Research, 2007, vol. 15, nr. 2, p. 119 - 125.

[111] *Chakraborty M., Gupta S.* Fuzzy mathematical programming for multi objective linear fractional programming problem. - Fuzzy Sets and Systems, 2002, vol. 125, p. 335 - 342.

[112] *Chandra S.S.* Decomposition principle for linear fractional functional programs. - Revue francaise d'informatique et de recherche operationnelle, 1968, Serie rouge, tome 2, nr. 2, p. 65 - 71.

[113] *Chandra S.S., Craven B.D., Mond B.* Generalized fractional programming duality: a ratio game approach. - J. Austral. Math. Soc., 1986, vol. 28, nr. 2, p. 170 - 180.

[114] *Chandra S.S., Craven B.D., Mond B.* Multiobjective fractional programming duality. A Lagrangian approach. - Optimization, 1991, vol. 22, nr. 4, p. 549 - 559.

[115] *Chandra S.S., Craven B.D., Mond B.* Symmetric dual fractional programming. - Mathematical Method of Operations Research, 1985, vol. 29, nr. 1, p. 59 - 64.

[116] *Chandra S.S., Craven B.D., Mond B.* Vector-valued Lagrangian and multiobjective fractional programming duality. - Numer. Funct. Anal. Optim., 1990, vol. 11, p. 239 - 254.

[117] *Chandra S.S., Husain I.* Symmetric dual continuous fractional programming. - J. Inform. Optim. Sci. (India), 1989, vol. 10, nr. 1, p.241 - 255

[118] *Chandra S.S., Kumar V.* Duality in fractional minimax programming. - Journal of Australian Mathematical Society, 1995, vol. 58, p. 376 - 386.

[119] *Chankong V., Haimes Y.Y.* Multiobjective decision making: theory and methodology. - North-Holland, New York, 1983.

[120] *Charles V., Dutta D.* A method for solving linear stochastic fractional programming problem with mixed constraits. - Acta Ciencia Indica, 2004, vol. XXX M, nr. 3, p. 497 - 506.

[121] *Charles V., Dutta D.* A parametric approach to linear probabilistic fractional programming problems. - Mathematical and Computational Models, 2005, nr. 13, p. 171 - 183.

[122] *Charles V., Dutta D.* Bi-weighted multi-objective stochastic fractional programming problem with mixed constraints, in: G. Arulmozhi and R. Nadarajan (Eds.). - Proceedings of the Second National Conference on Mathematical and Computational Methods, Allied Published, 2001, p.29 - 36.

[123] *Charles V., Dutta D.* Extremization of multi-objective stochastic fractional programming problem. - Annals of Operations Research, 2006, vol. 143, nr. 1, p. 297 - 304.

[124] *Charles V., Dutta D.* Identification of redundant objective functions in multi-objective stochastic fractional programming problems. - Asia-Pacific Journal of Operations Research, 2006, vol. 23, nr. 2, p. 155 - 170.

[125] *Charles V., Dutta D.* Linear stochastic fractional programming with branch-and-bound technique. - Proceedings of National Conference on Mathematical and Computational Methods, Allied Published, 2001, p. 131 - 139.

[126] *Charles V., Dutta D.* Non-linear stochastic fractional programming models of financial derivatives. - The ICFAI Journal of Applied Finance, 2005, vol. 11, nr. 6, p. 5 - 13.

[127] *Charles V., Dutta D.* Optimization of linear stochastic fractional programming problem using sign technique. - Mathematical and Computational Models, 2005, vol. 28, p. 302 - 314.

[128] *Charles V., Dutta D.* Two level linear stochastic fractional programming problem with discrepancy vector. - Journal of Indian Society of Statistics and Operations Research, 2002, vol. 23, nr. 1 - 4, p. 59 - 67.

[129] *Charles V., Dutta D., Appal Raju K.* Linear stochastic fractional programming problems. - Proceedings of International Conference on Mathematical Modelling, ITT Roorhee, 2001, p. 211 - 217.

[130] *Charles V., Dutta D., Appala Raju K.* Linear stochastic fractional programming problem. - Proceedings of International Conference on Mathematical Modelling, Tata McGraw Hill, 2001, p. 221 - 217.

[131] *Charnes A.* Optimality and degeneracy in linear programming. - Econometrica, 1952, vol. 20, p. 160 - 170.

[132] *Charnes A., Cooper W.W.* An explicit general solution in linear fractional programming. - Naval Research Logistics Quarterly, 1986, vol. 20, nr. 3, p. 449 - 467.

[133] *Charnes A., Cooper W.W.* Programming with linear fractional functionals. - Naval Research Logistics Quarterly, 1962, vol. 9, nr. 3 - 4, p. 181 - 186.

[134] *Chatelon J., Hearn D., Lowe T.J.* A subgradient algorithm for certain minimax and minisum problems. - Mathematical Programming, 1978, vol. 15, nr. 1, p. 130 - 145.

[135] *Chen D.Z., Daescu O., Dai Y, Katoh N., Wu X., Xu J.* Optimizing the sum of linear fractional functions and applications. - Proceeding of the eleventh annual ACM-SIAM symposium on Discrete algorithms. - Francisco, California, United States, 2000, p. 707 - 716.

[136] *Chen D.Z., Daescu O., Dai Y., Katoh N., Wu X., Xu J.* Efficient algorithms and implementations for optimizing the sum of linear fractional functions. - Applications Journal of Combinatorial Optimization, 2005, vol. 9, nr. 1, p. 69 - 90.

[137] *Chergui M., Moulai M.* An exact method for a discrete multiobjective linear fractional programming. - Journal of Applied Mathematics and Decision Sciences, 2008.

[138] *Chinchuluun A., Yuan D., Pardalos P.M.* Optimality conditions and duality for nondifferentiable multiobjective fractional programming with generalized convexity. - Annals of Operations Research, 2007, vol. 154, nr. 1, p. 133 - 147.

[139] *Choo E.U., Atkins D.R.* Bicriteria linear fractional programming. - Journal of Optimization Theory and Applications, 1982, vol. 36, nr. 2, p. 203 - 220.

[140] *Chou J. H., Hsia W.S., Lee T.Y.* On multiple objective programming problems with set functions. - Journal of Mathematical Analysis and Applications, 1985, vol. 105, p. 383 - 394.

[141] *Clarke F.H.* Optimization and Nonsmooth Analysis. - Wiley-Interscience, New York, 1983.

[142] *Clarke F.H.* Generalized gradients and applications. - Trans. Amer. Math. Soc., 1975, vol. 205, p. 47 - 262.

[143] *Clarke F.H., Stern J.S., Wollenski P.R.* Nonsmooth analysis and control theory. - Berlin: Springer-Verlag, 1998.

[144] *Cook M.D., Kirby M.J.L., Mehndiratta S.L.* A linear fractional max-min problem. - Operations Res., 1975, vol. 23, nr. 3, p. 511 - 521.

[145] *Corley H.W.* Optimization theory for n-set functions. - Journal of Mathematical Analysis and Applications, 1987, vol.127, p. 193 - 205.

[146] *Cottle R., Pang J. S., Stone R.E.* The linear complementarity problem. - Academic Press, New York, 1992.

[147] *Craven B.D.* Duality for generalized convex fractional programs. - Generalized convexity in optimization and economics. - Academic Press, New York, 1981, p. 473 - 489.

[148] *Craven B.D.* Fractional programming. - Berlin: Heldermann Verlag, 1988.

[149] *Craven B.D.* Lagrangian conditions for a minimax. - Proc. Center Math. Anal., Australian National University, 1988, p. 24 - 33.

[150] *Craven B.D., Mond B.* On fractional programming and equivalence. - Naval Res. Logist. Quart., 1975, vol. 22, nr. 2, p. 405 - 410.

[151] *Craven B.D., Mond B.* On maximizing a ratio of optimization problems. - Cahiers Centre Etudes Recherche Oper., 1983, vol. 25, nr. 1 - 2, p. 29 - 34.

[152] *Craven B.D., Mond B.* The dual of a fractional linear program. - J. of Mathem. Anal. and Appl., 1973, nr. 42, p. 507 - 512.

[153] *Crouzeix J.P., Ferland J.A.* Algorithms for generalized fractional programming. - Mathematical Programming, 1991, vol. 52, nr. 2, p. 191 - 207.

[154] *Crouzeix J.P., Ferland J.A., Nguyen V.H.* Revisiting Dinkelbach-type algorithms for generalized fractional programs. - Opsearch, 2008, vol. 45, nr. 2, p. 96 - 110.

[155] *Crouzeix J.P., Ferland J.A., Schaible S.* A note on an algorithm for generalized fractional programs. - Journal of Optimization Theory and Applications, 1986, vol. 50, nr. 1, p. 183 - 187.

[156] *Crouzeix J.P., Ferland J.A., Schaible S.* An algorithm for generalized fractional programs. - Journal of Optimization Theory and Applications, 1985, vol. 47, nr. 1, p. 35 - 49.

[157] *Crouzeix J.P., Ferland J.A., Schaible S.* Duality in generalized linear fractional programming. - Mathematical Programming, 1983, vol. 27, nr.3, p. 342 - 354.

[158] *Crouzeix J.P., Martinez-Legaz J.E., Volle M. (Eds.)* Generalized Convexity, Generalized Monotonicity. - Proceedings of the "Fifth International Symposium on Generalized Convexity held at the Centre International de Rencontres Mathematiques (CIRM), Luminy-Marseille (France), June 17 - 21, 1996. - Nonconvex Optimization and Its Applications, vol.27. - Kluwer Academic Publishers, Dordrecht, 1998.

[159] *Dai Y., Shi J., Wang S.* Conical partition algorithm for maximizing the sum of dc ratios. - Journal of Global Optimization, 2005, vol. 31, nr. 2, p. 253 - 270.

[160] *Danskin J. M.* The theory of max-min. - Springer, New York, 1967.

[161] *Danskin J.M.* The theory of max-min with applications. - SIAM J. Appl. Math., 1996, vol. 14, p. 641 - 664.

[162] *Dantzig G.* Linear programming and extensions. - Princeton University Press, Princeton, NJ, 1963.

[163] *Dantzig G.* Programming in a linear structure. - Economitrica, 1949, vol. 17, p. 73 - 74.

[164] *Dantzig G., Wolfe P.* Decomposition principles for linear programs. - Operations Research, 1960, vol. 8, p. 101 - 111.

[165] *Dantzig G., Wolfe P.* The decomposition algorithm for linear programming. - Econometrica, 1961, vol. 29, nr. 4, p. 767 - 778.

[166] *Dantzig G.B., Thapa M.N.* Linear programming 1: Introduction. - Springer-Verlag New York, Inc., Secaucus, NJ, 1997.

[167] *Dantzig G.B., Thapa M.N.* Linear programming 2: Theory and extensions. - Springer-Verlag New York, Inc., Secaucus, NJ, 2003.

[168] *Datta N.* Efficiency in multiobjective fractional functional programming. - Journal of Information and Optimization Sciences, 1982, vol. 3, nr. 3, p. 262 - 268.

[169] *Depertrini D., Locatelli M.* Approximation of linear fractional-multiplicative problems. - Mathematical Programming, 2009, Ser. A, Short Communication, Published online, 30 october 2009.

[170] *Dikin I.I.* Determination of an interior point of one system of linear inequalities. - Ribernetika and System Analysis, 1992, nr. 1, p. 74 - 96.

[171] *Dinkelbach W.* Maximierung eines quotienten zwei linearen funktionen unter linearen nebenbedingungen. - Z. Wahrsch. Verw. Gebiete, 1962, nr. 1, p. 141 - 145.

[172] *Dinkelbach W.* On nonlinear fractional programming. - Management Sci., 1967, vol. 13, nr. 7, p. 492 - 498.

[173] *Dorfman R., Samuelson P.A., Salow R.M.* Linear programming and economic analysis. - Dover Publications, 1987.

[174] *Dorn W.S.* Linear fractional programming. - IBM Research Report, RC-830, Nov. 27, 1962.

[175] *Du D.Z., Pardalos P.M.* Minimax and Applications. - Kluwer Academic Publishers, The Netherlands, 1995.

[176] *Dur M., Horst R., Thoai N.V.* Solving sum-of-ratios fractional programs using efficient points. - Optimization, 2001, vol. 49, nr. 5 - 6, p. 447 - 466.

[177] *Dutta D., Tiwari R.N., Rao J.R.* Multiple objective linear fractional programming - a fuzzy set theoretic approach. - Fuzzy Sets and Systems, 1992, vol. 52, nr.1, p. 39 - 45.

[178] *Eberhard A., Hadjisavvas N., Luc D.T. (Eds.)* Generalized Convexity, Generalized Monotonicity and Applications. - Nonconvex Optimization and Its Applications, vol. 77. - Springer-Verlag, 2005.

[179] *Egudo R.R.* Multiobjective fractional duality. - Bull. Austral. Math. Soc., 1988, vol. 37, p. 367 - 378.

[180] *Ehrgott M.* Multicriteria optimization. - Berlin: Springer-Verlag, 2000.

[181] *Elster K.H., Qolf A.*] Generalized convexity and fractional programming. - Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, 1990, vol. 345, p. 219-231.

[182] *Elster K.H., Wolf A.* Generalized convexity and fractional optimization. - Lecture Notes in Economics and Mathematical Systems, vol.345. - Springer-Verlag, Berlin, 1990, p.219 - 231.

[183] *Ermoliev Y.* Stochastic Quasi-Newton methods and their applications to systems optimization. - Stochastics, 1983, vol. 9, p. 1 - 36.

[184] *Ermoliev Y., Wets R.J.B (Eds)* Numerical techniques for stochastic optimization. - Berlin: Springer-Verlag, 1988.

[185] *Falk J.E., Palocsay S.W.* Image space analysis of generalized fractional programs. - Journal of Global Optimization, 1994, vol. 4, p. 63 - 88.

[186] *Falk J.E., Palocsay S.W.* Optimizing the sum of linear fractional functions. - Recent Advances in Global Optimization. - Princeton University Press, 1992, p. 221 - 258.

[187] *Falk J.E.* Lagrange multipliers and nonlinear programming. - J. Math. Anal. Appl., 1967, vol. 19, nr. 1.

[188] *Fang S.C., Gao D.Y., Sheu R.L., Xing W.X.* Global optimization for a class of fractional programming problems. - Journal of Global Optimization, 2009, vol. 45, nr. 3. p. 337 - 353.

[189] *Ferland J.A., Potvin, J.Y.* Generalized fractional programming: algorithms and numerical experimentation. - European Journal of Operational Research, 1985, vol. 20, nr. 1, p. 92 - 101.

[190] *Fiacco A.V., McCormick G.P.* Nonlinear programming: Sequential unconstrained minimization techniques. - John Wiley and Sons, NY, 1968.

[191] *Flachs J.* Generalized Cheney-Loeb-Dinkelbach-type algorithms. - Mathematics of Operations Research, 1985, vol. 10, nr. 4, p. 674 - 687.

[192] *Flachs J., Pollatschek M.A.* Equivalence between a generalized Fenchel duality theorem and a saddle-point theorem for fractional programs. - J. Optima. Theory Appl., 1982, vol. 37, nr. 1, p. 23 - 32.

[193] *Fletcher R.* Practical methods of optimization. - Wiley, Chichester, 1987.

[194] *Floudas C.A. et al.* Handbook of test problems in local and global optimization. - Kluwer Academic Publisher, 1999.

[195] *Floudas C.A., Gounaris C.E.* A review of recent advances in global optimization. - Journal of Global Optimization, 2009, vol. 45, nr. 1, p. 3 - 38.

[196] *Floudas C.A., Pardalos P.M. (Eds)* Recent advances in global optimization. - Princeton University Press, 1992.

[197] *Floudas C.A., Pardalos P.M.* State of the art in global optimization: computational methods and applications. - Kluwer Academic Publisher, 1996.

[198] *Floudas C.A., Visweswaran V.* Quadratic optimization. - Handbook of global optimization. - Kluwer Academic Publishers, Dordrecht/ Boston/ London, 1995, p. 217 - 270.

[199] *Ford L.R., Fulkerson D.R.* Flows in networks. - Princeton, New Jersey: Princeton University Press, 1962.

[200] *Frank M., Wolfe P.* An algorithm for quadratic programming. - Naval research Logistics Quarterly, 1956, vol. 3, p. 95 - 110.

[201] *Frauendorfer K.* Stochastic Two-stage programming. - Lecture Notes in Economics and Mathematical Systems, vol. 392. - Berlin: Springer-Verlag, 1992.

[202] *Frenk J.B.G., Gromicho J., Plastria F. and S. Zhang,* A deep cut ellipsoid algorithm and quasiconvex programming. - Generalized convexity and generalized monotonicity. - Lecture Notes in Economics and Mathematical Systems. - Berlin: Springer-Verlag, 1994, vol. 405, p.62 - 76.

[203] *Frenk J.B.G., Schaible S.* Fractional programming. - Handbook of generalized convexity and generalized monotonicity. - Berlin: Springer-Verlag, Heindelberg - New York, 2004, p. 335 - 386.

[204] *Frenk J.B.G., Sturm J.F., Zhang S.* An interior-point based subgradient method for nondifferentiable convex programming. - Optimization Methods and Software, 1988, vol. 10, p. 197 - 215.

[205] *Freund R.W.* Polinomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential functions. - Mathematical Programming, 1991, vol. 51, p. 203 - 222‘.

[206] *Freund R.W., Jarre F.* An interior-point method for convex fractional programming. - Mathematical Programming, 1994, vol. 67, nr. 3, p. 407 - 440.

[207] *Freund R.W., Jarre F.* An interior-point method for multifractional programs with convex constraints. - Journal of Optimization Theory and Applications, 1995, vol. 85, nr. 1, p. 125 - 161.

[208] *Freund R.W., Jarre F.* Solving the sum of ratios problem by an interior-point method. - Journal of Global Optimization, 2001, vol. 19, nr. 1, p. 83 - 102.

[209] *Freund R.W., Jarre F., Schaible S.* On self-concordant barrier functions for conic hulls and fractional programming. - Mathematical Programming, 1996, vol. 74, nr. 3, p. 237 - 246.

[210] *Gacs P., Lovasz L.* Khachiyan's algorithm for linear programming. - Math. Program. Studies, 1981, vol. 14, p. 61 - 68.

[211] *Gao D.Y.* Canonical dual transformation method and generalized triality theory in nonsmooth global optimization. - J. Global Optimization, 2000, vol.17 (1/4), p. 127 - 160.

[212] *Gao D.Y.* Canonical duality theory and solutions to constrained nonconvex quadratic programming. - Journal of Global Optimization, 2004, vol. 29, p. 377 - 399.

[213] *Gao D.Y.* Complete solution and extremality criteria to polynomial optimization problems. - Journal of Global Optimization, 2006, vol. 35, p. 131 - 143.

[214] *Gao D.Y.* Duality Principles in nonconvex systems: theory, method and applications. - Kluwer Academic Publishers, Dordrecht /Boston /London, vol. 18, 2000.

[215] *Gao D.Y., Ruan N.* Solutions and optimality criteria for nonconvex quadratic-exponential minimization problem. - Mathematical Methods of Operations Research, 2008, vol. 67, nr. 3, p. 479 - 496.

[216] *Gao Y., Xu C., Yang Y.* An outcome-space finite algorithm for solving linear multiplicative programming. - Applied Mathematics and Computation, 2006, vol. 179, p. 495 - 505.

[217] *Gass S.I.* Linear programming: Methods and applications. - McGraw-Hill, New York, 1964.

[218] *Gay D.M.* A variant of Karmarkar's linear programming algorithm for problems in standard form. - Mathematical Programming, 1987, vol. 37, nr. 1, p. 81 - 90.

[219] *Ghellink G., Vial J.P.* A polynomial Newton method for linear programming. - Algorithmica, 1986, nr. 1, p. 425 - 454.

[220] *Ghellink G., Vial J.P.* An extension of Karmarkar's algorithm for problems in standard form. - Mathematical Programming, 1987, vol. 37, p. 81 - 90.

[221] *Gill P.E., Murray W., Wright M. H.* Practical optimization. - Academic Press, New York, London, Toronto, Sydney, San Francisco, 1981.

[222] *Gilmore P.G., Gomory R.E.* A linear programming approach to the cutting stock problem. - Operation Research, 1961, vol. 9, p. 849 - 859.

[223] *Gilmore P.G., Gomory R.E.* A linear programming approach to the cutting stock problem. Part II. - Operation Research, 1963, vol. 11, p. 863 - 888.

[224] *Goffin J.L., Haurie A., Vial J.P.* Decompozition and nondifferentiable optimization with the projective algorithm. - Management Science, 1992, vol. 38, p. 284 - 302.

[225] *Goffin J.L., Luo Z.Q., Ye Y.* On the complexity of a column generation algorithm for convex or quasiconvex problems. - Large Scale Optimization: The State of the Art. - Kluwer Academic Publishers, 1993.

[226] *Gogia N.K.* The multiplex method for linear fractional programming. - Cah. Cent. Etud. Rech. Oper., 1967, vol. 9, nr. 3, p. 123 - 133.

[227] *Goldfarb D., Mehrotra S.* A relaxed varsion of Karmarkar's method. - Mathematical Programming, 1988, vol. 40, nr. 3, p. 289 - 315.

[228] *Goldfarb D., Mehrotra S.* Relaxed variants of Karmarkar's algorithm for linear programs with unknown optimal objective value. - Mathematical Programming, 1988, vol. 40, nr. 2, p. 183 - 195.

[229] *Gotoh J.Y., Konno H.* Maximization of the radio of the two convex quadratic functions over a polytope. - Computational Optimization and Applications, 2001, vol. 20, nr. 1, p. 43 - 60.

[230] *Grotchel M., Lovasz L., Schrijver A.* Geometric algorithms and combinatorial optimization. - Berlin: Springer-Verlag, 1988.

[231] *Gugat M.* A fast algorithm for a class of generalized fractional programs. - Management Science, 1996, vol. 42, nr.10, p. 1493 - 1499.

[232] *Gugat M.* Prox regularization methods for generalized faractional programming. - Journal of Optimization Theory and Applications, 1988, vol. 9, p. 691 - 722.

[233] *Gulati T.R.* Duality for a nondifferentiable fractional program. - Cahiers Centre Etudes Recherche Oper., 1979, vol. 21, nr. 4, p. 325 - 330.

[234] *Gupta A., Puri M.C.* Maximizing pseudoconvex transportation problem a special type. - OR Spectrum, 1995, vol. 17, nr. 1, p. 27 - 30.

[235] *Gupta B.* Programming with multi-objective linear fractional functionals. - Acta Ciennc. Indiaca Math., 1983, vol. 9, nr. 1 - 4, p. 195 - 201.

[236] *Gupta P., Bhatia D.* Duality for fractional min-max problems involving arcwise connected and generalized arcwise connected functions. - Generalized convexity and generalized monotonicity. - Lecture Notes in Economics and Mathematical Systems, vol. 502. - Berlin: Springer-Verlag, 2001, p. 218 - 230.

[237] *Gupta R.* Decomposition method and transportation type problems with a fractional objective functions. - Z. Angew. Math. Mech., 1977, vol. 57, nr. 2, p. 81 - 88.

[238] *Gupta S.N., Swarup K.* Stochastic fractional functionals programming. - Ricerca Operativa, 1979, vol. 9, nr. 10, p. 65 - 78.

[239] *Gwinner J., Jeyakumar V.* A solvability theorem and minimax fractional programming. - Mathematical Methods of Operations Research, 1993, vol. 37, nr. 1, p. 1 - 12.

[240] *Hadjisavvas N., Martinez-Legaz J.E., Penot J-P. (Eds.)* Generalized Convexity and Generalized Monotonicity. - Lecture Notes in Economics and Mathematical Systems, vol.502. - Springer-Verlag, 2001.

[241] *Hadley G.* Linear programming. - Addison-Wesley Publishing Company, Reading, Massachusetts, 1972.

[242] *Hager W., Hearn D., Pardalos P.M.* Large scale optimization: the state of the art. - Kluwer Academic Publishers, 1994.

[243] *Han S.P., Mangasarian O.L.* A dual differentiable exact penalty functions. - Mathematical Programming, 1983, vol. 25, p. 293 - 306.

[244] *Han S.P., Mangasarian O.L.* Exact penalty functions in nonlinear programming. - Mathematical Programming, 1979, vol. 17, p. 251 - 269.

[245] *Helmberg H., Rendl F., Vanderbey R.J., Wolkowicz H.* An interior-point method for semidefinite programming. - SIAM Journal on optimization, 1996, vol. 6, nr. 2, p. 342 - 361.

[246] *Higle J.L., Sen S.* Stochastic decomposition: A statistical method for large scale stochastic linear programming. - Kluwer Academic Publishers, Dordrecht, London, Boston, 1996.

[247] *Hillier F.S., Lieberman G.J.* Introduction to operations research. - McGraw-Hill, San Francisco, 1995.

[248] *Hirche J.* A note on programming problems with linear-plus-linear fractional objective functions. - European Journal of Operations Research, 1996, vol. 89, nr. 1, p. 212 - 214.

[249] *Hirche J.* On programming problems with a linear plus linear-fractional objective function. - Cahiers Centre Etudes Recherche Oper., 1984, vol. 26, nr. 1 - 2, p. 59 - 64.

[250] *Hirche J.* Optimizing of sums and products of linear fractional functions under linear constraints. - Optimization, 1996, vol. 38, nr. 1, p. 39 - 48.

[251] *Hirche J.* Some remarks on generalized convexity of sums and products. - Z. Angew. Math. Mech., 1985, vol. 65, nr. 1, p. 62 - 63.

[252] *Hoai-Phuong Ng. T., Tuy H.* A unified monotonic approach to generalized linear fractional programming. - Journal of Global Optimization, 2003, vol. 26, p. 229 - 259.

[253] *Holmberg K.* Efficient decompozition and linearization methods for the stochastic transportation problem. - Computational Optimization and Applications, 1995, vol. 4, nr. 4, p. 293 - 316.

[254] *Hooker J.N.* Karmarkar's linear programming algorithm. - Interfaces, 1986, vol.16, nr. 4, p. 75 - 90.

[255] *Horst R., Pardalos P.M. (Eds.)* Handbook of global optimization. - Kluwer Academic Publishers: Dordrecht, 1995.

[256] *Horst R., Pardalos P.M., Thoai N.V.* Introduction to global optimization. - Kluwer Academic Publishers: Dordrecht, 1995.

[257] *Horst R., Tuy H.* Global optimization: Deterministic approaches. - Springer-Verlag: Berlin, 1995.

[258] *Husain I., Jabeen Z.* On fractional programming containing support functions. - J. Appl. Math. and Comp., 2004, vol. 18, nr. 1 - 2, p. 361 - 376.

[259] *Husain Z., Jayswal A., Ahmad I.* Second order duality for nondifferentiable minimax programming problems with generalized convexity. - Journal of Global Optimization, 2009, vol. 44, nr. 4, p.593 - 608

[260] *Ibaraki T.* Parametric approaches to fractional programming. - Mathematical Programming, 1983, vol. 26, nr. 3, p. 345 - 362.

[261] *Ibaraki T.* Solving mathematical programming problems with fractional objective functions. - Generalized Concavity in Optimization and Eonomics. - Academic Press, New York, 1981, p. 441 - 472.

[262] *Ibaraki T., Ishii H., Iwase J., Hasegawa T., Mine H.* Algorithms for quadratic fractional programming problems. - Journal of the Operations Research Society of Japan, 1976, vol. 19, p.174 - 191.

[263] *Ibaraki T., Schaible S.* Invited review: fractional programming. - European Journal of Operations Research, 1983, vol. 12, p. 325 - 338.

[264] *Isbell J.R., Marlow W.H.* Attrition games. - Naval Res. Logist. Quart., 1956, vol. 3, nr. 1, p. 71 - 93.

[265] *Ishii H., Ibakari T., Mine H.* Fractional knapsack problems. - Mathematical Programming, 1976, vol. 13, nr. 3, p. 255 - 271.

[266] *Jagannathan R.* An algorithm for a class of nonconvex programming problems with nonlinear fractional objectives. - Management Science, 1985, vol. 31, nr. 7, p. 847 - 851.

[267] *Jagannathan R.* Duality for nonlinear fractional programming. - Zeitschrift fur Operations Research, 1973, vol. 17, nr. 1, p. 1 - 3.

[268] *Jagannathan R., Schaible S.* Duality in generalized fractional programming via Farkas' lemma. - Journal of Optimization Theory and Applications, 1983, vol. 41, nr. 3, p. 417 - 424.

[269] *Jahn J.* Introduction to the theory of nonlinear optimization. - Berlin: Springer-Verlag, 1994.

[270] *Jain O.P.* Duality for fractional functional programming. - Cahiers Centre Etudes Recherche Oper., 1979, vol. 21, nr. 1, p. 81 - 86.

[271] *Jarre F.* An interior-point methods for minimizing the maximum eigenvalue of a linear combination of matrices. - SIAM J. Control and Opt., 1993, vol. 3, p. 1360 - 1377.

[272] *Jarre J.* Interior-point methods for convex programming. - Appl. Math. Opt., 1992, vol. 26, p. 287 - 311.

[273] *Jaumard B., Meyer C., Tuy H.* Generalized convex multiplicative programming via quasiconcav minimization. - Journal of Gobal Optimization, 1997, vol. 10, nr. 3, p. 229 - 256.

[274] *Jayswal A.* Non-differentiable minimax fractional programming with generalized a-convexity. - Journal of Computational and Applied Mathematics, 2008, vol. 214, nr.1, p.121 - 135.

[275] *Jeyakumar V.* Equivalence of saddle-points and optima, and duality for a class of non-smooth non-convex problems. - Journal of Mathematical Analysis and Applications, vol. 130, 1988, p. 334 - 343.

[276] *Jeyakumar V.* Strong and week invexity in mathematical programming. - Methods. Oper. Res., 1985, vol. 55, p. 109 - 125.

[277] *Jeyakumar V., Mond B.* On generalized convex mathematical programming. - J. Austral. Math. Soc., 1992, vol. 34, p. 43 - 53.

[278] *Jiao H.W., Guo Y.R., Shen P.P.* Global optimization of generalized linear fractional programming with nonlinear constraints. - Appl. Math. Comput., 2006, vol. 183, nr. 2, p. 717 - 728.

[279] *Jo C.L., Kim D.S., Lee G.M.* Duality for multiobjective fractional programming involving n-set functions. - Optimization, 1994, vol. 29, nr. 3, p. 205 - 213.

[280] *Joksch H.C.* Programming with fractional linear objective. - Naval Research Logistics Quarterly, 1964, vol. 11, p. 197 - 204.

[281] *Kall P.* Stochastic linear programming. - Berlin: Springer-Verlag, 1972.

[282] *Kall P., Wallace S.W.* Stochastic programming. - Chichester; New York; Brisbane: John Wiley and Sons, 1994.

[283] *Kanchan P.K.* Linear-fractional functional programming. - Acta Cienc. Indica, 1976, vol. 2, nr. 4, p. 401 - 405.

[284] *Kanchan P.K., Holland A.S.B., Sahney B.N.* Transportation techniques in linear-plus-fractional programming. - Cahiers Centre Etudes Recherche Oper., 1981, vol. 23, p. 153 - 157.

[285] *Karmarkar N.* A new polynomial-time algorithm for linear programming. - Combinatorica, 1984, vol. 4, nr. 4, p. 373 - 395.

[286] *Karmarkar N.K., Ramakrishnan K.G.* Computational results of an interior point algorithm for large scale linear programming. - Mathematical Programming, 1991, vol. 52, p. 555 - 586.

[287] *Karmarkar N.K., Resende M.G.C., Ramakrishnan K.G.* An interior point algorithm to solve computationally difficult set covering problems. - Mathematical Programming, 1991, vol. 52, p. 597 - 618.

[288] *Kaska J.* Duality in linear fractional programs. - Ekonom. Math. Obzor, 1969, vol. 5, p. 4422 - 453.

[289] *Kaul R.N.* Generalized linear fractional programming. - Ekonomiko-matematicky obzor, 1974, vol. 10, nr. 3, p. 322 - 330.

[290] *Kaul R.N., Lata M.* A method of decomposition for linear fractional programming. - Opsearch, 1974, vol. 11, nr. 4, p. 183 - 192.

[291] *Kaul R.N., Lyall V.* A note on nonlinear fractional vector maximization. - Opsearch, 1989, vol. 26, nr. 2, p. 108 - 121.

[292] *Kaul R.N., Suneja S.K., Lalitha C.S.* Duality in pseudolinear multiobjective fractional programming. - Indian J. Pure Appl. Math., 1993, vol. 24, nr. 5, p. 279 - 290.

[293] *Khachiyan J., Kalantari B.* Diagonal matrix scaling and linear programming. - SIAM Journal on Optimization, 1992, vol. 2, nr. 4, p. 668 - 672.

[294] *Khan Z.A., Hanson M.A.* On ratio invexity in mathematical programming. - Journal of Mathematical Analysis and Applications, 1997, vol. 205, nr. 2, p. 330 - 336.

[295] *Kim D.S.* Multiobjective fractional programming with a modified objective function. - Commun. Korean Math. Soc., 2005, vol. 20, nr. 4, p. 837 - 847.

[296] *Kim D.S.* Nonsmooth multiobjective fractional programming with generalized invexity. - Taiwanese Journal of Mathematics, 2006, vol. 10, nr. 2, p. 457 - 478.

[297] *Kim D.S., Jo C.L., Lee G.M.* Duality relations for generalized fractional programming involving n-set functions. - Indian Journal of Pure and Applied Mathematics, 1996, vol. 27, nr. 12, p. 1167 - 1173.

[298] *Kim D.S., Jo C.L., Lee G.M.* Optimality and duality for multiobjective fractional programming involving n-set functions. - Journal of Mathematical Analysis and Applications, 1998, vol. 224, p. 1 - 13.

[299] *Kim D.S., Kim S.J., Kim M.H.* Optimality and duality for a class of nondifferentiable multiobjective fractional programming problems. - Journal of Optimization Theory and Applications, 2006, vol. 129, nr. 1, p. 131 - 146.

[300] *Kim D.S., Kim S.J.* Nonsmooth fractional programming with generalized ratio invexity. - Proc. of RIMS Symposium "Nonlinear Analysis and Convex Analysis 2004, p. 116 - 127.

[301] *Kim D.S., Lee Y.J, Bae K.D.* Duality in nondifferentiable multiobjective fractional programming insolving cones. - Taiwanese Journal of Mathematics, 2009, vol. 13, nr. 6A, p. 1811 - 1821.

[302] *Kiwiel K.C.* Methods of descent for nondifferentiable optimization. - Lecture Notes in Mathematics, vol. 1133, 1985.

[303] *Kojima M.S., Mizuno S., Yoshise A.* A polynomial time algorithm for linear complementarity problems. - Mathematical Programming, 1989, vol. 44, p. 1 - 26.

[304] *Komlosi, S., Rapcsak T., Schaible S. (Eds.)* Generalized Convexity. - Proceedings of the "Fourth International Workshop on Generalized Convexity held at Janus Pannonius University, Pecs (Hungary), August 31 - September 2, 1992. - Lecture Notes in Economics and Mathematical Systems, vol. 405. - Springer-Verlag, Berlin, 1994.

[305] *Konno H.* Minimization of the sum of several linear fractional functions. - Generalized Convexity and Generalized Monotonicity. - Lecture Notes in Economics and Mathematical Systems, vol.502. - Springer-Verlag, 2001, p.3 - 20.

[306] *Konno H., Abe N.* Minimization of the sum of three linear fractional funstions. - Journal of Global Optimization, 1999, vol. 15, nr. 4, p. 419 - 432.

[307] *Konno H., Fukaishi K.*] A branch-and-bound algorithm for solving low rank linear multiplicative and fractional programming problems. - Journal of Global Optimization, 2000, vol. 18, p. 283 - 299.

[308] *Konno H., Kuno T.* Generalized linear multiplicative and fractional programming. - Annals of Operations Research, 1990, vol. 25, nr. 1 - 4, p. 147 - 161.

[309] *Konno H., Kuno T.* Linear multiplicative programming. - Mathematical Programming, 1992, vol. 56, nr. 1 - 3, p. 51 - 64.

[310] *Konno H., Kuno T., Yajima Y.* Global minimization of a generalized convex multiplicative function. - Journal of Global Optimization, 1994, vol. 4, nr. 1, p. 47 - 62.

[311] *Konno H., Thach P.T., Tuy H.* Optimization of low rank nonconvex structures. - Kluwer Academic Publishers: Dordrecht, 1997.

[312] *Konno H., Yajima Y.* Minimizing and maximizing the product of linear fractional functions. - Recent Advances in Global Optimization. - Princeton, New Jersey: Princeton University Press, 1992, p. 259 - 273.

[313] *Konno H., Yamashita H.* Minimizing sums and products of linear fractional functions over a polytope. - Naval Research Logistics, vol. 46, 1999, p. 583 - 596.

[314] *Konnov I., Luc D.T. Rubinov A. (Eds)* Generalized Convexity and Related Topics. - Lecture Notes in Economics and Mathematical Systems, vol. 583. - Springer, 2006.

[315] *Kornbluth J. S. H., Steuer R. E.* Multiple objective linear fractional programming. - Management Science, 1981, vol. 27, nr. 9, p. 1024 - 1039.

[316] *Kornbluth J.S.H.* On the use of multiple objective linear programming algorithms to solve problems with fractional objectives. - European Journal of Operations Research, 1986, vol. 23, nr. 1, p. 78 - 81.

[317] *Kornbluth J.S.H., Salkin G.R.* The optimal dual solution in linear fractional decomposition problems. - Operations Research, 1974, vol. 22, p. 183 - 189.

[318] *Kornbluth J.S.H., Steuer R.E.* Goal programming with linear fractional criteria. - European J. Oper. Res., 1981, vol. 8, nr. 1, p. 58 - 65.

[319] *Korte B.* Modern applied mathematics. - North-Holland, N.Y., 1982.

[320] *Kovacs A., Stahl J.* On large scale linear fractional programs. - Lecture Notes Computer Science, 1976, nr. 41, p. 353 - 361.

[321] *Kuk H., Lee G.M., Tanino T.* Optimality and duality for nonsmooth multiobjective fractional programming with generalized invexity. - J. Math. Anal. Appl., 2001, vol. 262, p. 365 - 375.

[322] *Kuk H.* Duality for nonsmooth multiobjective fractional programming with V-p-invexity. - J. KSIAM, 2000, vol. 4, nr. 1, p. 1 - 10.

[323] *Kuno T.* A branch-and-bound algorithm for maximizing the sum of several linear ratios. - Journal of Global Optimization, 2002, vol. 22, nr. 1 - 4, p. 155 - 174.

[324] *Kuno T.* A finite brand-and-bound algorithm for linear multiplicative programming. - Computational Optimization and Applications, 2001, vol. 20, p. 119 - 135.

[325] *Kuno T.* A revision of the trapezoidal branch-and-bound algorithm for linear sum-of-ratios problems. - Journal of Global Optimization, 2005, vol. 33, nr. 2, p. 215 - 234.

[326] *Kunzi H.P., Krelle W.* Nonlinear programming. - Blaisdell, Waltham, 1966.

[327] *Kydland F.* Duality in fractional programming. - Naval Research Logistics Quarterly, 1972, vol. 19, p. 691 - 697.

[328] *Lai H.C., Lee J.C.* Necessary and sufficient conditions for minimax fractional programming. - J. Math Anal. Appl., 1999, vol. 230, nr. 2, p. 311 - 328.

[329] *Lai H.C., Lee J.C.* On duality theorems for a nondifferentiable minimax fractional programming. - Journal of Computational and Applied Mathematics, 2002, vol. 146, nr. 1, p. 115 - 126.

[330] *Lai H.C., Liu J.C.* Complex fractional programming involving generalized quasi/pseudo convex functions. - Journal of Applied Mathematics and Mechanics, 2002, vol. 82, nr. 3, p. 159 - 166.

[331] *Lai H.C., Liu J.C.* Minimax fractional programming involving generalized invex functions. - ANZIAM J., 2003, vol. 44, p. 339 - 354.

[332] *Lai H.C., Liu J.C.* On minimax fractional programming of generalized convex set functions. - J. Math. Anal. Appl., 2000, vol. 244, nr. 2, p. 442 - 465.

[333] *Lai H.C., Liu J.C., Tanaka K.* Duality without a constraint qualification for minimax fractional programming. - J. Optim. Theory Appl.. 1999, vol. 101, nr. 1, p. 109 - 125.

[334] *Lai H.C., Liu J.C., Tanaka K.* Necessary and sufficient conditions for minimax fractional programming. - J. Math Anal. Appl., 1999, vol. 230, nr. 2, p. 311 - 328.

[335] *Lasdon L.S.* Optimization theory for large systems. - Macmillan: London, 1970.

[336] *Lata M., Mittal B.S.* A decomposition method for interval linear fractional programming. - Z. Angew. Math. Mech., 1976, vol. 56, nr. 4, p. 153 - 159.

[337] *Lee B.I., Tcha D.W.* An iterative procedure for fuzzy programming with linear fractional objectives. - Computers ind. Engng, 1989, vol. 16, p. 269 - 275.

[338] *Lee J.C., Lai H.C* Parameter-free dual models for fractional programming with generalized invexity. - Annals Operation Research, 2005, vol. 133, p. 47 - 61.

[339] *Liang Z., Shi Z.* Optimality conditions and duality for a minimax fractional programming with generalized convexity. - J. Math. Anal. Appl., 2003, vol. 277, nr. 2, p. 474 - 488.

[340] *Liang Z.A., Huang H.X., Pardalos P.M.* Efficiency conditions and duality for a class of multiobjective fractional programming problems. - J. Global Optim., 2003, vol. 27, nr. 4, p. 447 - 471.

[341] *Liang Z.A., Huang H.X., Pardalos P.M.* Optimality conditions and duality for a class of nonlinear fractional programming problems. - J. Optim. Theory Appl., 2001, vol. 110, nr. 3, p. 611 - 619.

[342] *Liu J.C., Wu C.S., Sheu R.L.* Duality for fractional minimax programming. - Optimization, 1997, vol. 41, p. 117 - 133.

[343] *Liu J.C.* Optimality and duality for multiobjective fractional programming involving nonsmooth pseudoinvex functions. - J. Math. Anal. Appl., 1996, vol. 202, p. 667 - 685.

[344] *Liu J.C., Wu C.S.* On minimax fractional optimality conditions with invexity. - Journal of Mathematical Analysis and Applications, 1998, vol. 219, p. 21 - 35

[345] *Liu J.C., Wu C.S.* On minimax fractional optimality conditions with (F,q)-convexity. - Journal of Mathematical Analysis and Applications, 1998, vol. 219, p. 36 - 51

[346] *Long X.J., Huang N.J., Liu Z.B.* Optimality conditions, duality and saddle points for nondifferentiable multiobjective fractional programs. - Journal of Industrial and Management Optimization, 2008, vol. 4, nr. 2, p. 287 - 298.

[347] *Lozovanu D., Pickl S.* Optimization and multiobjective control of time-discrete systems. - Springer-Verlag Berlin Heidelberg, 2009

[348] *Lozovanu D., Solomon D., Zelikovsky A.* Multiobjective games and determining Pareto-Nash equilibria. - Buletinul ASM. Seria Matematica, 2005, vol. 49, nr. 3, p. 115 - 122.

[349] *Luenberger D.G.* Introduction to linear and non-linear programming. - Addison-Wesley Publishing Company, Reading, Massachusetts, 1973.

[350] *Luhandjula M.K.* Fuzzy approaches for multiple objective linear fractional optimization. - Fuzzy Sets Syst., 1984, vol. 13, nr. 1, p. 11 - 24.

[351] *Luo H.Z., Wu H.X.* On necessary conditions for a class of nondifferentiable minimax fractional programming. - Journal of Computational and Applied Mathematics, 2008, vol. 215, nr. 1, p.103 - 113.

[352] *Mangasarian O.L.* Nonlinear programming. - McGraw-Hill, New York, 1969.

[353] *Mangasarian O.L.* Nonlinear fractional programming. - Journal of the Operations Research Sosiety of Japan, 1969, vol. 12, p. 1 - 10.

[354] *Mangasarian O.L.* Sufficiency of exact penalty minimization. - SIAM J. Control Optim., 1985, vol. 23, p. 30 - 37.

[355] *Martein L.* Massimo della somma tra una funzione lineare ed una funzione lineare fratta. - Rev. mat. econ. e soc., 1985, vol. 8, nr. 1, p. 13 - 20.

[356] *Martos B.* Hyperbolic programming by simplex method. - Deuxieme Congres Mathematique Hongrois, Budapest, 24, August 31, 1960. -Akademiai Kaido, Budapest, 6, 1961, p. 44 - 48.

[357] *Martos B.* Hyperbolic programming. - Naval Research Logistics Quarterly, 1964, vol. 11, nr. 2 - 3, p. 135 - 156.

[358] *Martos B.* Hyperbolikus programozas. - Publications of the Math. Hungarian Academy Sci., 1960, 5, seria B, p. 383 - 406.

[359] *Martos B.* Nonlinear programming: theory and method. - North-Holland, Amsterdam, The Netherlands, 1975.

[360] *Martos B.* The direct power of adjacent vertex programming methods. - Management Science, 1965, vol. 17, p. 241 - 252.

[361] *Matsui T.* NP-hardness of linear multiplicative programming and related problems. - Journal of Global Optimization, 1996, vol. 9, p. 113 - 119.

[362] *Megiddo N.* Combinatorial optimization with rational objective functions. - Mathematics of Operations Research, 1979, vol. 4, p. 414 - 424.

[363] *Metev B., Gueorguieva D.* A simple method for obtaining weakly efficient points in multiobjective linear fractional programming problems. - European Journal of Operational Research, 2000, vol. 126, no. 2, p. 386 - 390.

[364] *Miettinen K.M.* Nonlinear multiobjective optimization. - Kluwer Academic Publishers: Dordrecht, 1999.

[365] *Migdalas A., Pardalos P.M., Varbrand P.* Multilevel optimization: algorithms and applications. - Kluwer Academic Publisher, 1997.

[366] *Minoux M.* Mathematical programming: theory and algorithms. - Wiley, New York, Chichester, 1986.

[367] *Mishra S. K.* Generalized fractional programming problems containing locally subdifferentiable and ?-univex functions. - Optimization, 1997, vol. 41, nr. 2, p. 135 - 158.

[368] *Mishra S., Das C.* The sum of a linear and linear fractional function and three dimensional transportation problems. - Opsearch, 1981, vol. 18, p. 139 - 157.

[369] *Mishra S.K.* Generalized pseudoconvex minmax programming. - Opsearch, 1998, vol. 35, p. 32 - 44.

[370] *Mishra S.K.* Lagrange multipliers saddle points and scalarizations in composite multiobjective nonsmooth programming. - Optimization, 1996, vol. 38, nr. 2, p. 93 - 105.

[371] *Mishra S.K.* On multiple-objective optimization with genaralized convexity. - Journal of Mathematical Analysis and Applications, 1998, vol. 224, p. 131 - 148.

[372] *Mishra S.K.* Pseudolinear fractional minmax programming. - Indian J. Pure Appl. Math., 1995, vol. 26, nr. 8, p. 763 - 772.

[373] *Mishra S.K., Ghosh A.* Interactive fuzzy programming approach to Bi-level quadratic fractional programming problems. - Annals of Operations Research, 2006, vol. 143, nr. 1, p. 131 - 148.

[374] *Mishra S.K., Mukherjee R.N.* Duality for multiobjective fractional variational problems. - J. Math. Anal. Appl., 1994, vol. 186, p. 711 - 725.

[375] *Mishra S.K., Mukherjee R.N.* Generalized cantinuous non-differentiable fractional programming problems with invexity. - J. Math. Anal. Appl., 1995, vol. 195, p. 191 - 213.

[376] *Mishra S.K., Wang S.Y., Lai K.K., Shi J.* Nondifferentiable minimax fractional programming under generalized univexity. - Journal of Computational and Applied Mathematics, 2003, vol. 158, nr. 2, p. 379 - 395.

[377] *Mitchell J.E.* Polynomial interior point cutting plane methods. - Optimization Methods and Software, 2003, vol. 18, nr. 5, p. 507 - 534.

[378] *Mitchell J.E., Todd M.J.* Solving combinatorial optimization problems using Karmarkar's algorithm. - Mathematical Programming, 1992, vol. 56, p. 245 - 284.

[379] *Mizuno S., Kojima M., Todd M.J.* Infeasible-interior-point primal-dual potential-reduction algorithms for linear programming. - SIAM Journal on Optimization, 1995, vol. 5, p. 52 - 67.

[380] *Mjelde K.M.* An incremental and parametrical algorithm for convex-concave fractional programming with a single constraint. - European Journal of Operations Research, 1986, vol. 23, nr. 3, p. 391 - 395.

[381] *Mond B.* On algorithmic equivalence in linear fractional programming. - Mathematics of Computation, 1981, vol. 37, nr. 155, p. 185 - 187.

[382] *Mond B.* On fractional programming and equivalence. - Naval Research Logistics Quarterly, 1975, vol. 22, p. 405 - 410.

[383] *Mond B., Craven B.D.* Nonlinear fractional programming. - Bull. Austral. Math. Soc., 1975, vol. 12, nr. 3, p. 391 - 397.

[384] *Mond B., Weir T.* Duality for fractional programming with generalized convexity conditions. - J. Inform. Optim. Sci., 1982, vol. 3, p. 105 - 124.

[385] *Mukherjee R.N., Rao Ch.P.* Multiobjective fractional programming under generalized invexity. - Indian J. Pure Appl. Math., 1996, vol. 27, p. 1175 - 1183.

[386] *Mukherjee R.N.* Generalized convex duality for multiobjective fractional programs. - Journal of Mathematical Analysis and Applications, 1991, vol. 16, nr. 2, p. 309 - 316.

[387] *Munteanu E., Rado F.* Calculul sarjelor celor mai economice la cuptoarele de topit fonta. - Studii si cercetari Matematice (Cluj) XI, fascicola anexa, 1960, p. 149 - 158.

[388] *Murty K.S.* Linear programming. - John Wiley and Sons, New York, 1983.

[389] *Nemirovskii A., Yudin D.* Problem complexity and method efficiency in optimization. - John Wiley and Sons, New York, 1983.

[390] *Nemirovskii A.S.* On polynomiality of the method of analytic centers for fractional problems. - Mathematical Programming, 1996, vol. 73, nr. 2, p. 175 - 198.

[391] *Nemirovskii A.S.* The long-step method of analytic centers for fractional problems. - Mathematical Programming, 1997, vol. 77, nr. 2, p. 191 - 224.

[392] *Nesterov Yu.* Semidefinite relaxation and nonconvex quadratic optimization. - Optimization method and Software, Special Issue. Febr., 1998, p. 141 - 160.

[393] *Nesterov Yu., Nemirovskii A.* Interior point polynomical algorithms in convex programming. - SIAM Studies in Applied Mathematics, vol. 13, 1993.

[394] *Nesterov Yu., Nemirovskii A.S.* Interior-point polinomial methods in convex programming: Theory and Applications. - SIAM, 1993.

[395] *Nesterov Yu.E., Nemirovskii A.S.* An interior-point method for generalized linear-fractional problems. - Mathematical Programming, 1995, vol. 69, nr. 1 - 3, p. 117 - 204.

[396] *Nesterov Yu.E., Nemirovskii A.S.* Interior point polynomial algorithms in convex programming. - Studies in Applied Mathematics, Philadelphia, 1994.

[397] *Neumann J. Morgenstern O.* Theory of games and economic behavior. - Princeton, New Jersey: Princeton University Press, 1944.

[398] *Nobakhtian S.* Optimality and duality for nonsmooth multiobjective fractional programming with mixed constraints. - Journal of Global Optimization, 2008, vol. 41, nr. 1, p. 103 - 115.

[399] *Nykowski I., Zolkiewski Z.* A compromise procedure for the multiple objective linear fractional programming problem. - European Journal of Operational Research, vol. 19, nr. 1, 1985, p. 91 - 97

[400] *Ohlendorf E., Tammer Ch.* Multicritera fractional programming - an approach by means of conjugate functions. - OR Spektrum, 1994, vol. 16, p. 249 - 254.

[401] *Ohlendorf E., Tammer Ch.* Multicriteria fractional programming - an approach by means of conjugate functions. - OR Spektrum, 1994, vol. 16, p. 249 - 254.

[402] *Pardalos P.M.*Complexity in numerical optimization. - World Scientific, 1993.

[403] *Pardalos P.M.* Polynomial time algoritms for some classes of constrained quadratic problems. - Optimization, 1990, vol. 21, p. 843 - 853

[404] *Patkar V., Stancu-Minasian I.M.* Recent results in disjunctive linear fractional programming. - Lecture Notes in Economics and Mathematical Systems, vol. 345. - Springer-Verlag, 1990, p. 99 - 105.

[405] *Peng J., Yuan Y.* Optimality conditions for the minimizations of a quadratic with two quadratic constraints. - SIAM Journal on Optimization, 1997, vol. 7, p. 579 - 594.

[406] *Penot J.P.* Characterization of solution sets of quasiconvex programs. - Journal of Optimization Theory and Applications, 2003, vol. 117, p. 627 - 636.

[407] *Phuong N.T.H., Tuy H.* A unified monotonic approach to generalized linear fractional programming. - Journal of Global Optimization, 2003, vol. 26, nr. 3, p. 229 - 259.

[408] *Poljak B.T.* Introduction to optimization, optimization software. - New York, NY, 1987.

[409] *Poljak B.T.* Subgradient methods: a survey of Soviet research. - Nonsmooth Optimization, Pergamon Press, 1978.

[410] *Polak E., Mayne D.Q., Wardi Y.* On the extension of constrained optimization algorithm from differentiable to nondifferentiable problems. - SIAM J. Control Optim., 1983, vol. 21, p.179 - 203.

[411] *Powell M.J.D., Yuan Y.* A recursive quadratic programming algorithm that uses differentiable penalty functions. - Mathematical Programming, 1986, vol. 7, p. 265 - 278.

[412] *Preda V.* On duality of multiobjective fractional measurable subset selection problems. - Journal of Mathematical Analysis and Applications, 1995, vol. 196, p 514 - 525.

[413] *Prekopa A.* Stochastic linear programming. - Kluwer Academic Publishers, Dordrecht, 1995.

[414] *Radzik T.* Fractional combinatorial optimization. - Handbook of Combinatorial Optimization, Kluwer Academic Publishers: Dordrecht-Boston-London, 1998, p. 429 - 478.

[415] *Reddy L.V., Mukherjee R.N.* Some results on mathematical programming with generalized ratio invexity. - Journal of Mathematical Analysis and Applications, 1999, vol. 240, nr. 2, p. 299 - 310.

[416] *Renegar J.* A polynomial time algorithm, besed on Newton's method, for linear programming. - Mathematical Programming, 1987, vol. 40, p. 59 - 93.

[417] *Rockafellar R.T.* Convex analysis. - Princeton University Press, Princeton, New Jersey, 1970.

[418] *Rockafellar R.T.* Network flous and monotropic optimization. - Wiley, New York, 1984.

[419] *Roland W.F., Jarre F.* An interior-point method for fractional programs with convex constraints. - Mathematical Programming, 1994, vol. 67, p. 407 - 440.

[420] *Roos C., Vial J.P.* A polynomial method of approximate centers for linear programming. - Mathematical Programming, 1992, vol. 54, p. 295 - 305.

[421] *Rosen J.B.* Convex partition programming. - Recent advances in matematical programming. - McGraw-Hill Inc., New York, 1963, p. 159 - 176.

[422] *Rosen J.B.* Primal partition programming for blok diagonal matrices. - Numerische Mathematik, 1964, vol. 6, p. 250 - 260.

[423] *Roubi A.* Method of centers for generalized fractional programming. - Journal of Optimization Theory and Applications, 2000, vol. 107, nr. 1, p. 123 - 143.

[424] *Rozen J.B.* Primal partition programming for block diagonal matrices. - Numerische Mathematik, 1964, vol. 6, nr. 3, p. 250 - 260.

[425] *Rubinov A.M.* Abstract convexity and global optimization. - Kluwer Academic Publishers: Dordrecht, 2000.

[426] *Rubinov A.M., Glover B.M.* Quasiconvexity via two step functions. - Generalized convexity and generalized monotonicity. - Kluwer Academic Publishers, Dordrecht, 1998, p.159 - 184.

[427] *Ryoo H.S., Sahinidis N.V.* A branch-and-reduce approach to global optimization. - Journal of Global Optimization, 1996, vol. 8, p. 107 - 138.

[428] *Ryoo H.S., Sahinidis N.V.* Global optimization of multiplicative programs. - Journal of Global Optimization, 2003, vol. 26, p. 387 - 418.

[429] *Sakthivel S., Ramraj E.* A new approach to solve linear fractional programming programs. - The Mathematics Education, 2005, vol. 40, p. 1 - 8.

[430] *Sawaragi Y., Nakayama H., Tanino T.* Theory of multiobjective optimization. - Academic Press, New York, 1985.

[431] *Schaible S.* A note on the sum of a linear and linear-fractional function. - Naval Res. Logist., 1977, vol. 24, nr. 4, p. 691 - 693.

[432] *Schaible S.* A survey of fractional programming. - Generalized convexity in optimization and economics. - Academic Press, New York, 1981, p. 417 - 440.

[433] *Schaible S.* Bibliography in fractional programming. - Mathematical Method of Operations Research, 1982, vol. 26, nr. 7, p. 211 -241.

[434] *Schaible S.* Duality in fractional programming: a unified approach. - Operations Research, 1976, vol. 24, nr. 3, p. 452 - 461.

[435] *Schaible S.* Fractional programming - I, duality. - Management Science, 1976, vol. 22, nr. 8, p. 858 - 867.

[436] *Schaible S.* Fractional programming - II, on Dinkelbach's algorithm. - Management Science, 1976, vol. 22, nr. 8, p. 868 - 873.

[437] *Schaible S.* Fractional programming - some recent development. - J. Inform. Optim. Sci. (India), 1989, vol. 10, nr. 1, p. 1 - 14.

[438] *Schaible S.* Fractional programming. - Handbook of global optimitation. - Kluwer Academic Publishers, Dordrecht-Boston-London, 1995, p. 495 - 608.

[439] *Schaible S.* Fractional programming. - Mathematical Methods of Operations Research, 1983, vol. 27, nr. 1, p. 39 - 54.

[440] *Schaible S.* Fractional programming: applications and algorithms. - European Journal of Operational Research, 1981, vol. 7, nr. 2, p. 111 - 120.

[441] *Schaible S.* Minimization of ratios. - Journal of Global Optimization Theory and Aplications, 1976, vol. 19, nr. 2, p. 347 - 352.

[442] *Schaible S.* Multi-ratio fractional programming - a survey. - Optimization, Parallel Processing and Aplications. - Lecture Notes in Economics and Mathematical Sciences. - Springer-Verlag, Berlin, vol. 304, 1988, p. 57 - 66.

[443] *Schaible S.* Generalized monotonicity - a survey. - Generalized convexity and generalized monotonicity. - Lecture Notes in Economics and Mathematical Systems. - Berlin: Springer-Verlag, 1994, vol. 405, p.229 - 249.

[444] *Schaible S., Ibaraki T.* Fractional programming. - European Journal of Operations Research, 1983, vol. 12, nr. 4, p. 325 - 338.

[445] *Schaible S., Shi J.* Fractional Programming: the sum of ratio case. - Optimization Methods and Software, 2003, vol. 18, nr. 2, p. 219 - 229.

[446] *Schaible S., Shi J.* Recent developments in fractional programming: single ratio and max-min case. - Proceeding of Nonlinear Analysis and Convex Analysis, 2004, p. 493 - 506.

[447] *Schaible S., Sodini C.* Finite alghorithm for generalized linear multiplicative programming. - Journal of Optimization Theory and Applications, 1995, vol. 87, p. 41 - 55.

[448] *Schaible S., Ziemba W.T. (Eds.)* Generalized Concavity in Optimization and Economics. - Proceedings of the "First Conference on Generalized Convexity held at University of British Columbia, Vancouver (British Columbia, Canada), August 4 - 15, 1980. - Academic Press, New York, 1981

[449] *Schechter M.* An extension of the Charnes-Cooper method in linear fractional programming. - J. Inform. Optim. Sci. (India), 1989, vol. 10, nr. 1, p.97 - 104.

[450] *Scott C.H., Jefferson T.R.* Conjugate duality for fractional programs. - J. Math. Anal. Appl., 1981, vol. 84, nr. 2, p. 381 - 389.

[451] *Scott C.H., Jefferson T.R.* Conjugate duality in generalized fractional programming. - Journal of Optimization Theory and Applications, 1989, vol. 60, nr. 3, p. 475 - 483.

[452] *Scott C.H., Jefferson T.R.* Convex dualy for quadratic concave fractional programs. - Journal of Optimization Theory and Applications, 1996, vol. 91, nr. 1, p. 115 - 122.

[453] *Scott C.H., Jefferson T.R.* Fractional programming duality via geometric programming duality. - J. Austral. Math. Soc., 1980, vol. 21, seria B, p. 398 - 401.

[454] *Scott C.H., Jefferson T.R., Frenk J.B.J.* A duality theory for a class of generalized fractional programs. - Journal of Global Optimization, 1998, vol. 12, n. 3, p. 239 - 245.

[455] *Sekitani K., Shi J., Yamamoto Y.* General fractional programming: min-max convex-convex quadratic case. - APORS'94 - Development in Diversity and Harmony, World Scientific, 1995, p. 505 - 514.

[456] *Sekitani K., Shi J., Yamamoto Y.* Generalized fractional programming: min-max convex-convex dquadratic case. - In: APORS-Development in Diversity and Harmony, Word Scientific, 1995, p. 505 - 514.

[457] *Seshan C.R.* On duality in linear fractional programming. - Proc. Indian Acad. Sci., Sect. A, Math. Sci., 1980, vol. 62, nr. 1, p. 35 - 42.

[458] *Seshan C.R., Tikekar V.G.* Algorithms for integer fractional programming. - Journal of the Indian Institute of Science, 1980, vol. 62, no. 2, p. 9 - 16.

[459] *Shapiro J.F.* Mathematical programming: Structure and algorithms. - John Wiley and Sons, New York, NY, 1979.

[460] *Sharma I.C., Swarup K.* On duality in linear fractional functionals programming. - Mathematical Method of Operations Research, 1972, vol. 16, nr. 3, p. 91 - 100.

[461] *Shen P.P., Duan Y.P., Pei Y.G.* A simplicial branch ang duality bound algorithm for the sum of convex-convex ratios problem. - Journal of Computational and Applied Mathematics, 2009, vol. 223, nr.1, p.145 - 158.

[462] *Shen P.P., Wang C.F.* Global optimization for sum of generalized fractional functions. - Journal of Computational and Applied Mathematics, 2008, vol. 214, nr. 1, p. 1 - 12.

[463] *Shen P.P., Yuan G.X.* Global optimization for the sum of generalized polynomial fractional functions. - Math. Meth. Oper. Res., 2007, vol. 65, nr. 3, p. 445 - 459.

[464] *Sherali H.D.* Global Optimization of nonconvex polynomial programming problems having rational exponents. - Journal of Global Optim., 1998, vol. 2, p. 267 - 283.

[465] *Sherali H.D. Alameddine A.R.* A new reformulation - linearization technique for bilinear programming problems. - Journal of Global Optimization, 1992, vol. 2, p. 379 - 410.

[466] *Shi J.* A combined algorithm for fractional programming. - Annals of Operations Research, 2001, vol. 103, p. 135 - 147.

[467] *Shor N.Z.* Dual estimates in multiextremal problems. - Journal of Global Optimization, 1992, nr. 2, p. 411 - 418.

[468] *Shor N.Z.* Generalized gradient methods of nondifferentiable optimization employing space dilatation operations. - Berlin: Springer-Verlag, 1982, p. 501 - 529.

[469] *Shor N.Z.* Minimization method for non-differentiable functions. - Berlin: Springer-Verlag, 1985.

[470] *Shor N.Z.* Nondifferentiable optimization and polynomial problems. - Boston; Dordrecht; London: Kluwer Academic Publishers, 1998.

[471] *Shor N.Z., Stetsyuk P.I.* Lagrangian bounds in multiextremal polynomial and discrete optimization problems. - Journal of Gobal Optimization, 2002, vol. 23, p. 1 - 41.

[472] *Sideri E.A.* A cutting plane algorithm for min-max fractional programming. - J. Inform. Optim. Sci. (India), 1989, vol. 10, nr. 1, p. 177 - 192.

[473] *Singh C.* Optimality conditions for fractional minmax programming. - J. Math. Anal. Appl., 1984, vol. 100, nr. 2, p. 409 - 415.

[474] *Singh C.* Optimality conditions in fractional programming. - Journal of Optimization Theory and Applications, 1981, vol 33, nr. 2, p. 287 - 294.

[475] *Singh C., Dass B.K.* Continuous-time, fractional and multiobjective programming. - Proceedings of the "Second Conference on Generalized Convexity held at St. Lawrence University, Canton (New York, U.S.A.), July 29 - August 1, 1986. - Analytic Publishing Co., Delhi, 1989. - Journal of Information and Optimization Sciences, 1989, vol. 10, nr. 1.

[476] *Singh C., Rueda N.* Generalized fractional programming: optimality and duality theory. - Journal of Optimization Theory and Applications, 1990, vol. 66, nr. 1, p. 149 - 159.

[477] *Singh C.* Convex programming with set-inclusive constraints and its applications to generalized linear and fractional programming. - Journal of Optimization Theory and Applications, 1982, vol. 38, nr. 1, p. 33 - 42.

[478] *Sniedovich M.* A new look at fractional programming. - Journal of Optimization Theory and Applications, 1987, vol. 54, nr. 1, p. 113 - 120.

[479] *Sniedovich M.* Fractional programming revisited. - European Journal of Operational Research, 1988, vol. 33, nr. 3, p. 334 - 341.

[480] *Sodini C.* Equivalence and parametric analysis in linear fractional programming. - Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, 1990, vol. 345, p. 143 - 154.

[481] *Solomon D.* Modelarea matematica in transportul auto. - Modelare matematica, optimizare si tehnologii informationale. - Chisinau, Evrica, 2008, p. 9 - 28.

[482] *Solomon D.* Problema de optimizare fractionar-polinomiala. - Analele ATIC-2001, Vol I (I), Chisinau, Evrica, 2002, p. 9 - 13.

[483] *Solomon D.* Probleme fractionare de prigramare stocastica in doua etape. - Analele ATIC - 2007-2008. - Chisinau, Evrica, 2009, p. 3 - 15.

[484] *Solomon D., Pluta V.* Algorithms for generalized fractional programming. - Computer Sciences Journal of Moldova, 1995, vol. 3, nr. 1, p. 93 - 116.

[485] *Solow D.* Linear programming: An introduction to finite improvement algorithms. - North-Holland, Amsterdam, the Netherlands, 1984.

[486] *Stahl J.* On the decomposition of fractional programming problem. - Szigma (Hungary), 1982, vol. 15, nr. 4, p. 289 - 292.

[487] *Stancu-Minasian I. M., Tigan S. Lozovanu D.D., Solomon D.I.* On a parametrical method for solving bictiterion max-min fractional problems. - Buletinul Academiei de Stiinte a Republicii Moldova. - Matematica, 1992, vol. 3, nr. 9, p. 56 - 61.

[488] *Stancu-Minasian I.M.* A fifth bibliography of fractional programming. - Optimization, 1999, vol. 45, nr. 1 - 4, p. 343 - 367.

[489] *Stancu-Minasian I.M.* A fourth bibliography of fractional programming: 1977 - 1981. - Optimization, 1982, vol. 23, nr. 1, p. 53 - 71.

[490] *Stancu-Minasian I.M.* A second bibliography of fractional programming: 1977-1981. - Pure Appl. Math. Sci., 1983, vol. 17, nr. 1 - 2, p. 87 - 102.

[491] *Stancu-Minasian I.M.* A sixth bibliography of fractional programming. - Optimization, 2006, vol. 55, nr. 4, p. 405 - 428.

[492] *Stancu-Minasian I.M.* Bibliography of fractional programming: 1960 - 1976. - Pure and applied mathematica sciences, 1981, vol. 13, nr. 1 - 2, p. 35 - 69.

[493] *Stancu-Minasian I.M.* Fractional Programming: Theory, Methods and Applications. - Kluwer Academic Publishers, Boston, 1997.

[494] *Stancu-Minasian I.M.* Metode de rezolvare a problemelor de programare fractionala. - Editura Academiei Romane, Bucuresti, 1992.

[495] *Stancu-Minasian I.M.* A third bibliography of fractional programming. - Pure and applied mathematica sciences, 1985, vol. 22, nr. 1 - 2, p. 109 - 122.

[496] *Stancu-Minasian I.M., Solomon D.I.* Metode de descompunere in programarea fractionala separabila. - Studii si Cercetari de Calcul Economic si Cibernetica Economica, 1994, vol. 28, nr. 1, p. 17 - 25.

[497] *Stancu-Minasian I.M., Tigan S.* Fractional programming under uncertainty. - Generalized convexity and generalized monotonicity. - Lecture Notes in Economics and Mathematical Systems. - Berlin: Springer-Verlag, 1994, vol. 405, p.322 - 333.

[498] *Stancu-Minasian I.M., Tigan* On some methods for solving fractional programming problems with inexact data. - Stud. Cerc. Mat., 1993, vol. 45, nr. 6, p. 517 - 532.

[499] *Stancu-Minasian I.M., Tigan S.* Methods for solving stochastic bilinear fractional max-min problem. - RAIRO/ Recherche Operationnelle/ Operations Research, 1996, vol. 30, nr. 1, p. 81 - 98.

[500] *Stancu-Minasian I.M., Tigan S.* The stochastic max-min problem. - Methods of Operations Research, 1984, vol.51, p. 119 - 126.

[501] *Stefanov S.M.* Convex quadratic minimization subject to a linear constraint and box constraints. - Applied Mathematics Research Express, 2004, nr. 1, p. 17 - 42.

[502] *Stefanov S.M.* Separable Programming. Theory and Methods. -Dordrecht: Kluwer Academic Publishers, vol. 53, 2001.

[503] *Steuer R.E.* Multiple criteria optimization. Theory, computation and application. - John Wiley: New York, 1985.

[504] *Strang G.* Introduction to applied mathematics. - Wellesley-Cambridge Press, Massachusetts, 1988.

[505] *Strodiot J.J., Crouzeix J.P., Ferland J.A., Nquyen V.H.* An inexact proximal point method for solving generalized fractional programming. - Journal of Global Optimization, 2008, vol. 42, nr. 1, p. 121 - 138.

[506] *Swarup K.* Duality for transportation problem in fractional programming. - Cahiers Centre Etudes Recherche Oper., 1968, vol. 10, nr. 1, p. 46 - 54.

[507] *Swarup K.* Duality in fractional programming. - Mathematical Method of Operations Research, 1968, vol. 12, nr. 2, p. 106 - 112.

[508] *Swarup K.* Linear fractional functionals programming. - Operations Res., 1965, vol. 13, nr. 6, p. 1029 - 1036.

[509] *Swarup K.* On varying all the parameters in a linear fractional functional programming problem. - Metrika, 1968, vol. 13, nr. 2/3, p. 196 - 205.

[510] *Swarup K.* Programming with quadratic fractional functional. - Opsearch, 1965, vol. 2, nr. 3 - 4, p. 23 - 30.

[511] *Swarup K.* Some aspects of duality in linear fractional programming. - ZAMM, 1967, vol. 47, p. 204.

[512] *Swarup K.* Transportation technique in linear fractional programming. - J.R.N.S.S., 1966, vol. 21, nr. 5, p. 256 - 260.

[513] *Swarup K.* Some aspects of linear fractional functionals programming. - Austral. J. Statist., 1966, vol. 7, nr. 3, p. 90 - 104.

[514] *Tanaka K., Maruyama Y.* The multiobjective optimization problems of set functions. - Journal of Information and Optimization Sciences, 1984, vol. 5, p. 293 - 306.

[515] *Tawarmalani M., Sahinidis N.V.* Semidefinite relaxations of fractional programs via novel convexification techniques. - Journal of Global Optimization, 2001, vol. 20, nr.2, p.133 - 154.

[516] *Thie P.R.* An introduction to linear programming and game theory. - John Wiley and Sons, New York, NY, 1988.

[517] *Thoai N.V.* Convergence and applications of a decomposition method using duality bounds for nonconvex global optimization. - Journal of Optimization Theory and Applications, 2002, vol. 113, p.165 - 193.

[518] *Thoai N.V.* Duality bound method for the general quadratic programming problem with quadratic constraints. - Journal of Optimization Theory and Applications, 2000, vol. 107, p. 331 - 354.

[519] *Thoai N.V.* On duality bound method in partly convex programming. - Journal of Global Optimization, 2002, vol. 22, p. 263 - 270.

[520] *Todd M.J.* Improved bound and containing ellipsoids in Karmarkar's linear programming algorithm. - Math. Oper. Res., 1988, vol.13, p. 650 - 659.

[521] *Tuy H.* Convex analysis and global optimization. - Kluwer Academic Publishers: Dordrecht, 1998.

[522] *Tuy H.* On nonconvex optimization problems with separated nonconvex variables. - Journal of Global Optimization, 1992, vol. 2, p. 133 - 144.

[523] *Tuy H., Thach P.T., Konno H.* Optimization of polynomial fractional functions. - Journal of Global Optimization, 2004, vol. 29, nr. 1, p. 19 - 44.

[524] *Vajda S.* Mathematical programming. - Addison-Wesley Publishing Company, Reading, Massachusetts, 1961.

[525] *Varma G.K.* General parametric linear fractional programming. - Metrika, 1972, vol. 19, nr. 1, p. 11 - 17.

[526] *Vartak M.N., Gupta I.* Duality theory for fractional programming problems under ?-convexity. - Opersearch (India), 1987, vol. 24, nr. 3, p. 163 - 174.

[527] *Vavasis S.A.* Approximation algorithms for indefinite quadratic programing. - Mathematical Programming, 1992, vol. 57, p. 279 - 311.

[528] *Vidon L., Kaur S.S.* Decompozition in generalized fractional programming and its optimal dual solution. - Indian J. Pure Appl. Math., 1987, vol. 18, p. 973 - 978.

[529] *Wadhwa V.* Linear fractional programs with variable coefficients. - Cahiers Centre Etudes Recherche Oper., 1972, vol. 14, p. 223 - 232.

[530] *Wadhwa V.* Programming with separable fractional functionals. - Journal of Mathematical Sciences, 1969, vol. 4, p. 51 - 60.

[531] *Wagner H.M., Yuan J.S.C.* Algorithmic equivalence in linear fractional programming. - Management Science, 1968, vol. 14, nr. 5, p. 301 - 306.

[532] *Walk M.* Theory of duality in mathematical programming. - Springer-Verlag, Wien / New York, 1989.

[533] *Wang Y.J., Shen P.P., Liang Z.* A branch-and-bound algorithm to globally solve the sum of several linear ratios. - Appl. Math. Comput., 2005, vol. 168, nr. 1, p. 89 - 101.

[534] *Wanka G., Bot R.I.* Multiobjective duality for convex ratios. - J. Math. Anal. Appl., 2002, vol. 275, nr. 1, p. 354 - 368.

[535] *Warburton A. R.* Parametric solution of bicriterion linear fractional programs. - Oper. Res., 1985, vol. 33, nr. 1, p. 74 - 84.

[536] *Weir T.* A dual for a multiobjective fractional programming problem. - J. Inform. Opt. Sci., 1986, vol. 7, pp 261 - 269

[537] *Weir T.* A duality theorem for a multiple objective fractional optimization problem. - Bull. Austral. Math. Soc., 1986, vol. 34, p. 415 - 425.

[538] *Weir T.* On duality in multiobjective fractional programming. - Opsearch., 1989, vol. 26, p. 151 - 158.

[539] *Weir T.* Symmetric dual multiobjective fractional programming. - J. Austral. Math. Soc. Ser. A, 1991, vol. 50, p. 67 - 74.

[540] *Weir T.* Pseudoconvex minimax programming. - Utilitas Math., 1992, vol. 42, p. 234 - 240.

[541] *Werner J.* Duality in generalized fractional programming. - Internat. Ser. Numer. Math., 1988, vol. 84, p. 341 - 351.

[542] *Williams A.C.* A stochastic transportation problem. - Operations Research, 1963, vol. 11, nr. 5, p. 759 - 770.

[543] *Williams A.C.* A treatment of transportation problems by decomposition. - J. Soc. Ind. Appl. Math., 1962, vol. 10, nr. 1, p. 35 - 48.

[544] *Winston W.* Operations Research: Applications and Algorithms. - Duxbury Press; 4th. Edition, 2003.

[545] *Wolf H.* A parametric method for solving the linear fractional programming problem. - Operation Research, 1985, vol. 33, nr. 4, p. 835 - 841.

[546] *Wolf H.* Parametric analysis in linear fractional programming. - Oper. Res., 1986, vol. 34, nr. 6, p. 930 - 937.

[547] *Wolfe P.* The simplex method for quadratic programming. - Econometrica, 1959, nr. 27, p. 382 - 398.

[548] *Wright S.I.* Primal-dual interior-point methods. - SIAM, Philadelphia, PA, 1996.

[549] *Wu W.Y., Sheu R.L., Birbil S.I.* Solving the sum of ratios problem by a stochastic search algorithm. - Journal of Global Optimization, 2008, vol. 42, nr. 1, p. 91 - 109.

[550] *Xu Z.K.* Duality in generalized nonlinear fractional programming. - Journal of Mathematical Analysis and Aplications, 1992, vol. 169, nr. 1, p. 1 - 9.

[551] *Xu Z.K.* Saddle-point type optimality criteria for generalized fractional programming. - Journal of Optimization Theory and Aplications, 1988, vol. 57, nr. 1, p. 189 - 196.

[552] *Yadav S.R., Mukherjee R.N.* Duality for fractional minimax programming problems. - Journal of Australian Mathematical Society, 1990, vol. 31, p. 484 - 492.

[553] *Yang X.M., Hou S.H.* On minimax fractional optimality and duality with generalized convexity. - Journal of Global Optimization, 2005, vol.31, nr.2, p. 235 - 252

[554] *Yang X.M., Yang X.Q., Teo K.L.* Duality and saddle-point type optimality for generalized nonlinear fractional programming. - Journal of Mathematical Analysis and Aplications, 2004, vol. 289, nr. 1, p. 100 - 109.

[555] *Yanjun W., Peiping S., Zhian L.* A branch-and-bound algorithm to globally solve the sum of several linear ratios. - Applied Mathematics and Computation, 2005, vol. 168, Issue 1, p. 89 - 101.

[556] *Yao D.D., Zhang S., Zhou X.Y.* Stochastic LQ control via semidefinite programmimg. - SIAM Journal on Control and Optmization, 2001, vol. 40, p. 801 - 823.

[557] *Ye Y.* On an affine scaling algorithm for nonconvex quadratic programming. - Mathematical Programming, 1992, vol. 56, p. 285 - 300.

[558] *Ye Y., Tse E.* An extension of Karmarkar's projective algorithm for convex quadratic programming. - Math. Programming, 1989, vol. 44, p. 157 - 180.

[559] *Ye Y., Zhang S.* New results on quadratic minimization. - SIAM Journal on Optimization, 2003, vol. 14, p. 245 - 267.

[560] *Yuan D.H., Liu X.L., Chinchuluun A., Pardalos P.M.* Nondiferentiable minimax fractional programming with (c, a, ?, d)-convexity. - Journal of Optimization Theory and Applications, 2006, vol. 126, nr. 1, p. 185 - 199.

[561] *Zalmai G.J.* Continuous-time multiobjective fractional programming. - Optimization, 1996, vol. 37, nr. 1, p. 1 - 25.

[562] *Zalmai G.J.* Duality for generalized fractional programs involving n-set functions. - J. Math. Anal. Appl., 1990, vol. 149, p. 339 - 350.

[563] *Zalmai G.J.* Optimality conditions and duality for a class of continuous-time generalized fractional programming problems. - J. Math. Anal. Appl., 1990, vol. 153, p. 356 - 371.

[564] *Zalmai G.J.* Optimality conditions and duality models for a class of nonsmooth constrained fractional optimal control problems. - Journal of Mathematical Analysis and Applications, 1997, vol. 210, p. 114 - 149.

[565] *Zalmai G.J.* Optimality conditions and duality models for generalized fractional programming problems containing locally subdifferentiable and ?-convex functions. - Optimization, 1995, vol. 32, p. 95 - 124.

[566] *Zalmai G.J.* Optimality principles and duality models for a class of continuous-time generalized fractional programming problems with operator constraints. - J. Stat. Mnag. Syst., 1998, vol. 1, p. 61 - 100.

[567] *Zalmai G.J.* Saddle points and Lagrangian-type duality for discrete minmax fractional subset programming problems with generalized convex functions. - Journal of Mathematical Analysis and Applications, 2006, vol. 313, p. 484 - 503

[568] *Zalmai G.J.* Saddle-point-type optimality conditions and Lagrangian-type duality for a class of constrained generalized fractional optimal control problems. - Optimization, 1998, vol. 44, p. 351 - 372.

[569] *Zang D., Shi J., Wang S.* Conical partition algorithm for maximizing the sum of ratios. - Journal of Global Optimization, 2005, vol. 31, nr. 2, p. 253 - 270.

[570] *Zangwill W.I.* Nonlinear programming via penalty functions. - Management Sci., 1967, vol. 13, p. 344 - 358.

[571] *Zangwill W.I.* Nonlinear programming: A unified approach. - Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969.

[572] *Zhang S.* Quadratic maximization and semidefinite relaxation. - Mathematical Programming, 2000, vol. 87, p. 453 - 465.

[573] *Zhang S., Huang Y. W.* Complex quadratic optmization and semidefinite programming. - SIAM Journal on Optmization, 2006, vol.16, p. 871 - 890.

[574] *Zhang S., Ye Y.* New results on quadratic minimization. - SIAM Journal on Optmization, 2003, vol. 14, p. 245 - 267.

[575] *Zhao G.* On the choice of parameter for power-series interior point algorithms for linear programming. - Mathematical Programming, 1995, vol. 68, p. 49 - 71.

[576] *Zhou H., Sun W.*] Mixed duality without a constraint qualification for minimax fractional programming. - Optimization, 2003, vol. 52, nr. 4 - 5, p. 617 - 627.

[577] *Zionts S.* Programming with linear fractional functionals. - Naval Res. Logistics Quart, 1968, vol. 15, nr. 3, p. 449 - 451.

[578] *Базара М., Шетти К.* Нелинейное программирование. Теория и алгоритмы. Пер. с англ. - М: Мир, 1982.

[579] *Бакаев А.А., Михалевич В.С., Брановицкая С. В., Шор Н.З.* Методика и опыт решения сетевых транспортных задач большого объема на ЭЦВМ. - Математические методы и проблемы производства, М., 1963.

[580] *Белинький А.С.* Минимаксные задачи планирования с линейными ограничениями и методы их решения. - Автоматика и телемеханика, 1981, № 10, с. 157 - 170.

[581] *Беллман Р.* Динамическое программирование. Пер. с англ. - М: И.Л., 1960.

[582] *Белых В.М., Гавурин М.К.* Алгоритм минимизации дробно-линейной функции. - Весник ЛГУ, 1980, № 19, с. 10 - 15.

[583] *Беляева Л.В., Журбенко Н.Г., Шор Н.З.* О методе решения одного класса динамических распределительных задач. - Экономика и математические методы, 1978, том 14, вып. 1, с. 137 - 146.

[584] *Бердсекас Д.* Условия оптимизации и методы множителей Лагранжа. - М.: Радио и связъ, 1987.

[585] *Быкадоров И.А.* Об условиях квазивыпуклости сумм дробно-линейных функций. - Оптимизация, 1986, вып. 39 (56), с. 25 - 41.

[586] *Вагнер Г.* Основы исследования операций. Пер. с англ. в трех томах.- М: Мир, 1972 - 1973.

[587] *Вентцель Е.С.* Исследование операций. - М: Сов. радио, 1972.

[588] *Вентцель Е.С.* Исследование операций. Задачи, принципы, методологии.- М: Наука, 1988.

[589] *Вертгейм Б.А., Рубинштейн Г.Ш.* К определению квазивыпуклых функций. - Математическое программирование. - М., 1966, с. 121 - 134.

[590] *Гаврин М.К.* Дробно-линейное программирование на неограниченном множестве. - Вестник ЛГУ, 1982, № 19, с. 12 - 16.

[591] *Гамецкий А.Ф., Соломон Д.И.* Асимтотические решения задачи дробно-линейного программирования. - Analele ATIC-2004, vol. I (VI) - Chisinau, Evrica, 2004, p. 3 - 9.

[592] *Гамецкий А.Ф., Соломон Д.И.* Исследования операций. Том 1. - Кшн.: Эврика, 2004.

[593] *Гамецкий А.Ф., Соломон Д.И.* Исследования операций. Том 2. - Кшн.: Эврика, 2008.

[594] *Гольштейн Е.Г.* Блочный метод выпуклого программирования. - ДАН СССР, 1986, Т. 288, № 1, с. 24 - 27.

[595] *Гольштейн Е.Г.* Двойственные задачи в выпуклом и дробно-выпуклом программировании в функциональном пространстве. - Доклады АН СССР, 1968, т. 172, № 5, с. 1007 - 1010.

[596] *Гольштейн Е.Г.* Двойственные задачи выпуклого и дробно-выпуклого программирования. - Исследования по математическому программированию. - М.: Наука, 1968, с. 10 - 108.

[597] *Гольштейн Е.Г.* Обобщенный градиентный метод отыскания седловых точек. - Экономика и математические методы, 1972, том VIII, вып. 4б с.

[598] *Гольштейн Е.Г.* Теория двойственности в математическом программировании и ее приложения. - М.: Наука, 1971.

[599] *Гольштейн Е.Г., Третьяков Н.В.* Градиентный метод минимизации и алгоритмы выпуклого программирования, связанные с модифицированными функциями Лагранжа. - Экономика и математические методы, 1975, том 11, вып. 4, с. 730 - 742.

[600] *Гольштейн Е.Г., Третьяков Н.В.* Модифицированные функции Лагранжа. - М: Наука, 1989.

[601] *Гольштейн Е.Г., Юдин Д.Б.* Задачи линейного программирования транспортного типа. - М: Наука, 1969.

[602] *Гольштейн Е.Г., Юдин Д.Б.* Новые направления в линейном программировании. - М.: Советское радио, 1966.

[603] *Гольштейн Е.Г.* Методы блочного программирования. - Экономика и математические методы, 1966, том 2, вып. 4.

[604] *Гольштейн Е.Г., Третьяков Н.В.* Модифицированные функции Лагранжа. - Экономика и математические методы, 1974, том 10, вып. 3.

[605] *Данциг Дж.* Линейное программирование, его обобщения и приложения. - М.: Прогресс, 1966.

[606] *Демьянов В.Ф.* Минимакс: дифференцируемость по направлениям. - Л.: ЛГУ, 1974.

[607] *Демьянов В.Ф., Васильев Л. В.* Недифференцируемая оптимизация. - М.: Наука, 1981.

[608] *Демьянов В.Ф., Малоземов В. Н.* Введение в минимакс. - М.: Наука, 1972.

[609] *Демьянов В.Ф., Рубинов А. М.* Основы негладкого анализа и квазидифференциального исчисления. - М.: Наука, 1990.

[610] *Дикин И.И.* Итеративное решение задач линейного и квадратичного программирования. - ДАН СССР, 1967, т. 174, с. 747 - 748.

[611] *Дикин И.И.* Итеративные алгоритмы решения задач линейного, квадратичного и выпуклого программирования. - Опыт решения экономических задач математическими методами. - Новосибирск: Наука, 1967, с. 31 - 38.

[612] *Дикин И.И., Зоркальцев В.И.* Итеративное решение задач математического программирования (алгоритмы метода внутренних точек). - Новосибирск: Наука, 1980.

[613] *Евтушенко Ю.Г.* Методы решения экстремальных задач и их применение в системах оптимизации. - М.: Наука, 1982.

[614] *Ермольев Ю.М.* Методы решения нелинейных экстремальных задач. - Кибернетика, 1966, № 4.

[615] *Ермольев Ю.М.* Методы стохастического программирования. - М: Наука, 1981.

[616] *Ермольев Ю.М., Гайворонский А.* Стохастические методы решения минимаксных задач. - Кибернетика, 1983, № 4, с. 92 - 97.

[617] *Ермольев Ю.М., Шор Н.З.* Метод случайного поиска для задач двухэтапного стохастического программирования и его обобщение. - Кибернетика, 1968, № 1, с. 90 - 92.

[618] *Жилинскас А.* Глобальная оптимизация. Аксиоматика статистических моделей, алгоритмы и применения. - Вильнюс: Мокслас, 1986.

[619] *Заботин Я.И., Кораблев А.М., Хабибулин Р.Д.* Об одном обобщении понятия опорного функционала. - Труды 5-й зимней школы по математическому программированию и смежным вопросам. Дрогобыч, 1972. - М., 1973, вып. 5, с. 190 - 203.

[620] *Зайтман А.А., Ненахов Э.И.* О некоторых методах решения задачи на минимакс. - Исследование методах решения экстремаъных задачи. - Киев: ИК АН УССР, 1990, с. 64 - 69.

[621] *Звягина Р.А.* Задачи линейного программирования с матрицами произвольной блочной структуры. - Доклады АН СССР, 1971, т. 196, № 4.

[622] Исследования по линейному и нелинейному программированию. - М: ИЛ, 1962.

[623] *Канторович Л.В.* Математические методы в организации и планирования производства. - Л: ЛГУ, 1939.

[624] *Канторович Л.В.* О перемещение мас. - ДАН СССР, 1942, Т. 37, № 7 - 8, с. 24 - 27.

[625] *Канторович Л.В.* Экономический расчет наилучшего использования ресурсов. - М: Академиздат, 1959.

[626] *Карлин С.* Математические методы в теории игрб программировании и экономики. - М., Мир, 1964.

[627] *Карманов В.Г.* Математическое программирование. - М.: Наука, 1975.

[628] *Клевачев В.И.* О решении задач дробно-линейного программирования. - Кибернетика, 1968, № 6, с. 27 - 31

[629] *Коннов И.В.* Применение метода сопряженных градиентов к минимизации квазивыпуклых функционалов. - Исследования по прикладной математике. - Казань: КГУ, 1984, вып. 12, с. 46 - 58.

[630] *Корнаи И., Липтак Т.* Планирование на двух уровнях. - Применение математики в экономических исследованиях. Том 3. - М.: Мыслъ. 1965.

[631] *Крупицкий А.Е.* Минимизация сумм двух дробно-линейных функций на выпуклом многогранном множестве. - Весник ЛГУ, 1983, № 13, с. 15 - 21.

[632] *Кюнци Г.П., Крелле В.* Нелинейное программирование. - М.: Сов. Радио, 1965.

[633] *Лозовану Д.Д., Соломон Д.И.* Нелинейные оптимизационные задачи. Алгоритмы и сложность. - Кшн.: Эврика, 1996.

[634] *Лотов А.В., Поспелова И.И.* Многокритериальных задачи принятия решений. - М.: МАКС Пресс, 2008

[635] *Лэсдон Л.С.* Оптимизация больших систем. - М.: Наука, 1975.

[636] *Манюров Д.М.* Блочные методы дробно-линейного программирования. - Вопросы математической кибернетики и прикладной математики. Вып. 1. - Баку: Элм, 1975, с. 51 - 71.

[637] Методы и алгоритмы решения транспортной задачи. - Госсатиздат, 1963.

[638] Методы решения общей задачи линейного программирования. - Госсатиздат, 1962.

[639] *Михалевич В.С., Трубин В.А., Шор Н.З.* Оптимизационные задачи производственно-транспортного планирования. Модели, методы, алгоритмы. - М.: Наука, 1986.

[640] *Михалевич В.С., Шор Н.З., Галустова Л.А. и др.* Вычислителъные методы выбора оптималъных проектных решений. - Киев: Наукова думка, 1977.

[641] *Муртаф Б.* Современное линейное программирование. - М: Мир, 1984.

[642] *Нейман Дж., Моргенштерн О.* Теория игр и экономическое поведение. Пер. с англ. - М: Наука, 1970.

[643] *Немировский А.С.* Об одном алгоритма типа Кармаркара. - Изв. АН СССР. Серия Техническая кибернетика, 1987, № 1, с. 105 - 118.

[644] *Немировский А.С., Юдин Д.Б.* Сложность задач и эффективность методов оптимизации. - М.: Наука, 1979.

[645] *Нестеров Ю.Е.* Методы минимизации негладких выпуклых и квазивыпуклых функций. - Экономика и математические методы, 1984, т. 20, № 3, с. 519 - 531.

[646] *Нестеров Ю.Е.* Методы минимизации негладких функций. - Экономика и математические методы, 1984, т. 29, № 3, с. 519 - 531.

[647] *Нестеров Ю.Е.* Эффективные методы в нелинейном программировании. - М.: Радио и связъ, 1989.

[648] *Нестеров Ю.Е., Немировский А.С.* Самосогласование функций и полиномиальные алгоритмы в выпуклом программировании. - М.: Центр. Эконом.-мат.-ин-т АН СССР, 1989.

[649] *Нестерова С.И., Скоков В.А.* Численный анализ программ негладкой безусловной оптимизации. - Экономика и математические методы, 1994, т. 30, № 2, с. 136 - 145.

[650] *Ногин В.Д.* Методы оптимальных решений. - СПб, СПб филиал ГУ - ВШЭ, 2006

[651] *Нурминский Е.А.* Численные методы решения детерминированных и стохастических минимаксных задач. - Киев: Наукова думка, 1979.

[652] *Нурминский Е.А.* Численные методы выпуклой оптимизации. - М. Наука, 1991.

[653] *Петровский А.Б.* Теория принятия решений. - М.: ИЦ Академия, 2009

[654] *Подиновский В.И., Ногин В.Д.* Парето-оптимальные решения многокритериальных задач. - М.: Наука, 1982

[655] *Полунин А.Ф.* Курс математического программирования. - М.: Наука, 1975.

[656] *Поляк Б.Т.* Введение в оптимизацию. - М.: Наука, 1983.

[657] *Пшеничный Б.Н.* Метод линеаризации. - М.: Наука, 1983.

[658] *Пшеничный Б.Н.* Необходимые условия экстремума. - М.: Наука. - 1983.

[659] *Пшеничный Б.Н., Данилин Ю. М.* Численные методы в экстремальных задачах. - М.: Наука, 1975.

[660] *Рокафеллер Р.* Выпуклый анализ. - М.: Мир, 1973.

[661] *Сергиенко И.В.* Матаматические модели и методы решения задач дискретной оптимизации. - Киев: Наукова думка, 1985.

[662] *Сергиенко И.В., Шор Н.З., Трубин В.А. и др.* Пакет прикладных программ для решения задач производственно-транспортного планирования большой размерности. - Кибернетика, 1983, № 3, с. 57 - 71.

[663] *Соломон Д.И.* Декомпозиционные алгоритмы решения обобщенных задач дробно-линейного программирования. - Системы оптимизации и обработки данных. - Мат. исслед., вып. 100. - Кишинев: Штиинца, 1988, с. 133 - 141.

[664] *Соломон Д.И.* Декомпозиционные методы в дробно-линейном программировании. - Автореф. дис. на соискание ученой степени канд. физ.-мат. наук. - Киев, ИК АН УССР, 1985.

[665] *Соломон Д.И.* Декомпозиционные методы в дробно-линейном программировании. - Системы оптимизации и обработки данных. - Мат. исслед., вып. 100. - Кишинев: Штиинца, 1988, с. 115 - 132.

[666] *Соломон Д.И.* Итеративный алгоритм решения обобщенной задачи дробно-линейного программирования. - Моделирование и оптимизация в задачах планирования и управления. - Мат. исслед., вып. 87. - Кишинев: Штиинца, 1985, с. 161 - 164.

[667] *Соломон Д.И.* Математические модели, методы и проблемно-ориентированные системы дробно-линейной оптимизации (на примере задач планирования автомобильных грузовых перевозок). - Автореф. дис. на соискание ученой степени докт. техн. наук. - Киев, ИК им. В.М. Глушкова НАН Украины, 1992.

[668] *Соломон Д.И.* Об одной задаче целочисленного дробно-линейного программирования. - Математические модели и методы оптимизации экономических систем. - Мат. исслед., вып. 72. - Кишинев: Штиинца, 1983, с. 122 - 131.

[669] *Соломон Д.И.* Об одной процедуре преобразования задач математического программирования, имеющие связующие ограничения и переменные. - Математическое моделирование экономических систем. - Мат. исслед., вып. 52. - Кишинев: Штиинца, 1979, с. 199 -205.

[670] *Соломон Д.И.* Об одном методе решения задач дробно-линейного программирования с матрицами блочно-диагоналъной структуры. - Изв. АН МССР. Сер. физ.-техн. и мат. наук, 1979, № 3, с. 68 - 70.

[671] *Соломон Д.И.* Обобщение линейной и дробно-линейной транспортной задачи. - Изв. АН МССР. Сер. физ.-техн. и мат. наук, 1984, № 1, с. 13 - 18.

[672] *Соломон Д.И.* Обобщенные задачи дробно-линейного программирования. - Математическое моделирование экономических систем. - Мат. исслед., вып. 52. - Кишинев: Штиинца, 1979, с. 206 -215.

[673] *Соломон Д.И.* Параметрический метод решения задач дробно-линейного программирования. - Оптимизация и обработка данных. - Мат. исслед., вып. 96. - Кишинев: Штиинца,1987, с. 124 - 134.

[674] *Соломон Д.И.* Применение метода обобщенного градиентного спуска при решении задач дробно-линейного программирования. - Изв. АН МССР. Сер. физ.-техн. и мат. наук, 1979, № 1, с. 7 - 13.

[675] *Соломон Д.И.* Применение схем разложения по переменным с использованием метода обобщенного спуска при решении задач дробно-линейного программирования. - Математические модели и методы оптимизации экономических систем. - Мат. исслед., вып. 82. - Кишинев: Штиинца, 1985, с. 121 - 134.

[676] *Соломон Д.И.* Решение дробно-линейной транспортной задачи методом обобщенного градиентного спуска. - Алгоритмы и программы, 1985, 4, с. 35.

[677] *Соломон Д.И.* Решение дробно-линейной задачи размещения методом обобщенного градиентного спуска. - Алгоритмы и программы, 1985, 4, с. 36.

[678] *Соломон Д.И.* Решение распределительной дробно-линейной задачи методом обобщенного градиентного спуска. - Алгоритмы и программы, 1988, 6, с. 8.

[679] *Соломон Д.И.* Решение сетевой дробно-линейной транспортной задачи параметрическим методом. - Алгоритмы и программы, 1987, 9, с. 36.

[680] *Соломон Д.И.* Матаматическая модель двухзвенной кольцевой маршрутизации. - Применение экономико-математических методов и ВТ планировании и управлении народным хозяйством МССР. - Кишинев, 1988, с. 141 - 142.

[681] *Соломон Д.И.* Комплекс программ решения дробно-линейных транспортных задач субградиентным методом. - Системы программного обеспечения решения задач оптимального планирования. - М., 1988, с. 165- 166.

[682] *Соломон Д.И.* Полиномиальный алгоритм решения задач дробно-линейного программирования. - Методы математического программирования и программное обеспечение. - Свердловск, 1989, с. 195- 196

[683] *Соломон Д.И.* Математические модели решения задач маршрутизации автомобильных грузовых перевозок. - Математическое моделирование и оптимизация. - Мат. исслед., вып. 110. - Кишинев: Штиинца, 1989, с. 107 - 115.

[684] *Соломон Д.И.* Математическая модель построения кольцевых маршрутов. - Изв. АН МССР. Сер. Матеиатика, 1990, № 1, с. 50 - 55.

[685] *Соломон Д.И.* Математическая модель планирования и совершенствования структуры подвижного состава на автомобильном транспорте. - Проблемы математизации народного хозяйства Молдавской ССР. - Кишинев, 1990, с. 122 - 126.

[686] *Соломон Д.И.* Комплекс программ решения задач оптимизации структуры подвижного состава автотранспортных предприятий. - Системы программного обеспечения решения задач оптимального планирования. - М., 1990, с. 174-175.

[687] *Соломон Д.И.* Математические модели решения задач закрепления маршрутов за автотранспортнми предприятиями. - Математическое моделирование экономических систем. - Мат. исслед., вып. 114. - Кишинев: Штиинца, 1990, с. 109 - 119.

[688] *Соломон Д.И., Жеденко З.С., Кроитору Э.В.* Программное обеспечение систем оперативного планирования и управления междугородными автомобильными перевозками. - Оптимизация и обработка данных. - Мат. исслед., вып. 121. - Кишинев: Штиинца, 1991, с. 133 - 138.

[689] *Соломон Д.И., Кранкер П.М., Парницкий В.Е.* Маршрутизация междугородных автомобильных перевозок грузов. - Применение экономико-математических методов и ВТ планировании и управлении народным хозяйством МССР. - Кишинев, 1988, с. 65 - 66.

[690] *Соломон Д.И., Лесник В.И.* Приближенный алгоритм решения задачи маршрутизации с ограниченными условиями. - Проблемы управления и моделирования социально-экономического развития региона. - Свердловск, 1978, с. 76 - 78.

[691] *Соломон Д.И., Лесник В.И.* Система оперативного управления междугородными и межрайонными автомобильными перевозками. - Проблемы управления и моделирования социально-экономического развития региона. - Свердловск, 1980, с. 55 - 58.

[692] *Соломон Д.И., Шаров В.Н.* Оптимизация структуры подвижного состава автотранспортных предприятий. - Оптимизация и обработка данных. - Мат. исслед., вып. 96. - Кишинев: Штиинца, 1987, с. 135 - 145.

[693] *Соломон Д.И., Шаров В.Н., Зоти А.Г.* Моделирование на ЭВМ процесса годового планирования автомобильных перевозок - Модели и алгоритмы решения задач планирования и управления. - Мат. исслед., вып. 68. - Кишинев: Штиинца, 1982, с. 125 - 134.

[694] *Соломон Д.И., Шпак А.Л.* Система моделирования транспортных сетей и расчета кратчайших расстояний - Проблемы математизации народного хозяйства Молдавской ССР. - Кишинев, 1990, с. 126 - 130.

[695] *Солтан В.П.* Введение в аксиоматическую теорию выпуклости. - Кишинев: Штиинца, 1984.

[696] *Стронгин Р.Г.* Численные методы в многоэкстремальных задачах. - М.: Наука, 1978.

[697] *Сухарев А.Г., Тимохов А.В., Федоров В.В.* Курс методов оптимизации. - М.: Наука, 1986.

[698] *Тарасов С.П., Хачиян Л.Г., Эрлих И.И.* Метод вписанных эллипсоидов. - Докл. АН СССР, 1988, т. 298, № 3, с. 1081 - 1085.

[699] *Таха Х.* Введение в исследование операций. Пер. с англ. в двух томах.- М: Мир, 1985.

[700] *Тетерев А.Г.* Об одном обобщении задач линейного и дробно-линейного программирования. - Экономика и математические методы, 1969, т. 5, № 3, с. 440 - 447.

[701] *Федоров В. В.* Численные методы максимина. - М.: Наука, 1979.

[702] *Филиппович Е.И.* Решение задач дробно-линейного программирования. - Труды по вопросам применения электронных вычислительных машин в народном хозяйстве. - Изд. Горьковского исследователького физико-математического института, Горкий, 1964.

[703] *Форд Х., Фалкерсон Д.* Потоки в сетях. Пер. с англ. - М: Мир, 1966.

[704] *Хабибулин Р.Ф.* Об одном методе для нахождения точки выпуклого множества. - Исследования по прикладной математике. - Казань: КГУ, 1977, вып. 4, с. 15 - 22.

[705] *Хачиян Л.Г.* Полиномиальные алгоритмы в линейном программировании. - Журнал вычислительной математики и математической физики, 1980, т. 20, № 1, с. 52 - 68.

[706] *Хачиян Л.Г.* Полиномиальный алгоритм в линейном программировании. - ДАН СССР, 1979, т. 244, № 5, с. 1093 - 1096.

[707] *Цурков В.И.* Декомпозиция в задачах большой размерности. - Наука, 1981.

[708] *Черноусько Ф.Л.* Оценивание фазового состояния динамических систем. Метод эллипсоидов. - М.: Наука, 1988.

[709] *Черчмен У., Акоф Р., Арноф Л.* Введение в исследование операций. Пер. с англ. - М: Наука, 1968.

[710] *Шварцман А.П.* Об одном алгоритме дробно-линейного программирования. - Экономика и математические методы, 1965, т. 1, № 4, с. 558 - 566.

[711] *Шор Н.З.* Использование операций растяжения пространства в задачах минимизации выпуклых функций. - Кибернетика, 1970, № 1, с. 6 - 12.

[712] *Шор Н.З.* Квадратные оптимизационные задачи. - Известия АН СССР. Техническая кибернетика, 1987, № 1, с. 128 - 139.

[713] *Шор Н.З.* Метод отсечения с растяжением пространства для решения задач выпуклого программирования. - Кибернетика, 1977, № 1, с. 94 - 95.

[714] *Шор Н.З.* Методы минимизации негладких функций и матричные задачи оптимизации: сборник избранных трудов. - Кишинэу, Эврика, 2009.

[715] *Шор Н.З.* Методы минимизации недифференцируемых функций и их приложения. - Киев: Наукова думка, 1979.

[716] *Шор Н.З.* Методы недифференцируемой оптимизации и сложные экстремальные задачи: сборник избранных трудов. - Кишинэу, Эврика, 2008.

[717] *Шор Н.З.* Монотонные модификации r-алгоритма и их приложения. - Кибернетика и системный анализ, 2002, № 6, с. 74 - 95.

[718] *Шор Н.З.* Новые направления в развитии методов негладкой оптимизации. - Кибернетика, 1977, № 6, с. 87 - 91.

[719] *Шор Н.З.* О классе почти-дифференцируемых функций и одном методе минимизации функций этого класса. - Кибернетика, 1972, № 4, с. 65 - 70.

[720] *Шор Н.З.* О скорости сходимости обобщенного градиентного спуска с растяжением пространства. - Кибернетика, 1970, № 2, с. 80 - 85.

[721] *Шор Н.З.* О скорости сходимости обобщенного градиентного спуска. - Кибернетика, 1968, № 3, с. 98 - 99.

[722] *Шор Н.З.* Об одном классе оценок глобального минимума полиномиальных функций. - Кибернетика, 1987, № 6, с. 9 - 11.

[723] *Шор Н.З.* Об одном подходе к получению глобальных экстремумов в полиномиальных задачах математического программирования. - Кибернетика, 1987, № 5, с. 102 - 106.

[724] *Шор Н.З.* Обобщенные градиентные методы минимизации негладких функций и их приложения к задачам математического программирования. Обзор. - Экономика и математические методы, 1976, т. 12, № 2, с. 337 - 356.

[725] *Шор Н.З.* Применение метода градиентного спуска для решения сетевой транспортной задачи. - Материалы научного семинара по теоретическим и прикладным вопросам кибернетики. - Экономическая кибернетика и исследования операций, вып. 1. - Киев, 1962, с. 1 - 17.

[726] *Шор Н.З.* Применение обобщенного градиента спуска в блочном программировании. - Кибернетика, 1967, № 3, с. 53 - 55.

[727] *Шор Н.З.* Роль избыточных ограничений в улучшении двойственных оценок для полиномиальных оптимизационных задач. - Кибернетика и системый анализ, 1998, №. 4, с. 106 - 121.

[728] *Шор Н.З., Бардадым Т.А., Журбенко Н.Г., Лиховид А.П., Стецюк П.И.* Использование методов негладкой оптимизации в задачах стохастического программирования. - Кибернетика и системный анализ, 1999, № 5, с. 33 - 47.

[729] *Шор Н.З., Гамбурд П.Р.* Некоторые вопросы сходимости обобщенного градиентного спуска. - Кибернетика, 1971, № 6, с. 82 - 84.

[730] *Шор Н.З., Гершович В.И.* Об одной модификации алгоритмов минимизации градиентного типа с растяжением пространства для решения задач большой размерности. - Кибернетика, 1981, № 5, с. 67 - 70.

[731] *Шор Н.З., Журбенко Н.Г.* Метод минимизации, использующий операцию растяжения пространства в направлении разности двух последовательных градиентов. - Кибернетика, 1971, № 3, с. 51 - 59.

[732] *Шор Н.З., Журбенко Н.Г., Лиховид А.П., Стецюк П.И.* Развитие алгоритмов недифференцируемой оптимизации и их приложения. - Кибернетика и системый анализ, 2003, № 4, с. 80 - 94.

[733] *Шор Н.З., Соломон Д.И.* Декомпозиционные методы в дробно-линейном программировании. - Кишинев: Штиинца, 1989.

[734] *Шор Н.З., Стеценко С.И.* Квадратичные экстремалъные задачи и недифференцируемая оптимизация. - Киев: Наукова думка, 1989.

[735] *Шор Н.З., Стецюк П.И.* Использование модификации r-алгоритма для нахождения глобального минимума полиномиальных функций. - Кибернетика и системный анализ, 1997, № 4, с. 28 - 49.

[736] *Шор Н.З., Шабашова Л.П.* О решении минимаксных задач методом обобщенного градиентного спуска с растяжением пространства. - Кибернетика, 1972, № 1, с. 82 - 88.

[737] *Шор Н.З., Щепакин М.Б.* Алгоритм решения двухэтапной задачи стохастического программирования. - Теория оптимальных решений. К.: Ин-т кибернетики АН УССР, 1967, № 2, с. 38 - 44.

[738] *Шор Н.З., Щепакин М.Б.* Алгоритм решения двухэтапной задачи стохастического программирования. - Кибернетика, 1968, № 3, с. 56 - 58.

[739] *Эддоус М., Стэнсфилд Р.* Методы принятия решение. Пер. с англ. - М: Аудит Юнити, 1997.

[740] *Юдин Д.Б., Гольштейн Е.Г.* Линейное программирование: Теория, методы, приложения. - М: Физматгиз, 1963.

[741] *Юдин Д.Б., Гольштейн Е.Г.* Линейное программирование: Теория, методы, приложения. - М: Наука, 1969.

[742] *Юдин Д.Б., Немировский А.С.* Информационная сложность и эффективные методы решения выпуклых экстремальных задач. - Экономика и математические методы, 1976, вып. 12, № 2, с. 357 - 365.

[743] *Юдин Д.Б.* Математические методы управления в условиях неполной информации. - М: Советское радио, 1974.

СОДЕРЖАНИЕ

НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК УКРАИНЫ

ИНСТИТУТ КИБЕРНЕТИКИ ИМ. В.М. ГЛУШКОВА

**МИНИСТЕРСТВО ПРОСВЯЩЕНИЯ
РЕСПУБЛИКИ МОЛДОВА**

**АКАДЕМИЯ ТРАНСПОРТА,
ИНФОРМАТИКИ И КОММУНИКАЦИЙ**

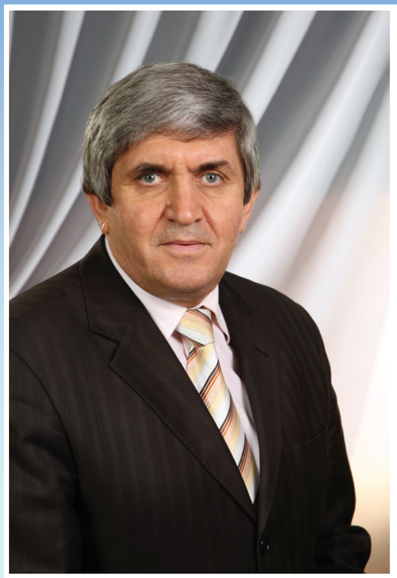
Д. И. СОЛОМОН

ДРОБНОЕ ПРОГРАММИРОВАНИЕ И НЕДИФФЕРЕНЦИРУЕМАЯ ОПТИМИЗАЦИЯ

Издательство “Эврика”, тел. 63-90-29
мун. Кишинэу, Республика Молдова
Мунчештское шоссе, 121а

Подписано в печать 22.03.2010
Формат 60x90/16
Усл. печ. листов 34,75
Тираж 150
Заказ № 10

Типография Академии наук Молдовы
г. Кишинев, ул. П. Мовилэ, 8



СОЛОМОН ДМИТРИЙ ИЛЬИЧ

Ректор Академии транспорта, информатики и коммуникаций Республики Молдова, доктор хабилитат технических наук, Академик Международной академии информатизации при ООН. Лауреат Государственной премии (1998 г.) и премии молодежи Молдавии (1984 г.) в области науки и техники.

Родился в 1951 году в селе Цариград, Дрокиевского района. Окончил факультет математики и кибернетики Кишиневского государственного университета (1974 г.) по специальности прикладная математика. Работал в Институте математики и информатики Академии наук Молдовы. (1974 - 1992 гг.; 1994 - 2000 гг.), Академии экономических знаний Молдовы (1993 -1994 гг.), Академии транспорта, информатики и коммуникаций (с 2000 г.)

В Институте кибернетики им. В.М. Глушкова НАН Украины защитил кандидатскую (1985 г.) и докторскую (1992 г.) диссертации в области дробно-линейной оптимизации, методов декомпозиции, математического моделирования и информационных технологий на автомобильном транспорте.

Является автором 12 монографий и более 100 научных работ.

Д. И. Соломон

ДРОБНОЕ ПРОГРАММИРОВАНИЕ И НЕДИФФЕРЕНЦИРУЕМАЯ ОПТИМИЗАЦИЯ

1