

Национальная академия наук Украины
Институт кибернетики имени В.М. Глушкова

На правах рукописи

Лаптин Юрий Петрович

УДК 519.8

**МЕТОДЫ НЕГЛАДКОЙ ОПТИМИЗАЦИИ РЕШЕНИЯ
СТРУКТУРИРОВАННЫХ ЗАДАЧ**

01.05.01 – теоретические основы информатики и кибернетики

Диссертация
на соискание научной степени доктора
физико-математических наук

Научный консультант – академик НАН Украины Сергиенко И.В.

Киев – 2015

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
РАЗДЕЛ 1. ОБЗОР ЛИТЕРАТУРЫ.....	13
РАЗДЕЛ 2. ВЫПУКЛЫЕ ПРОДОЛЖЕНИЯ ФУНКЦИЙ В ЗАДАЧАХ ОПТИМИЗАЦИИ.....	26
2.1. Ограниченные области определения функций	26
2.1.1. Задачи с блочной системой ограничений-равенств	28
2.1.2. Нелинейные задачи сетевой структуры.....	30
2.2. Конические продолжения функций	34
2.3. Коническая аппроксимация функций	56
2.4. Выводы.....	61
РАЗДЕЛ 3. ИСПОЛЬЗОВАНИЕ ТОЧНЫХ ШТРАФНЫХ ФУНКЦИЙ...	63
3.1. Штрафные коэффициенты и производные по направлению	63
3.2. Точные штрафные функции и невыпуклые задачи оптимизации ..	74
3.2.1. Постановка задачи и некоторые свойства штрафных функций.....	75
3.2.2. Дополнительные свойства и доверительные области.....	77
3.2.3. Общая схема решения задачи	84
3.3. Обобщенный подход к определению штрафных коэффициентов..	86
3.4. Выводы.....	94
РАЗДЕЛ 4. СХЕМЫ ДЕКОМПОЗИЦИИ ПО ПЕРЕМЕННЫМ В СТРУКТУРИРОВАННЫХ ЗАДАЧАХ ВЫПУКЛОЙ ОПТИМИЗАЦИИ	96
4.1. Декомпозиция по переменным и ε -субградиенты	96
4.2. Регуляризация задач в схемах декомпозиции.....	105
4.3. Точные штрафные функции в схемах декомпозиции по переменным	109
4.4. Ограниченные области определения функций	111

4.5. Особенности программной реализации для специального класса задач	113
4.6. Вложенные блочные задачи линейного программирования	115
4.6.1. Описание задач и схема декомпозиции	117
4.6.2. Стохастическая модель планирования инвестиций в электроэнергетике	127
4.6.3. Задача перспективного планирования в электроэнергетике с учетом суточных неравномерностей потребления электроэнергии	131
4.7. Выводы	147
РАЗДЕЛ 5. ИСПОЛЬЗОВАНИЕ НЕГЛАДКОЙ ОПТИМИЗАЦИИ В ЗАДАЧАХ ПОСТРОЕНИЯ АЛГОРИТМОВ КЛАССИФИКАЦИИ	150
5.1. Задачи построения линейных классификаторов	150
5.1.1. Формулировка задачи и линейная разделимость	151
5.1.2. Классификаторы для линейно разделимых множеств	157
5.1.3. Минимизация полосы неправильной классификации	160
5.1.4. Минимизация эмпирического риска для линейно неразделимых множеств	162
5.1.5. Сравнение с другими подходами	169
5.1.6. Результаты вычислительных экспериментов	172
5.2. Вложенные линейные классификаторы	183
5.3. Последовательные линейные классификаторы	186
5.4. Задачи выделения информативных признаков	187
5.5. Выводы	192
РАЗДЕЛ 6. ЗАДАЧИ ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ СЛОЖНЫХ ТЕХНИЧЕСКИХ ОБЪЕКТОВ	194
6. 1. Общая характеристика задач	196
6.2. Модели оптимального проектирования на номинальный режим	202
6.2.1. Структурное описание задач оптимального проектирования	204

6.2.2. Задача оптимального проектирования поверхностей нагрева конвективной шахты.....	205
6.2.3. Задача оптимального проектирования многоступенчатого пароперегревателя.....	208
6.2.4. Задача оптимального проектирования энергетического котлоагрегата в целом.....	209
6.3. Модель учета альтернативных режимов эксплуатации.....	215
6.4. Описание прототипа системы оптимального проектирования котлоагрегатов.....	217
6.4.1. Состав разрабатываемого программного обеспечения.....	218
6.4.2. База данных.....	220
6.4.3. Администратор базы данных.....	223
6.4.4. Менеджер вариантов.....	223
6.4.5. Графический редактор.....	225
6.4.6. Средства фактографического отображения данных.....	227
6.5. Описание оптимизационной подсистемы.....	227
6.5.1. Рекурсивная библиотека классов моделирования сложных объектов.....	228
6.5.2. Описание методик теплотехнических расчетов.....	231
6.5.3. Методы решения задач оптимального проектирования.....	235
6.5.4. Интерфейс пользователя.....	236
6.6. Результаты вычислительных экспериментов.....	240
6.7. Выводы.....	242
ЗАКЛЮЧЕНИЕ.....	245
Приложение А. Расчет конвективных и ширмовых поверхностей нагрева.....	252
Приложение Б. Описание уравнений материального и теплового баланса.....	257
ЛИТЕРАТУРА.....	267

ВВЕДЕНИЕ

Работа посвящена формированию новых подходов к решению сложных оптимизационных задач. Полученные результаты позволяют расширить области применения методов негладкой оптимизации, повысить эффективность существующих алгоритмов.

Актуальность темы определяется множеством оптимизационных задач, возникающих в самых разных сферах. Это задачи планирования, управления, проектирования сложных технических объектов, технической и медицинской диагностики, а также задачи в других областях. Для решения таких задач разрабатываются и широко используются в настоящее время разнообразные методы оптимизации и соответствующее программное обеспечение. Реальные задачи, возникающие на практике, часто оказываются задачами очень большой размерности, обладают специальной структурой и особенностями, которые должны учитываться при их решении. Это требует разработки новых методов, специальных подходов при использовании существующих алгоритмов.

Эффективными средствами решения прикладных задач являются методы негладкой оптимизации, r -алгоритм Н.З. Шора. Изначально эти методы разрабатывались для задач безусловной оптимизации. Для решения такими методами задач с ограничениями обычно используются точные штрафные функции. Одна из проблем, которая возникает при таком подходе, это определение значений штрафных коэффициентов, что требует решения вспомогательных специальных задач. Решение таких вспомогательных задач может быть достаточно трудоемким и не всегда возможным. В частности, это невозможно при использовании схем декомпозиции по переменным для решения структурированных задач большой размерности. На практике проблема выбора значений штрафных коэффициентов часто возлагается на пользователя. Это обычно приводит к использованию завышенных значений

коэффициентов или к повторному многократному решению исходной задачи при увеличивающихся значениях коэффициентов.

Еще одна проблема возникает в случае, когда функции, описывающие исходную задачу, определены на ограниченных множествах. Применение методов безусловной оптимизации для таких задач требует продолжения исходных функций на все пространство переменных исходной задачи.

Практическими задачами, которые требуют разработки специальных подходов к определению значений штрафных коэффициентов и продолжению функций, определенных на ограниченных множествах, являются задачи оптимального проектирования сложных технических объектов и задачи перспективного планирования в электроэнергетике. Другой сферой применения методов негладкой оптимизации являются задачи построения классификаторов при распознавании образов, в задачах диагностики (в медицине, технике, других областях). Особые проблемы возникают в этой области в случае, когда обучающие выборки оказываются линейно неразделимыми.

Связь работы с научными темами Института кибернетики им. В.М. Глушкова НАН Украины. Основные научные и прикладные результаты были получены в ходе выполнения следующих научно-исследовательских тем:

1. ВФК.120.07 „Розробка та теоретичне обґрунтування математичних моделей та ефективних алгоритмів оптимізації комплексних систем енергетики”, сроки выполнения: I кв. 2002 – IV кв. 2006 гг., № госрегистрации 0102U003210;
2. ВФ.120.08 „Розробка та обґрунтування нових ефективних чисельних методів розв'язування складних задач оптимізації”, сроки выполнения: I кв. 2004 – IV кв. 2007 гг., № госрегистрации 0104U000276;
3. ВК.120.10.07 „Розробка програмних засобів оптимального проектування при реконструкції та модернізації енергетичних

- котлоагрегатів ТЕС”, строки виконання: II кв. 2007 – IV кв. 2009 рр., № держреєстрації 0107U004962;
4. ВФ.120.09 “Розробити математичні та програмні засоби розв’язування деяких класів структурованих задач оптимізації”, № держреєстрації 0108U000280 (2008–2012 рр.);
 5. Совместный проект НАН Украины и РФФИ ВК.235.10.08 "Методи автоматичного інтелектуального аналізу даних в задачах розпізнавання об’єктів зі складними зв’язками", № держреєстрації 0108U002564 (2008–2009 рр.);
 6. ВК.120.12.08 „Розробка прототипу програмних засобів оптимального проектування при реконструкції та модернізації енергетичних котлоагрегатів ТЕС надкритичного тиску”, строки виконання: II кв. 2008 – IV кв. 2008 рр., № держреєстрації 0108U002200;
 7. Совместный проект НАН Украины и РФФИ ВК.235.11 «Оптимізаційні підходи в задачах машинного навчання та аналізу даних», № держреєстрації 0110U004033 (2010–2011 рр.);
 8. ВК.120.12.14 ”Розробка програмних засобів оптимального проектування енергетичних котлоагрегатів при реконструкції та модернізації ТЕС з урахуванням альтернативних режимів експлуатації”, строки виконання: I кв. 2014 – IV кв. 2014 рр., № держреєстрації 0114U001862;
 9. ВП.120.13 «Розробити послідовні та паралельні алгоритмічні та програмні засоби для розв’язування прикладних екстремальних задач методами негладкої оптимізації», строки виконання: I кв. 2013– IV кв. 2015 рр., № держреєстрації 0113U003146;
 10. Совместный проект НАН Украины и РФФИ ВК.120.16 “Розробка математичних методів розпізнавання та аналізу структурованих об’єктів з взаємно залежними компонентами», № держреєстрації 0114U004032 (2014–2015 рр.);

11.ВФ.120.14 «Розробити субградієнтні алгоритми розв'язання задач оптимізації з гарантованою точністю», строки выполнения: I кв. 2014 – IV кв. 2016 гг., № госрегистрации 0114U001055.

В темах из приведенного списка диссертант выполнял следующие работы:

в темах 1, 2, 4, 5, 7, 9, 10, 11 диссертант был ответственным исполнителем отдельных этапов работ;

в темах 3, 6, 8 диссертант был ответственным исполнителем темы в целом.

Цель и задача исследования.

Объект исследований – методы решения задач оптимизации с ограничениями.

Предмет исследований – методы построения задач безусловной оптимизации, эквивалентных исходным задачам с ограничениями; методы определения значений коэффициентов точных штрафных функций; негладкие методы решения сложных прикладных задач оптимизации, схемы декомпозиции блочных задач оптимизации со связывающими переменными.

Методы исследований – методы математического, негладкого и выпуклого анализа, математическое моделирование, проведение вычислительных экспериментов.

Научная новизна полученных результатов:

- о разработаны новые эффективные методы выпуклого продолжения целевой функций с допустимой области исходной задачи оптимизации на все пространство переменных; при этом для сформированной задачи безусловной оптимизации обеспечивается эквивалентность исходной задаче с ограничениями;

- разработаны новые методы определения значений коэффициентов точных штрафных функций, не требующие решения вспомогательных задач оптимизации;
- предложены новые подходы, позволяющие использовать разработанные методы выпуклого продолжения целевой функций и методы определения значений коэффициентов точных штрафных функций в схемах декомпозиции по переменным для сложных структурированных задач оптимизации большой размерности;
- разработана новая модель минимизации эмпирического риска для задачи построения линейного классификатора в случае линейно неразделимой обучающей выборки; показано, что оптимизационная задача, решаемая в известном методе опорных векторов, является частным случаем лагранжевой релаксации разработанной математической модели;
- показано, что специальная задача построения линейного классификатора в случае линейно неразделимой обучающей выборки полиномиально разрешима при определенных условиях.

Практическое значение полученных результатов:

- программная реализация метода выпуклого продолжения целевой функции с использованием r -алгоритма Н.З. Шора показала высокую устойчивость по отношению к плохой обусловленности задач специального вида;
- использование разработанных методов определения значений коэффициентов точных штрафных функций позволяет строить эффективные алгоритмы решения задач оптимизации с ограничениями;
- использование разработанных методов определения значений коэффициентов точных штрафных функций в схемах декомпозиции по переменным позволяет преодолеть существенные проблемы,

возникающие в случае, когда генерируемые подзадачи не имеют допустимых решений при некоторых значениях связывающих переменных;

- разработанные программные средства построения линейных классификаторов позволяют выделять информативные признаки в задачах медицинской диагностики;
- разработанные программные средства позволяют решать задачи оптимального проектирования сложных технических объектов на современных теплоэлектростанциях, соответствующее программное обеспечение передано в опытную эксплуатацию в Харьковское ЦКБ ЭНЕРГОПРОГРЕС.

Личный вклад соискателя

Все научные результаты получены соискателем самостоятельно. В работах, опубликованных в соавторстве, диссертанту принадлежат следующие результаты:

[100], [102], [120] – автору диссертации принадлежат общие системные решения, разработка математических моделей и оптимизационного программного обеспечения выполнена совместно с Журбенко Н.Г., соответствие математических моделей нормам теплотехнических расчетов обеспечивались Левиным М.М. и Волковицкой П.И.;

[106] – совместная программная реализация схемы декомпозиции (с использованием предложенной диссертантом процедуры вычисления ε -субградиентов) и проведение вычислительных экспериментов:

[108] – диссертанту принадлежат результаты, которые позволяют вычислять ε -субградиенты оптимальных значений подзадач на основе приближенных решений этих подзадач, схема регуляризации исходной задачи, которая использует процедуру линейного поиска допустимой точки, предложенную Журбенко Н.Г.:

[110], [117] – диссертантом предложена общая схема декомпозиции оптимизационных задач с вложенной структурой, программная реализация и

вычислительные эксперименты выполнены совместно с Лиховидом А.П., Стрюковой Н.Н.; [113], [115], [116], [121], [122], [126], [127], [128] – авторами совместно исследовались известные математические модели задач построения линейных классификаторов, диссертантом предложены новые модели и алгоритмы их решения; программная реализация выполнялась совместно с Лиховидом А.П., использовались программы оптимизации, разработанные Журбенко Н.Г.;

[111] – диссертантом предложена схема решения указанных задач, программная реализация выполнялась совместно с Крошко Д.П.;

[118] – диссертантом предложена схема решения указанных задач, программная реализация и вычислительные эксперименты выполнялись совместно с Лиховидом А.П.;

[123] – диссертантом предложено обобщение выпуклых продолжений функций с допустимой области на все пространство переменных, Бардадым Т.А. провела исследование операции инфимальной свёртки целевой функции с функцией Минковского [123, с. 57–60, раздел 1];

[132] – диссертантом предложена схема выделения информативных признаков в задаче медицинской диагностики, реализовано соответствующее программное обеспечение.

Апробация результатов диссертации

Результаты диссертационной работы прошли апробацию на 12 международных конференциях [187–198] и докладывались на научных семинарах в Институте кибернетики имени В.М. Глушкова НАН Украины.

Публикации

Основные научные результаты диссертации в полной мере изложены в 32 научных статьях:

- 24 работы [100, 101, 102, 104, 105, 106, 107, 108, 109, 110, 111, 112, 117, 118, 119, 120, 121, 122, 123, 124, 125, 129, 130] опубликованы в научных изданиях из Перечня научных специализированных изданий Украины, из

которых 6 работ [105, 108, 112, 123, 120, 121] – в изданиях Украины, включенных в международные наукометрические базы и переводятся на английский язык издательствами "Springer" та "Begell House , Inc.";

- 8 работ [113, 115, 116, 122, 126, 127, 131, 128] опубликованы в научных периодических изданиях иностранных государств;

- работа [132] опубликована в украинском международном научном журнале не по специальности диссертации, но имеет прикладное значение.

При выполнении работ для автора очень ценными были поддержка и советы со стороны академика НАН Украины Сергиенко И.В., член-корреспондента НАН Украины Кнопова П.С., профессора Данилина Ю.М. Важными были сотрудничество с сотрудниками отдела методов негладкой оптимизации Института кибернетики им. В.М. Глушкова НАН Украины, помощь и полезные замечания ст. н. с. Бардадым Т.А. Автор всем выражает глубокую благодарность.

РАЗДЕЛ 1. ОБЗОР ЛИТЕРАТУРЫ

Теория и методы решения задач оптимизации является сформировавшейся областью прикладной математики. В настоящее время разработано множество самых разных методов решения оптимизационных задач и соответствующее программное обеспечение. Тем не менее, реальные прикладные задачи часто требуют разработки специальных подходов для успешного решения. Это может быть связано с большой размерностью, необходимостью учета специальной структуры, математическими особенностями таких задач. Особую роль в этой области имеют методы негладкой оптимизации, одним из основателей которых является Н.З. Шор. Общему описанию методов негладкой оптимизации посвящено много публикаций и монографий, в частности, работы Н.З. Шора [176, 177, 178, 179, 180, 182, 183, 184], Kiwiel К.С. [42, 43], Поляка Б.Т. [150], Демьянова В. Ф. и Васильева Л. В. [76], Нурминского Е.А. [144], Стецюка П.И. [168], Пшеничного Б.Н., Ненахова Э.И., Кузьменко В.Н. [156], Журбенко Н.Г. [89, 90]. Применению методов негладкой оптимизации в задачах дробного программирования посвящены работы Соломона Д.И. [165, 181], в непрерывных задачах оптимального разбиения множеств – работы Киселевой Е.М. [94, 95].

Важную роль в теории и практике методов оптимизации играют точные штрафные функции. Их использование позволяет для оптимизационных задач с ограничениями формировать эквивалентные задачи безусловной оптимизации. При этом эквивалентность достигается при конечных значениях штрафных коэффициентов. Впервые точные штрафные функции были предложены в работах Zangwill W. [61] и Еремина И.И. [84]. Вопросам исследования и использования точных штрафных функций и их обобщений посвящено много работ (см., например, [77, 82, 83, 184, 155, 75, 70]).

Точные штрафные функции недифференцируемы, поэтому для решения сформированных задач безусловной оптимизации естественно применять методы негладкой оптимизации. Тем не менее, точные штрафные функции широко используются в гладких методах оптимизации. Это методы линеаризации [75, 155], методы последовательного квадратичного программирования (см., например, [16,17]) и другие.

Свойства точных штрафных функций исследовались в работах Шора Н.З. [184], Бертсекаса Д. [70], Пшеничного Б.Н. и Данилина Ю.М. [75, 155]. Вопросам построения самых разных форм точных штрафных функций посвящены работы Евтушенко Ю.Г. и Жадана В.Г. [82, 83]. При формулировке условий, которым должны удовлетворять штрафные коэффициенты, обычно используются оптимальные значения множителей Лагранжа исходной задачи.

В выпуклом случае решения сформированной задачи безусловной оптимизации совпадают с решениями исходной задачи. Для невыпуклых задач ситуация сложнее – точки локальных минимумов исходной задачи являются точками локальных минимумов штрафной функции, однако штрафная функция может иметь дополнительные локальные минимумы, которые не принадлежат допустимой области исходной задачи. Эти проблемы подробно исследовались в работах Демьянова В.Ф. [77, 78, 79].

Вопросы применения точных штрафных функций для решения некоторых классов прикладных задач рассматривались в работах [94, 95, 141, 165, 167, 180, 181]. Для оценки значений штрафных коэффициентов обычно применяются решения вспомогательных упрощенных задач. Это не всегда удобно и может быть достаточно трудоемко. Использование завышенных штрафных коэффициентов приводит к плохообусловленным овражным задачам. Поскольку методы негладкой оптимизации, r -алгоритм Н.З.Шора достаточно эффективны при решении плохообусловленных задач, вопросы определения значений штрафных коэффициентов часто возлагаются на пользователя. В отдельных случаях оценки штрафных коэффициентов могут

быть определены при анализе конкретной прикладной задачи. В работах Березовского О.А. и Стецюка П.И. [69] для такой оценки использовались подходы, предложенные Пшеничным Б.Н. [155].

Важной сферой приложений негладкой оптимизации являются схемы декомпозиции задач математического программирования большой размерности, обладающих специальной структурой. Такие схемы сводят решение исходной задачи к решению серии задач меньшего объема (блоков). К этим схемам относятся методы декомпозиции по ограничениям (Dantzig G.B. и Wolfe P. [18]), по переменным (Benders J.F. [6]) и методы декомпозиции по ресурсам (Geoffrion A.M. [19]). Подобные подходы оказываются особенно эффективными, если для задач, соответствующих отдельным блокам, существуют быстрые специализированные алгоритмы решения. Впоследствии этому направлению было посвящено много работ. Общие описания схем декомпозиции можно найти в работах Шора Н.З. [180, 184], Цуркова В. И. [173], Мину М. [140], описания приложений к различным прикладным задачам – в работах Михалевича В.С., Трубина В.А., Шора Н.З., Соломона Д.И. [141, 181, 165]. Широкие приложения схемы декомпозиции имеют в задачах стохастического программирования (см. работы Ермольева Ю.М. [85], Shapiro A., Dentcheva D., Ruszczyński A. [59]).

Остановимся более подробно на методах декомпозиции по переменным. Тут необходимо упомянуть L-Shaped метод [63] для задач линейного программирования (модификацию метода Бендерса), в котором используется метод секущих плоскостей для решения координирующей задачи; обобщенный алгоритм декомпозиции Бендерса [30, 33] для гладких задач нелинейного программирования; регуляризованный метод декомпозиции [58], в котором для решения координирующей задачи применяется bundle-метод, а в целевую функцию координирующей задачи включается регуляризирующая квадратичная штрафная добавка. В настоящее время предложены различные методы декомпозиции, использующие приближенные решения подзадач (см., например, [60, 28, 54, 55, 43]).

Для методов декомпозиции по переменным характерны определенные проблемы, связанные с отсутствием решений подзадач при некоторых значениях связывающих переменных. В работе [35] приводится анализ таких проблем, возникающих при использовании обобщенных алгоритмов декомпозиции Бендерса. Рассматривается пример нелинейной выпуклой задачи, для которой схема декомпозиции Бендерса не обеспечивает сходимости к стационарной точке. Обсуждаются возможные модификации метода Бендерса. Еще один подход [28] к преодолению указанных проблем связан с продолжением функций, определяющих зависимость оптимальных значений подзадач от значений связывающих переменных, с ограниченных множеств на все пространство связывающих переменных. При этом применяются специальным образом точные штрафные функции. Предложена методика оценки штрафных коэффициентов, которая может использоваться при решении практических задач специального вида.

Необходимо отметить, что в нелинейном случае даже выяснение вопроса, существует ли решение подзадачи при заданных значениях связывающих переменных, может оказаться очень непростым. Все эти проблемы легко решаются при использовании точных штрафных функций. Однако традиционные методики определения значений штрафных коэффициентов, основанные на решении вспомогательных задач, в схемах декомпозиции не применимы, поскольку информации, получаемой при решении отдельных подзадач, для этого недостаточно.

Новым подходам к определению значений штрафных коэффициентов, не требующих решения вспомогательных трудоемких задач, посвящен раздел 3 диссертационной работы. В разделе 4 эти подходы детализируются применительно к схемам декомпозиции по переменным. Также в этом разделе рассматриваются вопросы вычисления ε -субградиентов

оптимальных значений подзадач и вопросы реализации вложенных схем декомпозиции. В разделе 2 описываются новые приемы построения задач безусловной оптимизации, основанные на выпуклых продолжениях функций исходной задачи с ограничениями. Подходы к построению выпуклых продолжений приведены в работах Ириарт-Уррути Ж.-Б., Moreau J. [37, 52, 93], однако они оказываются слишком трудоемкими.

Раздел 5 посвящен применению методов негладкой оптимизации для решения задач построения классификаторов. Такого рода задачи в распознавании образов исследовались многими авторами, в том числе, – Вапником В. Н. [64, 71]; Журавлёвым Ю. И. [86, 87]; Петуниным Ю.И., Ключиным Д. А. и соавторами [97, 147, 148, 149, 158]; Шлезингером М.И. [174]; Гупалом А.М., Сергиенко И.В. [74]; Киселевой Е.М. [95]. В задачах распознавания рассматриваются объекты, которые однозначно определяются совокупностью (вектором) признаков. Множество объектов, объединенных по некоторому принципу, называется классом. В рассматриваемых задачах классы представляются конечными (обучающими) выборками своих объектов. Классификатором или решающим правилом называется правило отнесения объекта (образа) к одному из классов на основании его вектора признаков.

В случае двух классов для построения классификатора может использоваться, например, линейная функция в пространстве признаков. Обучением классификатора в этом случае является подбор коэффициентов линейной функции, обеспечивающих наилучшее в некотором смысле разделение обучающих выборок. При этом определение значений коэффициентов линейных или нелинейных функции сводится к задачам математического программирования с ограничениями.

Для характеристики сложности задачи построения линейных классификаторов важным является понятие линейной делимости обучающей выборки, т.е. существование линейного классификатора, правильно разделяющего точки обучающей выборки.

Для линейно разделимой выборки задача построения оптимального линейного классификатора обычно сводится к решению задачи квадратичного программирования, для линейно неразделимой выборки чаще всего используется метод опорных векторов (SVM). Математическая модель, используемая в методе SVM, может трактоваться как некоторая релаксация задачи минимизации эмпирического риска или как регуляризация по А.Н. Тихонову некорректно поставленной задачи (см., например, [137, 174]). Еще один подход к решению проблемы линейной неразделимости заключается в переходе от исходного пространства X признаков объектов к новому пространству H с помощью некоторого преобразования $\psi: X \rightarrow H$. Если пространство H имеет достаточно большую размерность, то можно надеяться, что в нем выборка окажется линейно разделимой. Пространство H называется спрямляющим. Такие подходы широко используются (см., например, [20]).

Задачи построения классификаторов в случае многих классов рассматривались в работах разных авторов, в частности в работах Bennett К.Р., Mangasarian О.Л. [10, 13]. Приводятся обобщения метода RLP (робастное линейное программирование), предложенного в [10], и метода SVM, используемых для построения бинарных классификаторов, на случай k классов, $k > 2$.

Пусть каждому классу поставлена в соответствие линейная функция $f_i(x) = x^T w^i - \gamma^i$, $x, w^i \in R^n$, $\gamma^i \in R$, $i = 1, \dots, k$. Рассматриваются линейные классификаторы следующего вида

$$i(w, \gamma, x) = \arg \max \{x^T w^i - \gamma^i, i = 1, \dots, k\}, \quad (1.1)$$

где $w = (w^1, \dots, w^k)$, $\gamma = (\gamma^1, \dots, \gamma^k)$. В [13] для отыскания значений $w^i \in R^n$, $\gamma^i \in R$, $i = 1, \dots, k$, предлагается использовать **M-RLP** метод, состоящий в решении задачи линейного программирования специального вида для заданной обучающей выборки. Если оптимальное значение этой задачи равно 0, то обучающая выборка кусочно-линейно разделима

(неразделима в противном случае). Необходимо отметить, что целевая функция сформулированной задачи не включает члены, которые бы позволили оптимизировать классификатор $i(w, \gamma, x)$ в случае кусочно-линейной разделимости обучающей выборки. Поэтому для такого случая предлагается использовать **M-SVM** метод, в котором в целевую функцию включаются члены, обеспечивающие максимизацию «зазора» классификации.

В работе [13] также рассматриваются обобщения приведенных подходов для случая, когда точки исходного пространства признаков R^n отображаются с помощью специальной функции в пространство большей размерности. Такой прием позволяет строить нелинейные классификаторы для k классов, $k > 2$.

Одним из существенных вопросов является определение необходимых и достаточных условий кусочно-линейной разделимости обучающей выборки (множеств Ω^i , $i = 1, \dots, k$). Для случая двух классов ($k = 2$) известно, что множества Ω^1, Ω^2 линейно разделимы тогда и только тогда, когда выпуклые оболочки этих множеств не пересекаются. Для случая $k = 3$ в работе Петунина Ю.И., Шульдешова Г.А. [147] приведен пример, в котором выпуклые оболочки множеств $\Omega^1, \Omega^2, \Omega^3$ не пересекаются, но эти множества не являются кусочно-линейно разделимыми. Условия, при которых обучающая выборка является линейно разделимой в случае многих классов, сформулированы в работах [147, 158].

Другие подходы к построению классификаторов в случае многих классов рассматривались в работах [12, 22, 38, 44, 45, 56]. В [12] предложен метод, получивший название «*один против всех*». Для совокупности множеств Ω^i , $i = 1, \dots, k$ составляется k бинарных SVM моделей. В i -й SVM

модели полагается $\Omega_1 = \Omega^i$, $\Omega_{-1} = \bigcup_{j=1, j \neq i}^k \Omega^j$. Решая задачи построения

классификатора для каждой SVM модели, получаем совокупность

дискриминантных функций $f^i(x) = \langle w^i, x \rangle - w_0^i$, $i = 1, \dots, k$. В качестве классификатора используется функция

$$i(x) = \arg \max \left\{ \langle w^i, x \rangle - w_0^i, i = 1, \dots, k \right\}, \quad (1.2)$$

В другом методе [44, 45], названном «каждый против каждого», для каждой пары множеств Ω^i, Ω^j , $i, j = 1, \dots, k, i \neq j$, формируется SVM модель и определяется дискриминантная функция $f^{ij}(x) = \langle w^{ij}, x \rangle - w_0^{ij}$. При классификации произвольной точки $x \in R^n$ используется стратегия «голосования». По результатам анализа значений функций $f^{ij}(x) = \langle w^{ij}, x \rangle - w_0^{ij}$ определяется число «голосов» за каждый класс $i = 1, \dots, k$. То есть просматриваются все $k(k-1)/2$ функции $f^{ij}(x) = \langle w^{ij}, x \rangle - w_0^{ij}$, при этом если по значению $\text{sign}(\langle w^{ij}, x \rangle - w_0^{ij})$ точка x относится к классу i , то число голосов за этот класс увеличивается на единицу, в противном случае на единицу увеличивается число голосов за класс j . После этого точка x относится к классу, набравшему максимальное число голосов.

Третий подход [56] (directed acyclic graph SVM – DAGSVM) использует такой же набор $k(k-1)/2$ SVM моделей и определяет дискриминантную функцию $f^{ij}(x) = \langle w^{ij}, x \rangle - w_0^{ij}$ для каждой пары множеств Ω^i, Ω^j , $i, j = 1, \dots, k, i \neq j$. Однако на этапе тестирования точки x используется ориентированный бинарный ациклический граф с выделенным корнем, который имеет $k(k-1)/2$ внутренних вершин и k висячих вершин. Каждой внутренней вершине соответствует бинарный классификатор двух классов i и j , каждой висячей вершине соответствует конкретный класс. Тестирование точки x начинается в корневой вершине. В каждой вершине оценивается значение функции $f^{ij}(x) = \langle w^{ij}, x \rangle - w_0^{ij}$ и, в зависимости от полученного

значения, выполняется переход к одному из двух потомков. При достижении висячей вершины определяется класс, к которому относится точка x .

В работе [56] приведены результаты вычислительных экспериментов, которые показали предпочтительность метода «каждый против каждого» и DAGSVM–метода по сравнению с другими методами классификации многих классов.

В [20] содержится описание библиотеки современных программных средств, реализующих методы опорных векторов для самых разных задач, возникающих на практике.

В работах Демьянова В.Ф. и других авторов [23, 67] обсуждаются вопросы применения негладкого дискриминантного анализа к задачам диагностики. Формулируется задача построения такого классификатора, чтобы количество неправильно идентифицированных точек было минимальным. В качестве целевой функции используются так называемые суррогатные функционалы [23]. Для решения задач построения классификаторов используются методы негладкой оптимизации. В качестве решения принимается локальный минимум сформулированной задачи. Рассмотренный подход применялся при решении практических задач диагностики в медицине [67].

В работах Astorino A., Gaudioso M. [1, 2], Bagirov A.M [3, 4, 5] вводятся понятия полиэдральных и **max-min** классификаторов для разделения двух множеств. Такие классификаторы описываются наборами гиперплоскостей в пространстве признаков. Формулируются определения и условия разрешимости множеств с помощью таких классификаторов. Задачи построения полиэдральных и **max-min** классификаторов являются задачами невыпуклой негладкой оптимизации. Для решения разрабатываются специальные алгоритмы. Близкие задачи рассматриваются в работах Стрекаловского А. С., Груздевой Т. В. [73, 170].

Раздел 6 посвящен задачам оптимального проектирования сложных технических объектов в электроэнергетике. Этими задачами много занимались в Институте систем энергетики им. Л.А. Мелентьева (ИСЭМ) СО РАН (бывший Сибирский энергетический институт СО АН СССР).

Для построения математических моделей сложных теплосиловых систем в ИСЭМ СО РАН разработана оригинальная методика и создан реализующий ее программно-вычислительный комплекс, называемый системой машинного построения программ (СМПП) [96, 139, 152]. Система позволяет по информации об элементном составе технологической схемы и связях между элементами автоматически генерировать программу расчета сложной теплоэнергетической установки. Для решения задач оптимизации теплосиловых систем в ИСЭМ СО РАН разработан и реализован оригинальный метод «с памятью» [139]. С использованием СМПП выполнены многочисленные оптимизационные исследования теплосиловых систем различных типов [135,136,139,145].

Проблемы оптимизация тепловых циклов и процессов ТЭС, оптимизация поверхностей нагрева и теплоэнергетических установок в целом рассматривались в работах [68,92,151,164,175]. Энергетические котлоагрегаты состоят из большого количества взаимосвязанных компонент. Большая часть компонент таких объектов работает при предельных температурах и нагрузках. При эксплуатации этих объектов должны выполняться различные технические, технологические и экологические ограничения. При их проектировании стремятся минимизировать стоимостные, технико-экономические (приведенные затраты) или экологические показатели. Разработка математических моделей сложных технических объектов представляет самостоятельную проблему, которая должна решаться совместно специалистами в области моделирования (оптимизации) и высококвалифицированными специалистами в предметной области. Разрабатываемые модели должны быть относительно простыми и адекватно отражать моделируемые объекты.

Одной из особенностей рассматриваемых задач является дискретность некоторых конструктивных характеристик проектируемых объектов. Число петель, заходность поверхностей нагрева должны быть целочисленными величинами, диаметры труб и толщины стенок должны выбираться из заданного перечня (сортамент труб). Учет этих особенностей приводит к формированию сложных дискретных моделей оптимального проектирования. Для решения этих задач должны использоваться соответствующие методы дискретного программирования, направленного, локального поиска. Общие подходы к построению таких методов рассматривались в работах Сергиенко И. В., Шило В. П., Гуляницкого Л. Ф. [159, 160, 161, 162].

Для решения прикладных задач оптимизации в настоящее время используется множество различных программных средств. В целом, эти средства можно разбить на две группы [26]: системы-решатели оптимизационных задач и системы моделирования прикладных задач.

К первой группе относятся SNOPT [34], KNITRO [15], IPOPT [39] и другие. Эти программные системы решают оптимизационные задачи, включающие как линейные, так и нелинейные ограничения. Целевая функция также может состоять из линейной и нелинейной частей. Нелинейные зависимости реализуются подпрограммами, которые разрабатывает пользователь, и вызываются из оптимизационных систем. Хотя многие нелинейные задачи большой размерности успешно решаются, в целом гарантировать решение конкретной задачи невозможно.

Описание структуры нелинейной задачи математического программирования и разработка подпрограмм, реализующих требуемые нелинейные зависимости, представляют самостоятельную проблему и являются предметом второй группы программных средств, к которым относятся GAMS – General Algebraic Modeling System [14], AMPL [31] и ряд других. Эти системы ориентированы на описание задач в формульном виде с

помощью специального языка. Для решения используются программные средства первой группы.

Приведенные программные средства позволяют эффективно решать нелинейные задачи математического программирования во многих областях, но для оптимизационного моделирования сложных технических объектов требуются дополнительные возможности.

Выводы. Для решения прикладных задач в настоящее время широко используются методы негладкой оптимизации и точные штрафные функции. Свойства точных штрафных функций теоретически всесторонне исследованы. Однако не известны методы определения значений штрафных коэффициентов, которые не требовали бы решения вспомогательных оптимизационных задач.

В схемах декомпозиции по переменным точные штрафные функции позволяют преодолеть существенные проблемы, связанные с отсутствием решений подзадач при некоторых значениях связывающих переменных. Однако решение вспомогательных оптимизационных задач для определения значений штрафных коэффициентов оказывается неприемлемым в связи с чрезмерно высокой трудоемкостью такого подхода.

Серьезные проблемы возникают для оптимизационных задач, в которых функции, описывающие эти задачи, определены на ограниченных множествах. Точные штрафные функции для таких задач использоваться не могут. Необходимо применять продолжения функций на все пространство переменных. Эффективные методы продолжения функций с сохранением свойств выпуклости в настоящее время неизвестны.

Методы негладкой оптимизации используются в настоящее время для решения задач построения классификаторов в распознавании образов. Актуальными для этой области являются вопросы улучшения качества получаемых решений и расширения классов решаемых задач.

Задачи оптимального проектирования сложных технических объектов описываются совокупностями взаимосвязанных компонент. Каждая

компонента характеризуется множествами входных и выходных параметров. Расчеты значений выходных параметров при заданных входах выполняются в соответствии с принятыми методиками и могут быть выполнены не при любых значениях входов. Выходы одних компонент являются входами других. Для решения таких задач оптимального проектирования должны применяться методы редукции исходной задачи к задачам меньшей размерности с использованием специальной структуры исходной задачи. Также для такого применения необходимо выполнять продолжения функций с ограниченных множеств на все пространство переменных.

РАЗДЕЛ 2. ВЫПУКЛЫЕ ПРОДОЛЖЕНИЯ ФУНКЦИЙ В ЗАДАЧАХ ОПТИМИЗАЦИИ

При решении практических задач оптимизации часто возникают существенные проблемы, если функции, описывающие исходную задачу, определены не на всем пространстве переменных. Это, в частности, характерно для задач оптимального проектирования сложных технических объектов, которые рассматриваются в разделе 6. Возможным подходом к преодолению таких проблем является продолжение функций на все пространство переменных. Задачи при этом становятся задачами негладкой оптимизации и требуют использования соответствующих методов решения. Примеры задач, для которых целесообразно применение продолжений функций приведены в подразделе 2.1.

Естественным требованием к продолжению функций является сохранение выпуклости, если исходная задача была выпуклой. Эффективные процедуры построения выпуклых продолжений рассматриваются в подразделах 2.2 и 2.3.

2.1. Ограниченные области определения функций

Рассмотрим задачу: найти

$$f^* = \min \{f(x) : x \in C\}, \quad (2.1)$$

где $C = \{x \in R^n : h(x) \leq 0\}$, $f, h : R^n \rightarrow R \cup \{+\infty\}$ – выпуклые функции, C – компактное множество, $C \subseteq \text{dom } f \cap \text{dom } h$.

Поскольку f, h могут быть не определены вне C , естественным является продолжение функции f с множества C на все пространство R^n . Ниже приведены эвристические варианты такого продолжения. Вопросам продолжения выпуклых функций с сохранением выпуклости посвящены следующие подразделы.

Пусть $p_C(x)$ – проекция точки $x \in R^n$ на множество C . В качестве продолжения f на все пространство рассмотрим функцию $f_1(x) = f(p_C(x)) + s\|x - p_C(x)\|$, где $s \in R$, и рассмотрим задачу: найти

$$f_1^* = \min \{ f(p_C(x)) + s\|x - p_C(x)\| : x \in R^n \}. \quad (2.2)$$

Нетрудно видеть, что при $s > 0$ задачи (2.1) и (2.2) эквивалентны.

Пусть задана точка $x^0 \in \text{int} C$. Для $x \in R^n$, $x \notin C$ рассмотрим процедуру поиска точки $\pi_C(x)$ пересечения отрезка $[x^0, x]$ (луча, исходящего из x^0) с границей множества C . Такого рода процедуры были предложены Н.Г. Журбенко вместо трудоемкой операции проектирования на допустимое множество. Близкий подход использовался в [108] для регуляризации задач в схемах декомпозиции по переменным.

Для продолжения f на все пространство положим

$$f_2(x) = \begin{cases} f(\pi_C(x)) + s\|x - \pi_C(x)\|, & \text{если } x \notin C, \\ f(x), & \text{если } x \in C. \end{cases} \quad (2.3)$$

Рассмотрим задачу: найти:

$$f_2^* = \min \{ f_2(x) : x \in R^n \}. \quad (2.4)$$

Как и ранее, нетрудно убедиться, что при $s > 0$ задачи (2.1) и (2.4) эквивалентны.

Рассмотренные функции $f_1(x)$, $f_2(x)$ – непрерывные, но негладкие, даже если функции f, h исходной задачи являются гладкими. Для решения задач (2.2), (2.4) должны использоваться методы негладкой оптимизации, например, r -алгоритм Н.З. Шора, однако, в целом такой подход следует считать эвристическим, поскольку функции $f_1(x)$, $f_2(x)$ могут быть невыпуклыми, и сказать что-либо о сходимости и скорости сходимости алгоритма затруднительно.

2.1.1. Задачи с блочной системой ограничений-равенств

Оптимизационные задачи, возникающие в ходе моделирования сложных технических объектов, обычно имеют специальную структуру, которую необходимо учитывать при разработке программных средств их решения. Одной из особенностей таких задач является наличие структурированных нелинейных ограничений-равенств. Такие ограничения часто распадаются на небольшие, слабо связанные, зависящие от параметров подсистемы нелинейных уравнений, которые в отдельных случаях удается эффективно решать на каждой итерации оптимизационных алгоритмов [103, 102]. В результате значительно понижается размерность оптимизационной задачи. В общем случае возникают проблемы, обусловленные тем, что нелинейные функции обычно определены на ограниченных областях, а решения подсистем уравнений должны принадлежать достаточно узким областям.

В данном пункте вводится понятие псевдорешения системы уравнений блока, что позволяет свести исходную оптимизационную задачу к редуцированной оптимизационной задаче меньшей размерности. Для ее решения могут использоваться методы негладких штрафных функций и методы негладкой оптимизации [180, 184].

Рассмотрим задачу математического программирования следующего вида: найти

$$\min f_0(x, y^1, \dots, y^Q) \quad (2.5)$$

при ограничениях

$$f_k(x, y^1, \dots, y^Q) \leq 0, \quad k = 1, \dots, K, \quad (2.6)$$

$$g^q(x, y^q) = 0, \quad q = 1, \dots, Q, \quad (2.7)$$

$$A_x^q x + A_y^q y^q \leq b^q, \quad q = 1, \dots, Q, \quad (2.8)$$

где $x \in R^L$, $y^q \in R^{N_q}$, функции $g^q(x, y^q): R^L \times R^{N_q} \rightarrow R^{N_q}$,
 $f_k(x, y^1, y^2, \dots, y^Q): R^L \times R^{N_1} \times \dots \times R^{N_Q} \rightarrow R$ непрерывны и непрерывно
дифференцируемы; $b^q \in R^{m_q}$, матрицы A_x^q , A_y^q имеют размерности $m_q \times L$ и
 $m_q \times N_q$.

Предполагается, что функции f_k определены при любых значениях
аргументов, функции g^q определены при любых значениях (x, y^q) ,
удовлетворяющих ограничениям (2.8).

Систему уравнений (2.7) и ограничений (2.8) для каждого q будем
называть блоком B^q . Будем предполагать, что если при некотором x
подсистема (2.7) для блока B^q имеет решение, удовлетворяющее
ограничениям (2.8), то такое решение единственно.

Обозначим S^q множество точек x , для которых подсистема (2.7), (2.8)
для блока B^q имеет решение, $y^q(x)$ – решение подсистемы (2.7) в точке
 $x \in S^q$.

Очевидно, что следующая задача эквивалентна исходной (2.5)–(2.8):
найти

$$f_0^* = \min f_0(x, y^1(x), \dots, y^Q(x)) \quad (2.9)$$

при ограничениях

$$f_k(x, y^1(x), \dots, y^Q(x)) \leq 0, k = 1, \dots, K, \quad (2.10)$$

$$x \in \bigcap_{q=1}^Q S^q. \quad (2.11)$$

Непосредственное решение задачи (2.9)–(2.11) затруднено, поскольку
множества S^q , $q = 1, \dots, Q$ заданы неявно.

Пусть $p_{S^q}(x)$ – проекция точки $x \in R^L$ на множество S^q , $d^q(x)$ –
расстояние от точки x до множества S^q , $q = 1, \dots, Q$,

$$d^q(x) = \|x - p_{S^q}(x)\| \quad (2.12)$$

Псевдорешением блока B^q (подсистемы (2.7), (2.8)) в произвольной точке x назовем отображение $\tilde{y}^q(x) = y^q(p_{S^q}(x))$. Положим $\varphi_k(x) = f_k(x, \tilde{y}^1(x), \dots, \tilde{y}^Q(x))$, $k = 0, \dots, K$ и рассмотрим задачу: найти

$$\varphi_0^* = \min_x \left\{ \varphi_0(x) : \varphi_k(x) \leq 0, d^q(x) \leq 0, k = 1, \dots, K, q = 1, \dots, Q \right\}. \quad (2.13)$$

Очевидно, что задачи (2.9)–(2.11) и (2.13) эквивалентны.

Пусть зафиксировано некоторое $q \in \{1, \dots, Q\}$. Для вычисления функции $d^q(x)$ введем вспомогательные переменные $z^q \in R^L$ и рассмотрим вспомогательную задачу: найти

$$d^q(x) = \min_{z, y} \left\{ \|x - z^q\| : g^q(z^q, y^q) = 0, A_x^q z^q + A_y^q y^q \leq b^q \right\}. \quad (2.14)$$

Очевидно, что если при некотором \bar{x} , решение подсистемы (2.7), (2.8) для блока B^q существует, то задача (2.14) имеет решения при любых x , функция $d^q(x)$ определена для всех x . Если при заданном x система (2.7), (2.8) разрешима, то $d^q(x) = 0$, иначе $d^q(x) > 0$.

Пусть (z^{*q}, y^{*q}) решение задачи (2.14) при заданном x . Нетрудно видеть, что, $p_{S^q}(x) = z^{*q}$, $\tilde{y}^q(x) = y^{*q}$. Тем самым в результате решения задач (2.14) вычисляются значения как функций $d^q(x)$, так и функций $\varphi_k(x)$.

Для решения задач (2.14) может использоваться метод линеаризации Б.Н. Пшеничного [155], обеспечивающий на каждой итерации выполнение линейных ограничений.

2.1.2. Нелинейные задачи сетевой структуры

В подразделе рассматриваются задачи, часто возникающие при проектировании сложных технических объектов [66], в частности, энергетических котлоагрегатов [172, 103]. К таким относятся задачи, которые могут быть описаны совокупностью взаимосвязанных блоков. Каждый блок

характеризуется наборами входов и выходов. При заданных входах однозначно определяются выходы блока. Некоторые выходы одних блоков могут быть входами для других блоков. Особенностью рассматриваемых задач является ограниченность областей определения функций, определяющих выходы блоков при заданных входах, что затрудняет применение существующих методов. Предлагается подход, основанный на продолжении рассматриваемых функций на все пространство. Результирующая задача является негладкой, и для ее решения могут использоваться методы негладкой оптимизации.

Будем считать заданной совокупность $V = \{1, \dots, Q\}$ непрерывно дифференцируемых вектор-функций $f^q(x^q)$, $x^q \in R^{n^q}$, $q \in V$. Обозначим I^q – множество индексов переменных, $|I^q| = n^q$, J^q – множество индексов функций для вектор-функции f^q . Предполагается, что область определения вектор-функции f^q задается множеством S^q простой структуры. Например,

$$S^q = \{x^q : A^q x^q \leq b^q\}, \quad (2.15)$$

где $b^q \in E^{m^q}$, A^q – матрица размерности $m^q \times n^q$.

Простота структуры множеств S^q подразумевает, что существуют эффективные алгоритмы решения задачи проектирования произвольной точки на множество S^q .

Вектор-функцию f^q вместе с областью определения S^q назовем блоком B^q , переменные $x_i^q, i \in I^q$ будем называть входами, функции $f_j^q, j \in J^q$ – выходами блока B^q .

Совокупность блоков является связанной, т.е. выходы некоторых блоков могут быть входами для других блоков. Связи для каждой пары блоков B^p и B^q будем описывать множествами

$M(p, q) = \{(i, j) : i \in I^q, j \in J^p\}$. Множество $M(p, q)$ определяет выходы блока B^p , которые являются входами блока B^q :

$$x_i^q = f_j^p(x^p), (i, j) \in M(p, q). \quad (2.16)$$

Будем говорить, что блоки p, q связаны дугой (p, q) , если $M(p, q) \neq \emptyset$, множество всех дуг обозначим E .

Предполагается, что в совокупности функций $f_i^q, j \in J^q, q \in V$ выделено подмножество $T \subseteq \{(j, q) : j \in J^q, q \in V\}$, которое определяет ограничения, и задана функция $F^0(x^1, \dots, x^Q)$, которая является целевой функцией оптимизационной задачи и определена при любых значениях переменных.

Рассматриваемую задачу сформулируем следующим образом: найти

$$\min F^0(x^1, \dots, x^Q) \quad (2.17)$$

при ограничениях

$$f_j^q(x^q) \leq 0, (j, q) \in T, \quad (2.18)$$

$$x_i^q = f_j^p(x^p), (i, j) \in M(p, q), (p, q) \in E, \quad (2.19)$$

$$x^q \in S^q, q \in V. \quad (2.20)$$

Совокупность блоков $B^q, q \in V$ и связей между ними E будем называть сетью функциональных блоков (V, E) , выходами сети блоков является объединение выходов всех блоков, входами сети блоков – переменные x_i^q , не являющиеся выходами каких-либо блоков. В дальнейшем будем рассматривать ациклические сети блоков.

В задачах проектирования сложных технических объектов [66] блоками являются отдельные компоненты таких объектов, функциональные зависимости $f^q(x^q)$ описываются инженерными методиками расчетов. Ограничениями вида (2.20) описывается область применения этих методик.

Очевидно, что соотношения (2.19) при условии ацикличности сети позволяют выразить любой выход сети (V, E) в виде функции от входов сети. При этом размерность задачи (2.17)–(2.20) существенно уменьшается, выходы сети являются сложной суперпозицией функций, описывающих отдельные блоки.

Такое представление задачи (2.17)–(2.20) с помощью суперпозиций функций, описывающих отдельные блоки, будем называть **прямой редукцией** исходной задачи.

Определенной проблемой при разработке алгоритмов решения редуцированной задачи является учет ограничений (2.20), которые задают области определения функций блоков. Пусть x – входы сети блоков, $x^q(x)$ – входы блока B^q , как функция от входов x . Очевидно, что если $x^q(x) \notin S^q$, то выходы блока B^q не определены, а также не определены входы и выходы любого блока B^p , для которого существует путь в сети (V, E) из B^q в B^p . Т.е. не определены функции, описывающие область определения (2.20) для блока B^p .

Для преодоления указанной проблемы рассмотрим подход, связанный с доопределением используемых функций на все пространство. Пусть задана сеть функциональных блоков (V, E) . Для блока B^q , $q \in V$ введем обозначения:

$p_{S^q}(x^q)$ – проекция точки $x^q \in R^{n^q}$ на множество S^q ,

$d_{S^q}(x^q) = \|x^q - p_{S^q}(x^q)\|$ – расстояние от точки x^q до множества S^q .

Расширением вектор-функции $f^q(x^q)$ для произвольной точки $x^q \in R^{n^q}$ назовем вектор-функцию $\bar{f}^q(x^q) = f^q(p_{S^q}(x^q))$. Рассмотрим задачу:
найти

$$\min F^0(x^1, \dots, x^Q) \quad (2.21)$$

при ограничениях

$$\bar{f}_j^q(x^q) \leq 0, \quad (j, q) \in T, \quad (2.22)$$

$$x_i^q = \bar{f}_j^p(x^p), \quad (i, j) \in M(p, q), \quad (p, q) \in E, \quad (2.23)$$

$$d_{S^q}(x^q) = 0, \quad q \in V. \quad (2.24)$$

Поскольку $\bar{f}^q(x^q) = f^q(x^q)$ при условии $d_{S^q}(x^q) = 0, q \in V$, то решения задач (2.17)–(2.20) и (2.21)–(2.24) совпадают.

Пусть сеть функциональных блоков (V, E) ациклическая, x – множество входов сети. Учитывая (2.23), представим выходы $\bar{f}^q(x^q)$ каждого блока как функцию от входов сети. Такое представление назовем расширенной редукцией сети блоков.

Обозначим $x^q(x)$ значения входов блока B^q для заданного входа x сети (V, E) , полученные при расширенной редукции сети блоков. Положим $\Phi^0(x) = F^0(x^1(x), \dots, x^Q(x))$, $\varphi^q(x) = \bar{f}^q(x^q(x))$, $\delta^q(x) = d_{S^q}(x^q(x))$, тогда задачу (2.21)–(2.24) можно представить в виде: найти

$$\min \Phi^0(x), \quad (2.25)$$

при ограничениях

$$\varphi_j^q(x) \leq 0, \quad (j, q) \in T, \quad (2.26)$$

$$\delta^q(x) \leq 0, \quad q \in V. \quad (2.27)$$

Функции $\Phi^0(x)$, $\varphi^q(x)$, $\delta^q(x)$ определены при любых x , и для решения задачи (2.25)–(2.27) может использоваться метод точных штрафных функций.

2.2. Конические продолжения функций

Основная трудность в построении продолжений выпуклой функции на все пространство заключается в обеспечении выпуклости такого продолжения. Некоторые известные подходы к построению выпуклых

продолжений приведены в в работах [37, 52, 93],, однако эти подходы достаточно трудоемкие.

Будем рассматривать исходную задачу (2.1), $f, h: R^n \rightarrow R \cup \{+\infty\}$ – выпуклые функции, C – компактное множество, $C \subseteq \text{dom } f \cap \text{dom } h$. Считаются заданными точка $x^0 \in \text{int } C$ и число $E, E < f(x^0)$, которое будем называть параметром продолжения целевой функции.

Положим

$$\begin{aligned} \chi_E(x) &= f(\pi_C(x)) + \delta_E(x) \|x - \pi_C(x)\|, \\ \psi_E(x) &= \begin{cases} \chi_E(x), & \text{если } x \notin C, \\ f(x), & \text{если } x \in C, \end{cases} \end{aligned} \quad (2.28)$$

где $\delta_E(x) = \frac{f(\pi_C(x)) - E}{\|\pi_C(x) - x^0\|}$, $\pi_C(x)$ – точка пересечения отрезка $[x^0, x]$ (луча,

исходящего из x^0) с границей множества C .

Функция $\psi_E(x)$ непрерывна и отличается от $f_2(x)$, которая определяется соотношением 2.3, множителем $\delta_E(x)$, зависящим от x . Пример функций $f(x)$ и $\chi_E(x)$ для случая одномерного пространства приведен на рис. 2.1.

Рассмотрим задачу

$$\psi_E^* = \inf \{ \psi_E(x) : x \in R^n \} . \quad (2.29)$$

Нетрудно видеть, что если $\delta_E(x) \geq 0$ для всех $x \notin C$, то $\psi_E^* = f^*$. Если найдется точка $x \notin C$ такая, что $\delta_E(x) < 0$, то $\psi_E^* = -\infty$.

Легко проверить, что функция $\chi_E(x)$ может быть представлена в виде

$$\chi_E(x) = E + (f(\pi_C(x)) - E) \frac{\|x - x^0\|}{\|\pi_C(x) - x^0\|}. \quad (2.30)$$

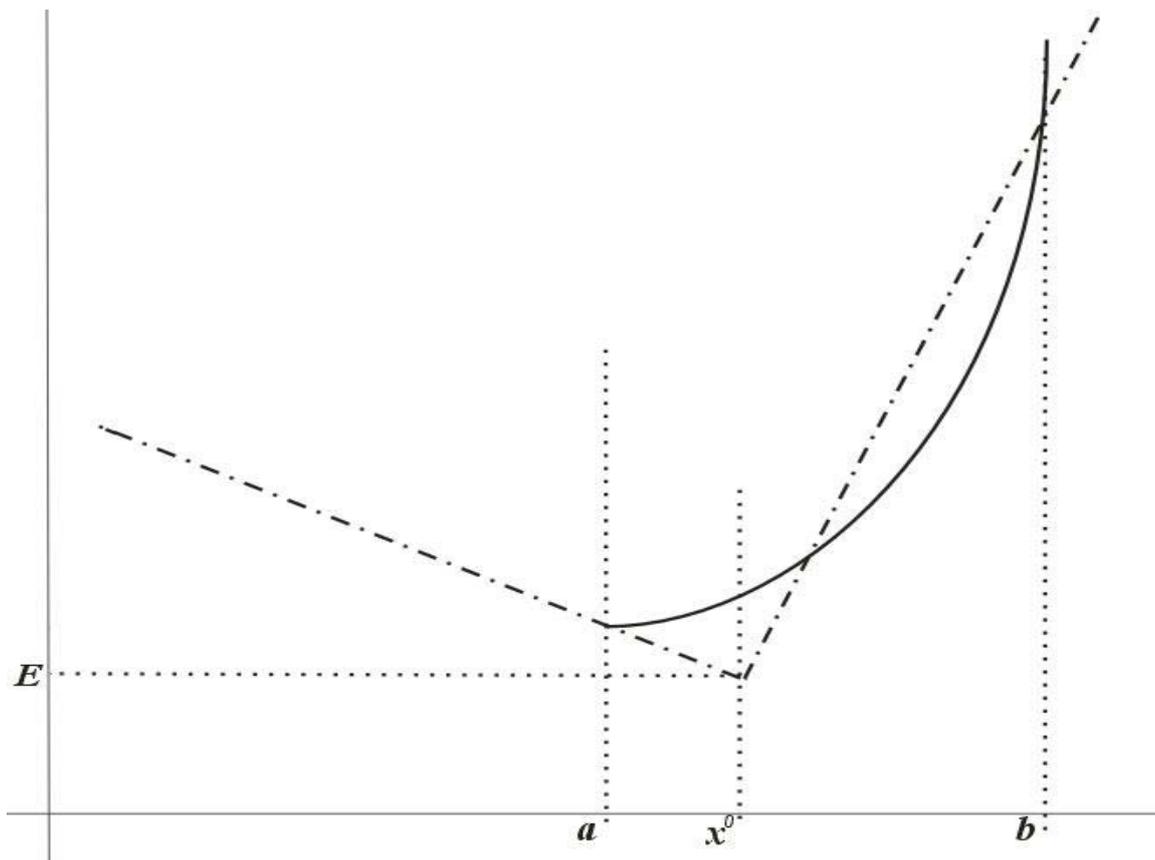


Рис. 2.1. Сплошной линией показана функция $f(x)$, штрих-пунктирной – функция $\chi^E(x)$, множество C – отрезок $[a, b]$

Заметим, что функция

$$R_C(x) = \frac{\|x - x^0\|}{\|\pi_C(x) - x^0\|} \quad (2.31)$$

есть функция Минковского (см., например, [154]) для множества C , записанная относительно точки x^0 . При этом $R_C(x) \leq 1$, если $x \in C$, $R_C(x) \geq 1$, при $x \notin C$, и $R_C(x) = 1$, если x лежит на границе множества C .

Лемма 2.1. Пусть $E \leq f^*$, тогда $\psi_E^* = f^*$ и решения задач (2.29) и (2.1) совпадают.

Доказательство очевидно, поскольку в этом случае $f(\pi_C(x)) - E \geq 0$ для всех $x \notin C$ и $\chi_E(x) \geq \chi_E(\pi_C(x)) = f(\pi_C(x))$.

Функцию $\psi_E(x)$ будем называть коническим продолжением функции $f(x)$.

Введем понятие конической аппроксимации $\gamma_E(x)$ для функции $f(x)$. Обозначим F надграфик функции f на множестве C

$$F = \{(\lambda, x) \in R \times C : \lambda \geq f(x)\}.$$

Положим $z = (\lambda, x)$, $z \in R \times R^n$. Рассмотрим коническую оболочку $K(E)$ надграфика F с вершиной в точке $z_E^0 = (E, x^0)$

$$K(E) = \{v : v \in R \times R^n, v = z_E^0 + \alpha(z - z_E^0), \alpha \geq 0, z \in F\}. \quad (2.32)$$

Множество $K(E)$ выпукло (поскольку выпуклым является множество F) и может рассматриваться как надграфик некоторой выпуклой функции. Эту функцию обозначим $\gamma_E(x)$ и будем называть конической аппроксимацией функции f на множестве C . Функция $\gamma_E(x)$ определена на всем пространстве R^n и принимает конечные значения при любых x . Пример конической аппроксимации функции приведен на рис. 2.2.

Обозначим $\gamma'_E(x^0, p)$ производную по направлению p функции $\gamma_E(x)$ в точке x^0 . Очевидно, что $\gamma_E(x^0) = E$, эта функция линейна по любому направлению p при смещении из точки x^0 , т.е. $\gamma_E(x^0 + pt) = E + \gamma'_E(x^0, p) \cdot t$, $t > 0$. Заметим также, что

$$\gamma_E(x) \leq f(x), \forall x \in C. \quad (2.33)$$

Положим frC – граница множества C , $f'(x, p)$ – производная функции f в точке x по направлению p , $p(x) = \frac{x - x^0}{\|x - x^0\|}$.

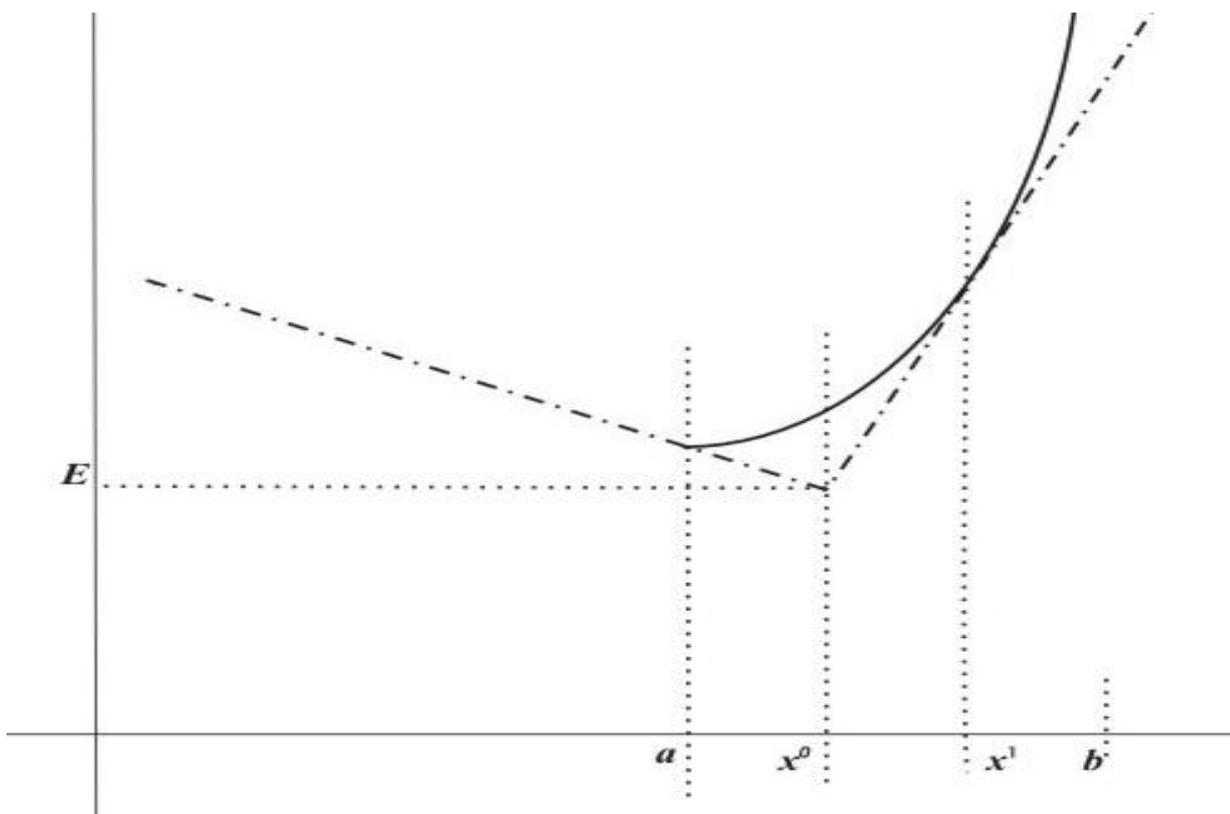


Рис. 2.2. Сплошной линией показана функция $f(x)$, штрих-пунктирной – функция $\gamma_E(x)$, в точках a, x^1 значения этих функций совпадают, множество C – отрезок $[a, b]$.

Лемма 2.2. Пусть величина E такая, что для $\forall x \in \text{fr}C$ выполняется $f(x) - f'(x, p(x)) \cdot \|x - x^0\| \geq E$. Тогда $\gamma_E(x) = \chi_E(x)$ для $\forall x \in R^n$ и $f'(x, p(x)) \leq \gamma'_E(x^0, p(x))$ для $\forall x \in \text{fr}C$.

Доказательство. Пусть $x \in \text{fr}C$. Предположим, что $f(x) > \gamma_E(x)$, тогда, очевидно, существует точка $x' \in [x^0, x]$ такая, что $\gamma_E(x') = f(x')$.

Откуда получаем

$$\begin{aligned} f(x) - f'(x, p(x)) \cdot \|x - x^0\| &\leq f(x) - \frac{f(x) - f(x')}{\|x - x'\|} \|x - x^0 - x' + x'\| = \\ &= f(x') - \frac{f(x) - f(x')}{\|x - x'\|} \|x' - x^0\| = \gamma_E(x') - \frac{f(x) - \gamma_E(x')}{\|x - x'\|} \|x' - x^0\| < \\ &< \gamma_E(x') - \frac{\gamma_E(x) - \gamma_E(x')}{\|x - x'\|} \|x' - x^0\| = \gamma_E(x') - \gamma'_E(x^0, p(x)) \|x' - x^0\| = E. \end{aligned}$$

Это противоречит условию леммы, т.е. $\gamma_E(x) = f(x)$ и $\gamma_E(x) = \chi_E(x)$. Нетрудно видеть, что если последнее равенство имеет место при $\forall x \in frC$, то $\gamma_E(x) = \chi_E(x)$ для $\forall x \in R^n$.

Пусть $x \in frC$. Из условий леммы следует

$$f'(x, p(x)) \cdot \|x - x^0\| \leq f(x) - E = \gamma_E(x) - E = \gamma'_E(x, p(x)) \cdot \|x - x^0\|.$$

То есть $f'(x, p(x)) \leq \gamma'_E(x^0, p(x))$. Лемма доказана. ■

Положим

$$E(x) = f(x) - f'(x, p(x)) \cdot \|x - x^0\|, \quad (2.34)$$

$$E^* = \inf \{E(x) : x \in frC\}. \quad (2.35)$$

Теорема 2.1. Пусть f – выпуклая липшицева на C функция, C – замкнутое ограниченное выпуклое множество, $C \subset \text{int dom } f$. Тогда E^* конечно и для всех $E \leq E^*$ функция $\psi_E(x)$ является выпуклой.

Доказательство. Величина E^* конечна, поскольку f – Липшицева функция, а C ограниченное множество.

Пусть $E \leq E^*$, тогда из леммы 2.2 следует $\gamma_E(x) = f(x)$ для $\forall x \in frC$.

Будем говорить, что линейная функция $l(x) = \langle g, x \rangle - \mu$ является опорной относительно множества C к функции $f(x)$, если $l(x) \leq f(x)$, $\forall x \in C$, существует точка $x' \in S$ такая, что $l(x') = f(x')$ и $g \in \partial f(x')$, где $\partial f(x')$ – субдифференциал функции $f(x)$ в точке x' . Точку x' будем называть опорной точкой функции $l(x)$.

Рассмотрим совокупность всех опорных относительно множества C функций $l(x) = \langle g, x \rangle - \mu$, обозначим H множество всех пар (g, μ) , порождающих такие функции.

Положим

$$\bar{f}(x) = \max \{ \langle g, x \rangle - \mu : (g, \mu) \in H \}, \quad x \in R^n.$$

Ясно, что функция $\bar{f}(x)$ – выпуклая, и $\bar{f}(x) = f(x), \forall x \in C$.

Можно показать, что при $E \leq E^*$ из леммы 2.2 следует $\gamma_E(x) \geq \bar{f}(x), \forall x \notin C$, откуда учитывая, что $\gamma_E(x) \leq f(x), \forall x \in C$, получаем $\psi_E(x) = \max \{ \bar{f}(x), \gamma_E(x) \}$. Т.е. функция $\psi_E(x)$ – выпуклая. Теорема доказана.

■

Обозначим $g_f(x), g_h(x)$ субградиенты функций f и h в точке x .

Теорема 2.2. Пусть $E \leq E^*$, $h(x^0) < 0$, $\bar{x} = \pi_C(x)$, $x \neq x^0$. Тогда $(g_h(\bar{x}), x^0 - \bar{x}) \neq 0$ и вектор

$$g = g_f(\bar{x}) + \frac{E - f(\bar{x}) - (g_f(\bar{x}), x^0 - \bar{x})}{(g_h(\bar{x}), x^0 - \bar{x})} g_h(\bar{x}) \quad (2.36)$$

есть субградиент функции $\chi_E(x)$ в точке x .

Доказательство. Предположим, что $(g_h(\bar{x}), x^0 - \bar{x}) = 0$, тогда $h(x^0) \geq h(\bar{x}) + (g_h(\bar{x}), x^0 - \bar{x}) = 0$, что противоречит условию $h(x^0) < 0$. Рассмотрим линейные функции $f_L(y) = f(\bar{x}) + (g_f(\bar{x}), y - \bar{x})$, $h_L(y) = h(\bar{x}) + (g_h(\bar{x}), y - \bar{x}) = (g_h(\bar{x}), y - \bar{x})$. Положим $C_L = \{y \in R^n : h_L(y) \leq 0\}$, рассмотрим $F_L = \{(\lambda, y) \in R \times C_L : \lambda \geq f_L(y)\}$ – надграфик функции f_L на множестве C_L и выпуклую коническую оболочку K_L надграфика F_L относительно точки $z^E = (E, x^0)$,

$$K_L = \{v = z^E + \alpha(z - z^E) \mid \alpha \geq 0, z \in F_L\}. \quad (2.37)$$

Множество $K_L(E)$ является надграфиком некоторой выпуклой функции, которую обозначим $\gamma_E^L(y)$. По построению $\gamma_E^L(\bar{x}) = f(\bar{x})$, в области $\{y \notin C_L\}$ функция $\gamma_E^L(y)$ – линейна, т.е. $\gamma_E^L(y) = f(\bar{x}) + (g, y - \bar{x})$, где вектор

g однозначно определяется по векторам $g_f(\bar{x})$, $g_h(\bar{x})$. Более того $\gamma_E^L(x^0 + \alpha(\bar{x} - x^0)) = \gamma_E(x^0 + \alpha(\bar{x} - x^0))$, $\alpha \geq 0$. Поскольку надграфик функции $\gamma_E(y)$ (или функции $\chi_E(x)$) принадлежит надграфику функции $\gamma_E^L(y)$, то функция $\gamma_E^L(y) = f(\bar{x}) + (g, y - \bar{x})$ является опорной для $\gamma_E(y)$, а вектор g – субградиент функции $\gamma_E(y)$ во всех точках $y = x^0 + \alpha(\bar{x} - x^0)$, $\alpha \geq 0$.

Нетрудно проверить, что для вектора g должны выполняться соотношения

$$f(\bar{x}) + (g, x^0 - \bar{x}) = E, \quad (2.38)$$

$$(g, y - \bar{x}) = (g_f(\bar{x}), y - \bar{x}) \quad (2.39)$$

для всех y таких, что

$$(g_h(\bar{x}), y - \bar{x}) = 0. \quad (2.40)$$

Представим векторы g , g_f в виде

$$g = g^\perp + g^\parallel, \text{ где } (g_h(\bar{x}), g^\perp) = 0, \quad g^\parallel = \lambda g_h(\bar{x}),$$

$$g_f = g_f^\perp + g_f^\parallel, \text{ где } (g_h(\bar{x}), g_f^\perp) = 0, \quad g_f^\parallel = \gamma g_h(\bar{x}), \quad \gamma = \frac{(g_f(\bar{x}), g_h(\bar{x}))}{\|g_h(\bar{x})\|^2}.$$

Нетрудно видеть, что из (2.39), (2.40) следует $g^\perp = g_f^\perp$, т.е.

$$g^\perp = g_f(\bar{x}) - \frac{(g_f(\bar{x}), g_h(\bar{x}))}{\|g_h(\bar{x})\|^2} g_h(\bar{x}).$$

(2.38), получаем $\lambda = \frac{(g_f(\bar{x}), g_h(\bar{x}))}{\|g_h(\bar{x})\|^2} + \frac{E - f(\bar{x}) - (g_f(\bar{x}), x^0 - \bar{x})}{(g_h(\bar{x}), x^0 - \bar{x})}$. Откуда

следует (2.36). Теорема доказана. ■

Замечание. Пусть функция h дифференцируема, тогда субградиент функций $\chi_E(x)$, $\psi_E(x)$ инвариантен относительно умножения функции $h(x)$

на произвольную дифференцируемую функцию $r: R^n \rightarrow R$ такую, что $r(x) > 0, x \in R^n$.

Если величина E удовлетворяет условиям леммы 2.1 и теоремы 2.1, то для решения задачи (2.29) может применяться любой алгоритм минимизации выпуклых функций. Рассмотрим случай, когда значения f^* и E^* неизвестны.

Пусть задан некоторый сходящийся алгоритм A безусловной минимизации выпуклых функций.

Теорема 2.3. Пусть алгоритм A применяется для решения задачи (2.29), для некоторых чисел $\delta > 0, E$ выполняются условия: если $x^k \notin C$, то

$$E < f(\bar{x}^k) - \delta, \quad (2.41)$$

$$E < f(\bar{x}^k) - f'(\bar{x}^k, p(\bar{x}^k)) \cdot \|\bar{x}^k - x^0\|, \quad (2.42)$$

где $\bar{x}^k = \pi_C(x^k)$, x^k – текущая точка на итерации $k, k = 0, 1, \dots$. Тогда последовательность точек, генерируемых алгоритмом A , сходится к решению задачи (2.1).

Доказательство.

Пусть при решении задачи (2.29) алгоритмом A была сгенерирована последовательность точек $x^k, k = 0, 1, \dots$. Обозначим g_Ψ^k субградиент функции Ψ_E в точке x^k , используемый алгоритмом $A, k = 0, 1, \dots$.

Рассмотрим функцию $\Psi(x) = \max_k \{ \Psi_E(x^k) + (g_\Psi^k, x - x^k) : k = 0, 1, \dots \}$. Эта функция выпукла и является нижней аппроксимацией функции $f(x)$ при $x \in C$.

Заметим, что $\Psi(x^k) = \Psi_E(x^k)$.

Можно показать, что при использовании алгоритма A для решения задачи

$$\Psi^* = \inf \{ \Psi(x) : x \in R^n \} \quad (2.43)$$

будет генерироваться та же последовательность точек x^k , $k = 0, 1, \dots$, что и при решении задачи (2.29).

Поскольку алгоритм A сходящийся, то для последовательности x^k , $k = 0, 1, \dots$, выполняется $\Psi(x^k) \rightarrow \Psi^*$, $k \rightarrow \infty$. В силу (2.41), (2.42) имеем $E < \Psi^*$.

Очевидно, что последовательность x^k , $k = 0, 1, \dots$, может либо иметь предельные точки, тогда каждая предельная точка является решением задачи (2.43), либо не иметь предельных точек, тогда точная нижняя грань в (2.43) не достигается.

Рассмотрим первый случай – последовательность x^k , $k = 0, 1, \dots$, имеет предельные точки. Обозначим $x^* = \arg \min \{f(x) : x \in C\}$, $\tilde{x} = \arg \min \{\Psi(x) : x \in R^n\}$. Для простоты будем считать, что x^* , \tilde{x} – единственные точки минимума задач (2.1) и (2.43). Предположим, что утверждение теоремы неверно, т.е. $\tilde{x} \neq x^*$. Рассмотрим возможные варианты: $\tilde{x} \in C$ и $\tilde{x} \notin C$.

1) $\tilde{x} \in C$, тогда $\Psi(\tilde{x}) = f(\tilde{x}) > f(x^*) \geq \Psi(x^*)$. Это противоречит предположению о том, что $\tilde{x} = \arg \min \{\Psi(x) : x \in R^n\}$.

2) $\tilde{x} \notin C$, тогда $\Psi(\tilde{x}) = \chi_E(\tilde{x}) = E + (f(\bar{x}) - E) \frac{\|\tilde{x} - x^0\|}{\|\bar{x} - x^0\|}$, где $\bar{x} = \pi_C(\tilde{x})$. В

силу (2.41) выполняется $f(\bar{x}) - E > \delta$. Откуда $H(\tilde{x}) > f(\bar{x}) \geq H(\bar{x})$, что опять противоречит предположению об экстремальности точки \tilde{x} .

Таким образом, теорема справедлива, если последовательность x^k , $k = 0, 1, \dots$, имеет предельные точки.

Рассмотрим второй случай – предположим, что последовательность x^k , $k = 0, 1, \dots$, не имеет предельных точек. Очевидно, что тогда для любого $r > 0$ существует k_r такое, что $\|x^k - x^0\| > r$, $k > k_r$. В силу ограниченности

множества C для любого $\rho > 0$ можно подобрать величину r , что

будет выполняться $\frac{\|x^k - x^0\|}{\|\bar{x}^k - x^0\|} > \rho$ и, учитывая (2.41),

$$\Psi(x^k) = E + (f(\bar{x}^k) - E) \frac{\|x^k - x^0\|}{\|\bar{x}^k - x^0\|} \geq E + \delta\rho, \quad k > k_r, \quad (2.44)$$

где $\bar{x}^k = \pi_C(x^k)$.

В силу сходимости последовательности $\Psi(x^k) \rightarrow \Psi^*$, $k \rightarrow \infty$, для любого $\varepsilon > 0$ существует k_ε такое, что

$$\Psi(x^k) - \Psi^* \leq \varepsilon, \quad k > k_\varepsilon. \quad (2.45)$$

Понятно, что всегда можно подобрать величину ρ , чтобы выполнялось $\Psi(x^k) \geq E + \delta\rho > f(x^0) + \varepsilon \geq \Psi^* + \varepsilon$, что противоречит (2.45) и предположению о том, что последовательность x^k , $k = 0, 1, \dots$, не имеет предельных точек.

Теорема доказана. ■

При неизвестных f^* и E^* , значение E необходимо уточнять итеративно. Рассмотрим модификацию \bar{A} исходного алгоритма. Обозначим E^k значение параметра продолжения целевой функции, используемого на итерации k , x^k – текущую точку итерации k . Пусть заданы начальное значение E^0 и параметры $q > 1$, $\delta > 0$, $B > 0$. Каждая итерация k алгоритма \bar{A} состоит из итерации алгоритма A , примененного к функции ψ^E , и дополнительных действий:

1) если $x^k \in S$, полагается $E^{k+1} = E^k$ и осуществляется переход к следующей итерации алгоритма \bar{A} ,

2) вычисляются в текущей точке x^k кроме значения функции ψ^E также величины $E_1 = f(\bar{x}^k) - \delta$ и $E_2 = f(\bar{x}^k) - f'(\bar{x}^k, p(\bar{x}^k)) \cdot \|\bar{x} - x^0\|$,

$\bar{E} = \min\{E_1, E_2\}$, где $\bar{x}^k = \pi_S(x^k)$,

3) если $E^k < \bar{E}$, полагается $E^{k+1} = E^k$ и осуществляется переход к следующей итерации алгоритма \bar{A} ,

4) полагается $E^{k+1} = E^k - q \max\{E^k - \bar{E}, B\}$, алгоритм A запускается из текущей точки x^k для уточненной функции ψ^E (с новым значением $E^0 = E^{k+1}$).

Алгоритм \bar{A} завершает работу, когда срабатывают критерии останова алгоритма A .

Нетрудно видеть, что пункт 4 алгоритма \bar{A} выполняется конечное число раз, после чего условия (2.41), (2.42) выполняются на всех итерациях, т.е. в силу теоремы 5 алгоритм сходится к оптимальному решению. Количество срабатываний пункта 4 алгоритма \bar{A} зависит от величины параметров q, B .

В рассмотренном алгоритме базовая точка x^0 , относительно которой строится продолжение функции f , считалась зафиксированной. Первоначальное значение этой точки может оказаться неудачным, что приведет к необходимости выбора больших (по абсолютной величине) значений величины E .

Рассмотрим задачу уточнения базовой точки после некоторого числа итераций алгоритма. Пусть на предыдущих итерациях сгенерированы точки $x^k, \bar{x}^k = \pi_S(x^0, x^k), k = 1, \dots, K$. Обозначим y уточненную базовую точку. Будем считать, что y принадлежит выпуклой оболочке точек $\bar{x}^k, k = 1, \dots, K$.

Нетрудно видеть, что для величины E должны выполняться следующие неравенства

$$E \leq f(\bar{x}^k) - (g_f(\bar{x}^k), \bar{x}^k - y), \quad k = 1, \dots, K, \quad (2.46)$$

$$f(\bar{x}^k) - E \geq 0, \quad k = 1, \dots, K. \quad (2.47)$$

Естественно стремиться к тому, чтобы разница $f^* - E$ по ходу работы алгоритма не принимала больших значений. Учитывая, что

$$y = \sum_{k=1}^K \lambda_k \bar{x}^k, \lambda_k \geq 0, \sum_{k=1}^K \lambda_k = 1, \text{ получаем, что для выбора базовой точки } y$$

нужно решать задачу линейного программирования: найти

$$\max_{\lambda, E} E \quad (2.48)$$

при ограничениях (2.46), (2.47) и дополнительных ограничениях

$$y = \sum_{k=1}^K \lambda_k \bar{x}^k, \quad (2.49)$$

$$\sum_{k=1}^K \lambda_k = 1, \quad (2.50)$$

$$\lambda_k \geq 0, k = 1, \dots, K. \quad (2.51)$$

Уточнение базовой точки может производиться периодически, для заданного интервала итераций алгоритма, или при нарушении дополнительных заданных условий.

Отличительной особенностью алгоритмов, построенных на предложенной процедуре выпуклого продолжения, является их устойчивость относительно некоторых преобразований неравенств, описывающих ограничения решаемой задачи, и оставляющих допустимую область без изменений. В программной реализации предлагаемого подхода базовая точка x^0 считается заданной и не изменяется в ходе вычислений, в качестве алгоритма безусловной минимизации использовался r -алгоритм [180, 176], для определения точки \bar{x} (пересечения отрезка $[x^0, x]$ с границей множества S) использовался дихотомический поиск. Точность определения точки \bar{x} фиксирована и является параметром алгоритма. Разработанные программные средства обеспечивают интерфейс со стандартной

программной средой AMPL [31]. Это позволило в ходе вычислительного эксперимента провести сравнение с существующими, как коммерческими, так и некоммерческими пакетами программ.

Целью вычислительного эксперимента было:

сравнение метода негладких штрафов, использующего r -алгоритм, и предлагаемого подхода на задачах, функции которых определены на всем пространстве R^n ;

сравнение методов негладкой оптимизации и различных современных пакетов программ на плохо обусловленных, плохо отмасштабированных задачах;

сравнение предлагаемого подхода и различных современных пакетов программ на задачах, функции которых определены на ограниченных множествах.

Вычислительный эксперимент проводился на тестовых задачах, сформированных на основе базовых задач (приведены ниже) вида: найти

$$f^* = \min f^0(x) \quad (2.52)$$

при ограничениях

$$f_k(x) \leq 0, \quad k = 1, \dots, m, \quad (2.53)$$

где $x = (x_1, x_2, \dots, x_n)$.

Тестовые задачи формировались путем замены ограничений вида (2.53) на

$$\varphi_k(x) f_k(x) \leq 0, \quad k = 1, \dots, m, \quad (2.54)$$

где $\varphi_k(x) > 0$, $k = 1, \dots, m$ (допустимое множество задачи при этом не изменяется).

Функции $\varphi_k(x)$ имеют вид

$$\varphi_k(x) = \begin{cases} \left(\mu + \|x - x^*\|^2 \right)^\gamma, & k = 1, \dots, m_1, \\ 1, & k = m_1 + 1, \dots, m \end{cases} \quad (2.55)$$

или

$$\varphi_k(x) = \alpha + \sin\left(\beta / \left(\mu + (x_k - x_k^*)^2\right)\right), k = 1, \dots, m, \quad (2.56)$$

где $\alpha > 1$, $\mu > 0$, $m_1 = [m/2]$, x^* – оптимальное решение базовой задачи, x_k, x_k^* – k -е компоненты векторов x, x^* .

Целевая функция заменялась следующей

$$\tilde{f}_0(x) = \begin{cases} f_0(x), & \text{если } f_k(x) \leq \varepsilon : k = 1, \dots, m, \\ \sqrt{-\max\{f_k(x) : k = 1, \dots, m\}}, & \text{в противном случае,} \end{cases} \quad (2.57)$$

где $\varepsilon \geq 0$. В допустимой области $f_0(x)$ и $\tilde{f}_0(x)$ совпадают, вне допустимой области при $\varepsilon = 0$ функция $\tilde{f}_0(x)$ не определена.

При использовании функций $\varphi_k(x)$ вида (2.55) тестовая задача имеет вырожденное масштабирование в окрестности оптимального решения, если $\gamma \neq 0$. В случае, когда функции $\varphi_k(x)$ имеют вид (2.56), штрафная функция тестовой задачи становится многоэкстремальной, если $\beta \neq 0$.

Базовая задача 1. Найти

$$f^* = \min(c, x) \quad (2.58)$$

при ограничениях

$$x_i - \frac{\chi}{n} \sum_{j=1}^n x_j \leq 0, i = 1, \dots, n, \quad (2.59)$$

где $\chi > 1$, $c_i = \frac{i}{10} > 0, i = 1, \dots, n$.

Допустимое множество есть конус, содержащий вектор $(1, \dots, 1)$. При $\chi = 1$ конус вырождается в луч, порожденный вектором $(1, \dots, 1)$. Базовая точка $x^0 = (1, \dots, 1)$. Решение: $x^* = 0, f^* = 0$.

Базовая задача 2. Найти

$$f^* = \min\{f^0(x) : x \in R^n\}, \quad (2.60)$$

где $f^0(x) = \max \{f_k(x) : k = 1, \dots, n\}$, $f_k(x) = \sum_{i=1}^n (x_i - x_i^k)^2$,

$x^k = (x_1^k, \dots, x_n^k)$ – заданные точки, $k = 1, \dots, n$. Здесь $x_i^k = 0$, если $i \neq k$, $x_i^k = 1$, если $i = k$, $k = 1, \dots, n$, $i = 1, \dots, n$.

Эквивалентная постановка

$$\min Y, \quad (2.61)$$

при ограничениях

$$f_k(x) - Y \leq 0, \quad k = 1, \dots, n. \quad (2.62)$$

Базовая точка $x^0 = (1, 0, \dots, 0)$, $y^0 = 2$. Решение:

$$x^* = \left(\frac{1}{n}, \dots, \frac{1}{n}\right), \quad f^* = y^* = \frac{n-1}{n}.$$

Базовая задача 3. Найти

$$f^* = \min_x \left\{ \lambda \cdot (p, x) - \sqrt{\delta^2 \cdot (p, x)^2 - \|x - (x, p) \cdot p\|^2} \right\} \quad (2.63)$$

при ограничениях

$$\delta^2 \cdot (p, x)^2 \geq \|x - (x, p) \cdot p\|^2 + \sigma^2, \quad x \in R^n, \quad (2.64)$$

$$(p, x) \geq 0, \quad x \in R^n, \quad (2.65)$$

где p – заданный вектор, $\|p\| = 1$, $p_i = 1/\sqrt{n}$, $i = 1, \dots, n$, $\lambda > \delta > 0$, $\sigma > 0$ – заданные числа. Норма градиента целевой функции на границе допустимого множества при $\sigma = 0$ равна $+\infty$.

Базовая точка x^0 выбиралась с некоторым смещением от луча, порожденного вектором p .

Если x^* – оптимальное решение, то $x^* = pt^*$, где $t^* \in R$. Откуда $(p, x) = t^*$, $\delta \cdot t^* = \sigma$. Решение: $x^* = p \cdot \frac{\sigma}{\delta}$, $f^* = \lambda \cdot \frac{\sigma}{\delta} - \sigma = \sigma \left(\frac{\lambda}{\delta} - 1\right)$.

Все тестовые задачи были реализованы в языке AMPL и при сравнении с существующими пакетами программ решались на сервере NEOS (<http://www-neos.mcs.anl.gov/>). Число переменных во всех задачах

(размерность вектора x) равно 50. В качестве начальной точки для всех пакетов программ использовалась базовая точка x^0 . Для метода выпуклых продолжений и метода негладких штрафных функций использовались одинаковые параметры r -алгоритма [176, 180]. Для всех пакетов программ использовались стандартные настройки. Результаты вычислительных экспериментов приведены ниже. Если пакет программ не смог решить текущую задачу, или полученное решение нарушает ограничения задачи, то вместо значения целевой функции ставится символ F (fail).

В табл. 2.1–2.4 приводятся результаты вычислительных экспериментов по базовой задаче 1.

Таблица 2.1.

Сравнение метода выпуклых продолжений и метода негладких штрафных функций на исходной задаче 1. Штрафной коэффициент 1000

χ	Метод	Рекордное значение целевой функции	Число вызовов функции
1	2	3	4
1.50	Выпуклое продолжение	0.0006658	875
	Негладкие штрафы	0.0010041	1259
1.45	Выпуклое продолжение	0.0004572	922
	Негладкие штрафы	0.0010863	1188
1.40	Выпуклое продолжение	0.0015920	990
	Негладкие штрафы	0.0026862	959
1.35	Выпуклое продолжение	0.0015888	827
	Негладкие штрафы	0.0023529	1037
1.30	Выпуклое продолжение	0.0009020	994
	Негладкие штрафы	0.0023775	1036
1.25	Выпуклое продолжение	0.0011574	1098
	Негладкие штрафы	0.0066865	1018
1.20	Выпуклое продолжение	0.0011774	1096
	Негладкие штрафы	0.0011554	1045
1.15	Выпуклое продолжение	0.0068407	882
	Негладкие штрафы	0.0027255	1044

Продолж. табл. 2.1.

1	2	3	4
1.10	Выпуклое продолжение	0.0080652	992
	Негладкие штрафы	F	668
1.05	Выпуклое продолжение	0.0026539	1162
	Негладкие штрафы	F	461

В табл. 2.1 сравниваются метод выпуклых продолжений и метод негладких штрафных функций. Для $\chi = 1.10$ и $\chi = 1.05$ штрафная функция не ограничена снизу. Это связано с необходимостью подбирать штрафной коэффициент индивидуально для каждой задачи. Метод выпуклых продолжений обеспечил несколько более высокую точность по целевой функции и меньшее число вызовов функции в r -алгоритме, чем при использовании метода негладких штрафных функций. Каждый вызов при этом существенно более трудоемкая процедура, поскольку решается задача одномерного поиска.

В табл. 2.2 сравниваются различные пакеты программ на задаче, имеющей вырожденность масштабирования в оптимальной точке. При отрицательных γ все программные средства, кроме метода выпуклых продолжений, находят неверные решения. Значение целевой функции на решении, найденном методом выпуклых продолжений, также зависит от значения параметра γ . При положительных γ различия в решениях, найденных различными пакетами программ, не столь существенны. В табл. 2.3 приводятся результаты сравнения различных пакетов программ на задаче, в которой к вырожденности масштабирования в оптимальной точке добавлена ограниченность области определения целевой функции. При положительных γ результаты качественно такие же, как и в табл. 2.2.

Таблица 2.2.

Сравнение различных пакетов программ (рекордное значение целевой функции) на задаче 1, $\mu = 1e-16$, $\chi = 1.1$, штрафной коэффициент 1000 (вырожденность масштабирования в точке x^*)

γ	snopt	minos	loqo	Выпуклое продолжение	Негладкие штрафы
-2.5	F	F	F	0.4629	F
-2	F	F	F	0.2354	F
-1.5	F	F	F	0.0856	F
-1	F	F	F	0.01468	F
-0.5	F	F	F	0.0078	0.0199
0	0	0	0	0.0080	F
0.5	0.0104	0.00006	F	0.0017	F
1	0.2359	0.0258	F	0.0206	F
1.5	0.8902	0.1621	F	0.0899	F
2	1.6615	0.6147	F	0.2741	F
2.5	F	11.1109	F	0.5577	F
3	3.1324	F	F	0.9132	F

Результаты сравнения пакетов программ на задачах, содержащих осциллирующий множитель в функции ограничений (табл. 2.4) также показывают преимущества метода выпуклых продолжений. Результаты экспериментов по базовой задаче 2 аналогичны результатам по задаче 1 и поэтому не приводятся.

Таблица 2.3.

Сравнение различных пакетов программ (рекордное значение целевой функции) на задаче 1, $\varepsilon = 0.00001$, $\mu = 1e-16$, $\chi = 1.1$ (вырожденность масштабирования в точке x^* , ограниченность области определения целевой функции)

γ	snopt	minos	loqo	Выпуклое продолжение
0.0	0.0000	0.0000	F	0.0080
0.5	0.0104	0.00006	F	0.0017
1.0	0.2359	F	F	0.0206
1.5	1.0088	F	F	0.0899
2.0	1.6615	F	F	0.2741
2.5	1.2714	F	2.7193	0.5577
3.0	F	F	4.646	0.9132

Базовая задача 3 специально подбиралась как задача трудная для метода выпуклых продолжений. Вычислительный эксперимент проводился непосредственно на базовой задаче. Результаты приведены в табл. 2.5 – метод выпуклых продолжений оказался более устойчивым для плохо обусловленных задач.

Результаты сравнительного вычислительного анализа показывают существенную устойчивость разработанного подхода к плохой обусловленности задач. Используемые тестовые задачи и предварительная версия программных средств размещены для свободного доступа на сайте http://www.icyb.kiev.ua/file/NonDiffOpt_Nurm/indexk.htm.

Таблица 2.4.

Сравнение пакетов программ (рекордное значение целевой функции) на задаче 1, $\alpha = 1.1$, $\mu = 1e-16$, $\chi = 1.15$ (осциллирующий множитель в функции ограничений)

β	snopt	minos	loqo	Выпуклое продолжение	Негладкие штрафы
0.	0	0	0	0.0068	0.0027
1	F	F	F	0.0816	F
2	F	F	F	0.1210	F
3	F	F	F	0.1326	F
4	F	F	F	0.1714	F
5	36.74	F	F	0.1497	F

Таблица 2.5.

Сравнение пакетов программ (рекордное значение целевой функции) на задаче 3 ($\lambda = 0.02$)

σ	Оптимальное значение	snopt	loqo	Выпуклое продолжение
1	2	3	4	5
$\delta = 0.00005$				
0.004	1.5960	F	1.6049	1.5968
0.003	1.1970	F	1.2126	1.1970
0.002	0.7980	F	0.8164	0.7994
0.001	0.3990	F	0.4190	0.3990
$\delta = 0.00004$				
0.004	1.9960	F	2.0040	1.9960
0.003	1.4970	F	1.5165	1.4970
0.002	0.9980	F	1.0191	0.9980
0.001	0.4990	F	0.5191	0.4990

Продолж. табл. 2.5.

1	2	3	4	5
$\delta = 0.00003$				
0.004	2.6626	F	2.6686	2.6626
0.003	1.9970	F	F	1.9970
0.002	1.3313	F	F	1.3328
0.001	0.6656	F	0.6852	0.6657

2.3. Коническая аппроксимация функций

Подход, предложенный в предыдущем подразделе, оказался достаточно эффективным, разработанные программы нечувствительны к плохому масштабированию задач и показали преимущества по сравнению с известными программными средствами при решении плохо обусловленных классов задач. В данном подразделе будут рассмотрены обобщения этого подхода.

Как и ранее рассматривается исходная задача (2.1).

Пусть задано некоторое число $E < f(x^0)$. Рассмотрим введенную в предыдущем разделе коническую аппроксимацию $\gamma_E(x)$ функции f на множестве C . Функция $\gamma_E(x)$ выпуклая и принимает конечные значения при любых $x \in R^n$.

Нетрудно видеть, что для произвольной точки $x \in R^n$, $x \neq x^0$, на луче, выходящем из точки x^0 и проходящем через x , найдется точка \bar{x} , $\bar{x} \in C$ (возможно не одна) такая, что $f(\bar{x}) = \gamma_E(\bar{x})$. Обозначим $\mu_E(x)$ такую точку, ближайшую к x^0 . Положим

$$\varphi_E(x) = \begin{cases} f(x), & \text{если } \|x - x^0\| \leq \|\mu_E(x) - x^0\|, \\ \gamma_E(x), & \text{если } \|x - x^0\| > \|\mu_E(x) - x^0\|. \end{cases} \quad (2.66)$$

Лемма 2.3. Функция $\gamma_E(x)$ может быть представлена в виде

$$\gamma_E(x) = E + (f(\mu_E(x)) - E) \frac{\|x - x^0\|}{\|\mu_E(x) - x^0\|}. \quad (2.67)$$

Доказательство следует из того, что графиком функции $\tilde{\gamma}_E(t) = \gamma_E(x^0 + pt)$, $p = \frac{x - x^0}{\|x - x^0\|}$ является линейная функция, принимающая значения E в точке $t = 0$ и $f(\mu_E(x))$ в точке $t = \|\mu_E(x) - x^0\|$. ■

Рассмотрим задачу: найти

$$\varphi_E^* = \inf \{ \varphi_E(x) : x \in R^n \}. \quad (2.68)$$

Теорема 2.4. Пусть $E \leq f^*$, тогда $\varphi_E^* = f^*$, и решения задач (2.68) и (2.1) совпадают.

Доказательство. По построению $f(x) \geq \gamma_E(x)$ и $f(x) \geq \varphi_E(x)$ для любых $x \in C$. Таким образом $f^* \geq \varphi_E^*$. Заметим также, что $\{x : \|x - x^0\| \leq \|\mu_E(x) - x^0\|\} \subseteq C$. Рассмотрим точку x такую, что $\|x - x^0\| > \|\mu_E(x) - x^0\|$ (т.е. $\varphi_E(x) = \gamma_E(x)$). Поскольку $f(\mu_E(x)) - E \geq f^* - E > 0$ и $\|x - x^0\| = \|x - \mu_E(x)\| + \|\mu_E(x) - x^0\| > \|\mu_E(x) - x^0\|$, то из (2.67) следует, что $\gamma_E(x) > f(\mu_E(x))$. Отсюда получаем

$$\begin{aligned} \varphi_E^* &= \inf \{ \varphi_E(x) : x \in R^n \} = \inf \{ \varphi_E(x) : \|x - x^0\| \leq \|\mu_E(x) - x^0\| \} = \\ &= \inf \{ f(x) : \|x - x^0\| \leq \|\mu_E(x) - x^0\| \} \geq \inf \{ f(x) : x \in C \} \geq f^*, \text{ т.е. } \varphi_E^* = f^*. \blacksquare \end{aligned}$$

Теорема 2.5. Пусть C – выпуклый компакт, $C \subseteq \text{dom } f$, $x^0 \in \text{int } C$, $E < f(x^0)$. Тогда $\varphi_E : R^n \rightarrow R$ – выпуклая функция.

Доказательство. Рассмотрим надграфик $\Phi(E)$ функции $\varphi_E(x)$. Нетрудно видеть, что

$$\Phi(E) = \{ v : v \in R \times R^n, v = z_E^0 + \alpha(z - z_E^0), \alpha \geq 1, z \in F \}, \quad (2.69)$$

где, как и ранее, F – надграфик функции f на множестве C ,

$z_E^0 = (E, x^0)$. Пусть $u, w \in \Phi(E)$, покажем, что $v = \delta u + (1 - \delta)w \in \Phi(E)$ для любых $0 \leq \delta \leq 1$. В силу (2.69) существуют $\bar{u}, \bar{w} \in F$ и коэффициенты $\alpha_u, \alpha_w \geq 1$, для которых имеет место $u = z_E^0 + \alpha_u(\bar{u} - z_E^0)$, $w = z_E^0 + \alpha_w(\bar{w} - z_E^0)$.

Откуда $v = \delta u + (1 - \delta)w = z_E^0 + (\delta\alpha_u + (1 - \delta)\alpha_w) \left[\frac{\delta\alpha_u\bar{u} + (1 - \delta)\alpha_w\bar{w}}{\delta\alpha_u + (1 - \delta)\alpha_w} - z_E^0 \right]$.

Полагая $\bar{\delta} = \frac{\delta\alpha_u}{\delta\alpha_u + (1 - \delta)\alpha_w}$, $\bar{\alpha} = \delta\alpha_u + (1 - \delta)\alpha_w$, $\bar{v} = \bar{\delta}\bar{u} + (1 - \bar{\delta})\bar{w}$, получаем

$v = z_E^0 + \bar{\alpha}(\bar{v} - z_E^0)$. Очевидно, что $0 \leq \bar{\delta} \leq 1$, $\bar{\alpha} \geq 1$, откуда $\bar{v} \in F$, $v \in \Phi(E)$.

Таким образом, $\varphi_E(x)$ выпуклая функция. Эта функция принимает конечные значения при любых x , поскольку таким свойством обладает функция $\gamma_E(x)$. ■

Для использования рассматриваемого подхода в алгоритмах решения оптимизационных задач должны быть определены эффективные процедуры вычисления значений функции $\varphi_E(x)$ и ее субградиентов.

Обозначим $f'(y, p)$ производную функции f в точке $y \in C$ по направлению p . Пусть зафиксирована некоторая точка $x \in C$, положим

$$p = \frac{x - x^0}{\|x - x^0\|}, \quad t^* = \|\mu_E(x) - x^0\|.$$

Лемма 2.4. Имеет место соотношение

$$t^* = \sup \left\{ t : \frac{f(x^0 + tp) - E}{t} > -f'(x^0 + tp, -p), t \geq 0, x^0 + tp \in C \right\}. \quad (2.70)$$

Доказательство. Пусть $t > 0$, $x^0 + pt \in C$. Покажем, что

$$\frac{f(x^0 + tp) - E}{t} > f'(x^0 + tp, p), \text{ если } t < t^*, \quad (2.71)$$

$$\frac{f(x^0 + tp) - E}{t} \leq -f'(x^0 + tp, -p), \text{ если } t > t^*. \quad (2.72)$$

Обозначим $\bar{f}(t) = f(x^0 + pt)$, $\bar{f}'_+(t)$ – производная функции $\bar{f}(t)$ в точке t при сдвиге в направлении увеличения аргумента, $\bar{f}'_-(t)$ – производная при сдвиге в направлении уменьшения аргумента, $t \geq 0$, $x^0 + pt \in C$.

Пусть зафиксировано некоторое $\bar{t} > 0$, $x^0 + p\bar{t} \in C$. Рассмотрим линейную функцию $\tilde{\gamma}(t)$ такую, что $\tilde{\gamma}(0) = E$, $\tilde{\gamma}(\bar{t}) = \bar{f}(\bar{t})$. Нетрудно видеть, что если $\bar{t} = t^*$, то $\tilde{\gamma}(t)$ является касательной к функции $\bar{f}(t)$ в этой точке (если точка касания не одна, то величина t^* соответствует точке касания, ближайшей к x^0), если $\bar{t} > t^*$, то $\tilde{\gamma}(t^*) \geq \bar{f}(t^*)$, если $\bar{t} < t^*$, то $\tilde{\gamma}(t^*) > \bar{f}(t^*)$.

Пусть $\bar{t} < t^*$, тогда $\bar{f}(\bar{t}) + \bar{f}'_+(\bar{t}) \cdot (t^* - \bar{t}) \leq \bar{f}(t^*) < \tilde{\gamma}(t^*) = \bar{f}(\bar{t}) + \frac{(\bar{f}(\bar{t}) - E)}{\bar{t}}(t^* - \bar{t})$. Отсюда следует (2.71).

Для случая $\bar{t} > t^*$ аналогично $\tilde{\gamma}(t^*) = \bar{f}(\bar{t}) + \frac{(\bar{f}(\bar{t}) - E)}{\bar{t}}(t^* - \bar{t}) \geq \bar{f}(t^*) \geq \bar{f}(\bar{t}) + \bar{f}'_-(\bar{t}) \cdot (\bar{t} - t^*)$.

Из выпуклости $f(x)$ имеем $f'(x^0 + tp, p) \geq -f'(x^0 + tp, -p)$. Откуда, с учетом (2.71), (2.72), получаем (2.70). ■

Теорема 2.6. Пусть точка x такая, что $\bar{x} = \mu_E(x)$ – внутренняя точка множества C . Тогда в точке \bar{x} существует субградиент \bar{g} функции f , для которого выполняется

$$f(\bar{x}) - E = (\bar{g}, \bar{x} - x^0), \quad (2.73)$$

и вектор \bar{g} есть субградиент функции γ_E в точке x .

Доказательство. Из соотношений $\bar{x} = \mu_E(x) = x^0 + t^*p$, $f'(x^0 + tp, p) \geq -f'(x^0 + tp, -p)$, (2.71), (2.72) получаем,

$$-f'(\bar{x}, -p) \leq \frac{f(\bar{x}) - E}{t^*} \leq f'(\bar{x}, p). \quad (2.74)$$

Очевидно, в точке \bar{x} существуют субградиенты g^1, g^2 (возможно $g^1 = g^2$) функции f , для которых $(g^1, -p) = -f'(\bar{x}, -p)$, $(g^2, p) = f'(\bar{x}, p)$. Нетрудно видеть, что из (2.74) следует существование такого α , $0 \leq \alpha \leq 1$, что для вектора $\bar{g} = \alpha g^1 + (1 - \alpha)g^2$ будет выполняться $\frac{f(\bar{x}) - E}{t^*} = (\bar{g}, p)$. Вектор \bar{g} есть субградиент функции f в точке \bar{x} . Учитывая, что $t^* p = \bar{x} - x^0$, получаем (2.73).

Докажем, что вектор \bar{g} есть субградиент функции γ_E в точке \tilde{x} . Для этого достаточно показать, что надграфик $K(E)$ функции γ_E принадлежит надграфику K_L функции $\gamma_L(x) = E + (\bar{g}, x - x^0)$. Надграфик K_L есть полупространство в R^{n+1} . Пусть точка $z^1 \in R^{n+1}$ лежит на границе множества K_L , а $z^2 \in K_L$. Тогда, очевидно, луч, исходящий из z^1 и порожденный вектором $z^2 - z^1$, содержится в K_L . Нетрудно видеть, что надграфик F функции f принадлежит K_L , точка $z_E^0 = (E, x^0)$ лежит на границе множества K_L . Откуда следует, что любая точка $v = z_E^0 + \alpha(z - z_E^0)$, $\alpha \geq 0, z \in F$, также принадлежит K_L . Т.е. надграфик $K(E)$ функции γ_E принадлежит надграфику K_L . ■

Субградиент \bar{g} функции f в точке $\bar{x} = \mu_E(x) \in \text{int dom } C$, удовлетворяющий соотношению (2.73), может быть определен в ходе решения задачи одномерного поиска (2.70).

Теорема 2.7. Пусть точка x такая, что $\bar{x} = \mu_E(x)$ принадлежит границе множества C , $x \in \text{int dom } f \cap \text{int dom } h$, $h(x^0) < 0$, $g_f(\bar{x})$, $g_h(\bar{x})$ – субградиенты функций f и h в точке \bar{x} . Тогда $(g_h(\bar{x}), x - \bar{x}) \neq 0$ и вектор

$$g = g_f(\bar{x}) + \frac{E - f(\bar{x}) - (g_f(\bar{x}), x^0 - \bar{x})}{(g_h(\bar{x}), x^0 - \bar{x})} g_h(\bar{x}) \quad (2.75)$$

есть субградиент функции $\gamma_E(x)$ в точке x .

Доказательство аналогично доказательству теоремы 2.2.

При условии $E \leq f^*$ задачу (2.68) будем называть конической регуляризацией задачи (2.1). При этом условии для решения задач (2.68) может использоваться любой метод безусловной минимизации выпуклых функций. Заметим, что если точка x^0 близка к решению x^* задачи (2.1), а разность $f^* - E$ невелика, то поведение функции $\varphi_E(x)$ определяется поведением функции $f(x)$ в окрестности решения x^* . Приведенные утверждения позволяют эффективно вычислять значения функции $\varphi_E(x)$ (требуется решить задачу одномерного поиска (2.70)) и ее субградиенты (см. теоремы 2.6, 2.7). Если значение f^* заранее не известно, то величина E должна уточняться по ходу оптимизации.

2.4. Выводы

Рассмотрены выпуклые задачи оптимизации с ограничениями в случае, когда функции, описывающие задачи, определены на ограниченном множестве. Предложены новые эффективные методы формирования задач безусловной оптимизации, эквивалентных исходным задачам.

Первый метод – конического продолжения целевой функций с допустимого множества на все пространство. Свойства порождаемой функции $\psi_E(x)$ зависят от одного числового параметра E . Сформулированы условия, зависящие от поведения исходной целевой функции на границе допустимого множества, при которых существует конечная величина E^* такая, что при $E \leq E^*$ порождаемая функция $\psi_E(x)$ выпукла.

Второй метод – коническая аппроксимация целевой функции. Формируемая функция $\varphi_E(x)$ выпукла при естественных условиях. При

условии $E < f^*$, где f^* – оптимальное значение исходной задачи, минимумы функций $\psi_E(x)$ и $\varphi_E(x)$ совпадают с f^* . Уточнение параметра E может выполняться по ходу решения задачи безусловной оптимизации.

РАЗДЕЛ 3. ИСПОЛЬЗОВАНИЕ ТОЧНЫХ ШТРАФНЫХ ФУНКЦИЙ

Метод точных штрафных функций является одним из основных средств для решения оптимизационных задач с ограничениями с помощью r -алгоритма Н.З. Шора. Однако использование точных штрафных функций связано с определенными проблемами – отсутствуют простые методики вычисления приемлемых значений штрафных коэффициентов. Выбор значений коэффициентов, обычно, возлагается на пользователя, что приводит либо к завышению используемых значений, либо к необходимости многократного решения одной и той же задачи для удовлетворительного подбора штрафных коэффициентов. В данном разделе предлагаются подходы, позволяющие построить процедуру автоматического определения значений штрафных коэффициентов по ходу работы алгоритма безусловной оптимизации.

3.1. Штрафные коэффициенты и производные по направлению

Рассматривается задача: найти

$$f^* = \min \{ f(x) : x \in C \}, \quad (3.1)$$

здесь $C = \{ x \in R^n : h(x) \leq 0 \}$, $f, h : R^n \rightarrow R$ – выпуклые функции, принимающие конечные значения при любых x , C – ограниченное замкнутое множество. Будем рассматривать штрафные функции вида

$$F_\lambda(x) = f(x) + \lambda \cdot h^+(x), \quad \lambda \in R, \lambda \geq 0, \quad (3.2)$$

где $h^+ = \max \{ 0, h \}$, и выпуклую задачу: найти

$$F_\lambda^* = \min \{ F_\lambda(x) : x \in R^n \}. \quad (3.3)$$

Для выпуклых задач штрафная функция $F_\lambda(x)$ называется точной при заданном значении штрафного коэффициента λ , если $F_\lambda^* = f^*$ и решения задач (3.1) и (3.3) совпадают.

Достаточные условия существования точных штрафных функций в общем случае, когда функции f и h определены на произвольном метрическом пространстве, приведены в [77]. Для выпуклых функций обозначим $F'_\lambda(x, p)$, $f'(x, p)$, $h'(x, p)$ односторонние производные функций F_λ , f , h в точке $x \in R^n$ по направлению p . Следуя [77], обозначим $h^\downarrow(x)$ – скорость наискорейшего спуска функции h в точке x ,

$$h^\downarrow(x) = \min \{h'(x, p) : \|p\| = 1\}.$$

Пусть для выпуклой задачи (3.1) выполняется условие Слейтера, тогда существуют числа $\lambda^* \geq 0$, $a > 0$ такие, что $h^\downarrow(x) \leq -a$, $\forall x \notin C$, $x \in R^n$, и $F_\lambda(x)$ – точная штрафная функция при $\lambda > \lambda^*$.

Нетрудно видеть, что при $\lambda > \lambda^*$ для $\forall x \notin C$ выполняется

$$\min \{F'_\lambda(x, p) : \|p\| = 1\} < 0. \quad (3.4)$$

Обозначим $p(x, y)$ вектор, определяющий направление из точки x в точку y , $p(x, y) = (y - x) / \|y - x\|$, $y \neq x$. Предположим заданной допустимую точку y^0 такую, что $h(y^0) < 0$. Легко видеть, что если для $\forall x \notin C$ выполняется условие

$$F'_\lambda(x, p(x, y^0)) = f'(x, p(x, y^0)) + \lambda h'(x, p(x, y^0)) < 0, \quad (3.5)$$

то $F_\lambda(x)$ – точная штрафная функция. При этом значение λ может быть завышено по сравнению с λ^* .

Если при решении задачи (3.3) некоторым сходящимся алгоритмом порождаемые точки не попадают в множество C , то условие (3.4) (или условие (3.5)) можно было бы использовать для подбора значения λ , т.е. если это условие не выполняется для текущей точки $x \notin C$, коэффициент λ следует увеличить.

Однако можно построить примеры, когда данные условия выполняются для любой точки последовательности, формируемой оптимизационным

алгоритмом, но не выполняются для предельной точки (F_λ в таком случае может не быть точной штрафной функцией).

Для преодоления этой проблемы несколько изменим условие (3.5). Для точек $x \notin C$, $y \in C$ положим $\pi_C(x, y)$ – точка пересечения отрезка $[x, y]$ с границей множества C .

Теорема 3.1. Пусть заданы точка $y^0 \in C$, числа $\varepsilon > 0$, $\lambda > 0$, последовательность $x^k \in R^n$, $k = 0, 1, \dots$, сходится к решению \tilde{x} задачи (3.3) и для каждой точки x^k , $x^k \notin C$, $k = 0, 1, \dots$ выполняется условие

$$F_\lambda'(\bar{x}^k, p(\bar{x}^k, x^k)) = f'(\bar{x}^k, p(\bar{x}^k, x^k)) + \lambda \cdot h'(\bar{x}^k, p(\bar{x}^k, x^k)) \geq \varepsilon, \quad (3.6)$$

где $\bar{x}^k = \pi_C(x^k, y^0)$. Тогда \tilde{x} является решением задачи (3.1), т.е. $F_\lambda(x)$ – точная штрафная функция.

Доказательство. Понятно, что если $\tilde{x} \in C$, то \tilde{x} – решение задачи (3.1).

Предположим, что $\tilde{x} \notin C$. Без ограничения общности можно считать, что $x^k \notin C$ для всех $k = 0, 1, \dots$. В силу сходимости последовательности x^k , $k = 0, 1, \dots$ последовательность \bar{x}^k , $k = 0, 1, \dots$ сходится к точке $\tilde{z} = \pi_C(\tilde{x}, y^0)$. В силу выпуклости функции $F_\lambda(x)$ имеем

$$\frac{F_\lambda(\tilde{z}) - F_\lambda(\tilde{x})}{\|\tilde{x} - \tilde{z}\|} \geq F_\lambda'(\tilde{x}, p(\tilde{x}, \tilde{z})), \quad (3.7)$$

$$\frac{F_\lambda(x^k) - F_\lambda(\bar{x}^k)}{\|x^k - \bar{x}^k\|} \geq F_\lambda'(\bar{x}^k, p(\bar{x}^k, x^k)). \quad (3.8)$$

Обозначим $\tilde{\rho} = (F_\lambda(\tilde{x}) - F_\lambda(\tilde{z})) / \|\tilde{x} - \tilde{z}\|$, $\rho_k = (F_\lambda(x^k) - F_\lambda(\bar{x}^k)) / \|x^k - \bar{x}^k\|$. Из непрерывности $F_\lambda(x)$ имеем $\tilde{\rho} = \lim_{k \rightarrow \infty} \rho_k$, т. е. для $\forall \delta > 0$ существует K_δ такое, что $|\tilde{\rho} - \rho_k| < \delta$, если $k > K_\delta$. Откуда $\tilde{\rho} > \rho_k - \delta$, для $k > K_\delta$. Выбирая $\delta = \varepsilon/2$ и принимая во внимание (3.6) и (3.8), получаем $\tilde{\rho} \geq \varepsilon/2$.

Учитывая (3.7), имеем $F_{\lambda}'(\tilde{x}, p(\tilde{x}, \tilde{z})) \leq -\tilde{\rho} \leq -\varepsilon/2$. То есть в точке \tilde{x} существует направление убывания $p(\tilde{x}, \tilde{z})$, что противоречит предположению об оптимальности точки \tilde{x} . Теорема доказана. ■

Таким образом, для контроля значения штрафного коэффициента λ на каждом шаге оптимизационного алгоритма необходимо проверять условие (3.6). Это требует решения одномерной задачи поиска точки $\bar{x}^k = \pi_C(x^k, y^0)$ пересечения отрезка $[x^k, y^0]$ с границей множества C . Процедура такого поиска может быть реализована достаточно эффективно.

В случае, когда неравенство (3.6) на некоторой итерации алгоритма нарушается, будем увеличивать штрафной коэффициент λ так, чтобы неравенство (3.6) выполнилось. Это возможно, если $h'(\bar{x}^k, p(\bar{x}^k, x^k)) > 0$. Причем увеличение штрафного коэффициента будем производить на величину не менее B , где $B > 0$ – заданный параметр. Легко видеть, что если существует конечное $\bar{\lambda}$ такое, что при $\lambda > \bar{\lambda}$ неравенство (3.6) выполняется на всех итерациях алгоритма, то количество таких увеличений штрафного коэффициента по ходу работы оптимизационного алгоритма будет конечно. Условия существования конечного $\bar{\lambda}$ будут сформулированы в теореме 3.2.

$$\text{Положим } r(x, y) = \max \left\{ 0, -\frac{f'(x, p(x, y))}{h'(x, p(x, y))} \right\},$$

$$R(y^0) = \sup \{ r(\bar{x}, x) : \bar{x} = \pi_C(x, y^0), x \notin C \}, y^0 \in C. \quad (3.9)$$

Лемма Ошибка! Источник ссылки не найден.1. Пусть для точки $y^0 \in C$ выполняется $h(y^0) < 0$, тогда $R(y^0) < +\infty$.

Доказательство. В силу ограниченности множества C функция f липшицева, и существует $M > 0$ такое, что $|f'(\bar{x}, p(\bar{x}, x))| \leq M$ для любых $x \notin C, \bar{x} = \pi_C(x, y^0)$.

Положим $m = \max \left\{ \|\bar{x} - y^0\| : \bar{x} = \pi_C(x, y^0), x \notin C \right\}$. В силу

выпуклости функции h имеем $h'(\bar{x}, p(\bar{x}, x)) \geq \frac{h(\bar{x}) - h(y^0)}{\|\bar{x} - y^0\|}$. Откуда

$$h'(\bar{x}, p(\bar{x}, x)) \geq |h(y^0)| m^{-1}, \quad (3.10)$$

и $R(y^0) \leq \frac{mM}{|h(y^0)|}$. Лемма доказана. ■

Теорема 3.2. Пусть $h(y^0) < 0$. Тогда существует $\bar{\lambda} < \infty$ такое, что при любом λ , $\lambda > \bar{\lambda}$, найдется $\varepsilon(\lambda) > 0$, при котором для последовательности $x^k \in R^n$, $k = 0, 1, \dots$, сходящейся к решению \tilde{x} задачи (3.3), выполняется

$$F_\lambda'(\bar{x}^k, p(\bar{x}^k, x^k)) \geq \varepsilon(\lambda), \text{ если } x^k \notin C, k = 0, 1, \dots \quad (3.11)$$

где $\bar{x}^k = \pi_C(x^k, y^0)$. То есть \tilde{x} является решением задачи (3.1) и $F_\lambda(x)$ – точная штрафная функция.

Доказательство. Положим $\bar{\lambda} = R(y^0)$, $\lambda > \bar{\lambda}$. С учетом (3.10) нетрудно видеть, что

$$\begin{aligned} F_\lambda'(\bar{x}^k, p(\bar{x}^k, x^k)) &= f'(\bar{x}^k, p(\bar{x}^k, x^k)) + (\lambda + \bar{\lambda} - \bar{\lambda}) \cdot h'(\bar{x}^k, p(\bar{x}^k, x^k)) = \\ &= f'(\bar{x}^k, p(\bar{x}^k, x^k)) + \bar{\lambda} \cdot h'(\bar{x}^k, p(\bar{x}^k, x^k)) + (\lambda - \bar{\lambda}) \cdot h'(\bar{x}^k, p(\bar{x}^k, x^k)) = \\ &= h'(\bar{x}^k, p(\bar{x}^k, x^k)) \left[\bar{\lambda} - \frac{-f'(\bar{x}^k, p(\bar{x}^k, x^k))}{h'(\bar{x}^k, p(\bar{x}^k, x^k))} \right] + (\lambda - \bar{\lambda}) \cdot h'(\bar{x}^k, p(\bar{x}^k, x^k)) \geq \\ &\geq (\lambda - \bar{\lambda}) \cdot h'(\bar{x}^k, p(\bar{x}^k, x^k)). \end{aligned}$$

Учитывая (3.10) и полагая $\varepsilon = (\lambda - \bar{\lambda}) |h(y^0)| m^{-1}$, получаем (3.11). Отсюда из теоремы 3.1 следует, что $F_\lambda(x)$ – точная штрафная функция. Теорема доказана. ■

Приведенные результаты позволяют строить достаточно эффективные процедуры определения значений штрафных коэффициентов по ходу работы алгоритмов безусловной оптимизации. Однако при неудачном выборе точки y^0 штрафные коэффициенты могут устанавливаться достаточно большими.

Рассмотрим вопросы варьирования точки y^0 .

Теорема 3.3. Пусть множество C компактно, заданы числа $\varepsilon > 0$, $\lambda > 0$, последовательность $x^k \in R^n, k = 0, 1, \dots$, сходится к решению \tilde{x} задачи (3.3), каждой точке x^k поставлена в соответствие точка $y^k \in C$, и для каждого $x^k, x^k \notin C, k = 0, 1, \dots$ выполняется условие

$$F_\lambda'(\bar{x}^k, p(\bar{x}^k, x^k)) = f'(\bar{x}^k, p(\bar{x}^k, x^k)) + \lambda \cdot h'(\bar{x}^k, p(\bar{x}^k, x^k)) \geq \varepsilon, \quad (3.12)$$

где $\bar{x}^k = \pi_C(x^k, y^k)$. Тогда \tilde{x} является решением задачи (3.1), т. е. $F_\lambda(x)$ – точная штрафная функция.

Доказательство. Без ограничения общности можно считать, что $y^k, k = 0, 1, \dots$, сходящаяся последовательность (в силу ограниченности C всегда можно выделить сходящуюся подпоследовательность).

Предположим, что $\tilde{x} \notin C$. Обозначим $\tilde{y} = \lim_{k \rightarrow \infty} y^k$. В силу сходимости последовательностей $x^k, y^k, k = 0, 1, \dots$, последовательность $\bar{x}^k, k = 0, 1, \dots$, сходится к точке $\tilde{z} = \pi_C(\tilde{x}, \tilde{y})$. Далее, повторяя для точек \tilde{z}, \tilde{x} рассуждения, приведенные при доказательстве теоремы 3.1, получаем требуемые утверждения. Теорема доказана. ■

Правило, по которому каждой точке $x^k, k = 0, 1, \dots$, порождаемой алгоритмом безусловной оптимизации, ставится в соответствие точка $y^k \in C$, можно задавать различным образом. Например, выбирать y^k среди допустимых точек, полученных на предыдущих итерациях, или решать некоторую вспомогательную задачу.

Будем выбирать точки y^k на каждой итерации оптимизационного алгоритма так, чтобы неравенство (3.12) выполнялось при наименьшем значении штрафного коэффициента.

Обозначив, как и ранее, $r(x, y) = \max \left\{ 0, -\frac{f'(x, p(x, y))}{h'(x, p(x, y))} \right\}$, положим

$$r(x^k) = \min \left\{ r(\bar{x}^k, x^k) : \bar{x}^k = \pi_C(x^k, y), y \in C \right\}, \quad (3.13)$$

$$y^k = \arg \min \left\{ r(\bar{x}^k, x^k) : \bar{x}^k = \pi_C(x^k, y), y \in C \right\}, \quad (3.14)$$

$$r^* = \sup \{ r(x) : x \notin C \}. \quad (3.15)$$

Используя для выбора y^k правило (3.14), и выбирая $\lambda > r^*$, получим выполнение неравенств (3.12) на каждой итерации оптимизационного алгоритма. Таким образом, справедлива

Лемма 3.2. Пусть $\lambda > r^*$, тогда $F_\lambda(x)$ – точная штрафная функция.

Заметим, что любое правило выбора точек $y^k \in C, k = 0, 1, \dots$, можно рассматривать, как использование некоторого приближенного решения задачи (3.14).

Пусть $f, h_i : R^n \rightarrow R, i = 1, \dots, m$ – выпуклые собственные функции,

$$f^* = \min \left\{ f(x) : h_i(x) \leq 0, i = 1, \dots, m, x \in R^n \right\}, \quad (3.16)$$

$$h(x) = \max \{ h_i(x) : i = 1, \dots, m \}.$$

Известно (см., например, [155, 184]), что $F_\lambda(x)$ точная штрафная функция для задачи (3.16), если $\lambda > \lambda^* = \sum_{i=1}^m u_i^*$, где (u_1^*, \dots, u_m^*) – оптимальные значения множителей Лагранжа (двойственных переменных) задачи (3.16).

Естественным образом возникает вопрос – как соотносятся величины r^* и λ^* , не превышает ли многократно значение r^* величину λ^* .

Ответ на этот вопрос получен в частном случае, когда задача (3.16) является задачей линейного программирования: найти

$$f^* = \min \langle c, x \rangle, \quad (3.17)$$

$$\langle a_i, x \rangle + b_i \leq 0, i = 1, \dots, m. \quad (3.18)$$

Для этой задачи $f(x) = \langle c, x \rangle$, $h(x) = \max \{h_i(x), i = 1, \dots, m\}$, где $h_i(x) = \langle a_i, x \rangle + b_i, i = 1, \dots, m$.

Теорема 3.4. Пусть задача (3.17), (3.18) имеет единственное оптимальное решение x^* , ограничения задачи, активные в точке x^* , линейно независимы, (u_1^*, \dots, u_m^*) – оптимальные значения двойственных переменных.

Тогда $r^* = \sum_{i=1}^m u_i^*$.

Доказательство. Поскольку $r(x) = \min \{r(\tilde{x}, x) : \tilde{x} = \pi_C(x, y), y \in C\} \leq r(x^*, x)$ при $x \notin C$, имеем $r^* = \sup \{r(x) : x \notin C\} \leq \sup \{r(x^*, x) : x \notin C\}$.

Положим $X^+ = \{x : x \notin C, f(x) \geq f^*\}$. Понятно, что $\{x : x \notin C\} = X^+ \cup \{x : f(x) < f^*\}$.

Пусть $x \in X^+$, тогда $f'(x^*, p(x^*, x)) = \frac{f(x) - f(x^*)}{\|x - x^*\|} \geq 0$ и $r(x^*, x) = 0$.

Отсюда $\sup \{r(x^*, x) : x \notin C\} = \sup \{r(x^*, x) : f(x) < f^*\}$.

Обозначим

I^* множество ограничений задачи (3.17), (3.18), активных в точке x^* ,

$$h^*(x) = \max \{h_i(x) : i \in I^*\},$$

$$r^*(x, y) = \max \left\{ 0, -\frac{f'(x, p(x, y))}{h^{*'}(x, p(x, y))} \right\}.$$

Надграфик функции $\max(0, h^*(x))$ есть многогранный конус с вершиной в точке $(x^*, 0)$. Отсюда следует, что функция $h^*(x)$ линейна на любом луче, выходящем из точки x^* , и $h^{*'}(x^*, p(x^*, x)) = -h^{*'}(x, p(x, x^*))$ для x таких, что $f(x) < f^*$. Для линейной функции $f(x)$ также справедливо

соотношение $f'(x^*, p(x^*, x)) = -f'(x, p(x, x^*))$. Отсюда следует, что $r(x^*, x) = r^*(x^*, x)$ для любого x , $f(x) < f^*$.

Нетрудно видеть, что $r^*(x^*, x) = r^*(x, x^*)$ и для любого $\delta > 0$

$$\begin{aligned} \sup\{r(x^*, x) : f(x) < f^*\} &= \sup\{r^*(x^*, x) : f(x) < f^*, h^*(x) = \delta\} = \\ &= \sup\{r^*(x, x^*) : f(x) < f^*, h^*(x) = \delta\}. \end{aligned}$$

Обозначим

$$\bar{r} = \sup\{r^*(x, x^*) : f(x) < f^*, h^*(x) = \delta\}. \quad (3.19)$$

Таким образом для величины r^* получаем неравенство

$$r^* = \sup\{r(x) : x \notin C\} \leq \bar{r}.$$

Ограничения задачи (3.17), (3.18), активные в оптимальной точке x^* , являются линейно независимыми. Отсюда следует, что множество $\{x : h^*(x) \leq \delta\}$ имеет единственную крайнюю точку \bar{x} , и $h_i(\bar{x}) = \delta, i \in I^*$.

Очевидно, что

$$\langle c, \bar{x} \rangle = \min\{\langle c, x \rangle : h^*(x) \leq \delta\}. \quad (3.20)$$

Покажем, что \bar{x} – оптимальное решение задачи (3.19). Пусть x – допустимая точка задачи (3.19). Обозначим $I(x) = \{i : i \in I^*, h_i(x) = \delta\}$.

Субдифференциал $G_{h^*}(x)$ функции h^* в точке x равен $G_{h^*}(x) = \text{conv}\{a_i, i \in I(x)\}$. Производная функции $h^*(x)$ в точке x по направлению $p(x, x^*) = (x^* - x) / \|x^* - x\|$ равна $h^{*'}(x, p(x, x^*)) = \max_{g \in G_{h^*}(x)} \langle g, p(x, x^*) \rangle$. Положим

$$g(x) = \arg \max_{g \in G_{h^*}(x)} \langle g, p(x, x^*) \rangle, \text{ тогда } h^{*'}(x, p(x, x^*)) = \langle g(x), p(x, x^*) \rangle.$$

Поскольку вектор $g(x)$ можно записать в виде

$$g(x) = \sum_{i \in I(x)} \lambda_i a_i, \lambda_i \geq 0, i \in I(x), \text{ то } \langle g(x), x' \rangle + b' = 0 \text{ для точек } x' \text{ таких, что}$$

$$\langle a_i, x' \rangle + b_i = 0, i \in I(x) \text{ где } b' = \sum_{i \in I(x)} \lambda_i b_i. \text{ Откуда следует}$$

$$\langle g(x), x \rangle = \langle g(x), \bar{x} \rangle. \quad (3.21)$$

Рассмотрим функцию h^* в точке \bar{x} . Очевидно, $g(x) \in G_{h^*}(x) \subseteq G_{h^*}(\bar{x})$,

откуда

$$0 \geq h^*(\bar{x}, p(\bar{x}, x^*)) = \max_{g \in G_{h^*}(\bar{x})} \langle g, p(\bar{x}, x^*) \rangle \geq \langle g(x), p(\bar{x}, x^*) \rangle. \quad (3.22)$$

Учитывая (3.20), (3.21) и (3.22), нетрудно проверить, что

$$\begin{aligned} r^*(x, x^*) - r^*(\bar{x}, x^*) &= \frac{\langle c, x^* - x \rangle}{-h^*(x, p(x, x^*)) \|x^* - x\|} - \frac{\langle c, x^* - \bar{x} \rangle}{-h^*(\bar{x}, p(\bar{x}, x^*)) \|x^* - \bar{x}\|} \leq \\ &\leq \frac{\langle c, x^* - x \rangle}{-\langle g(x), x^* - x \rangle} - \frac{\langle c, x^* - \bar{x} \rangle}{-\langle g(x), x^* - \bar{x} \rangle} = \frac{1}{-\langle g(x), x^* - x \rangle} (\langle c, \bar{x} \rangle - \langle c, x \rangle) \leq 0. \end{aligned}$$

Следовательно, максимум в (3.19) достигается в точке \bar{x} и $\bar{r} = r^*(\bar{x}, x^*)$.

Обозначим A матрицу, строками которой являются векторы $a_i, i \in I^*$.

Тогда $A(x^* - \bar{x}) = -\delta e$, где $e = (1, \dots, 1)$ — n -мерный вектор, $x^* - \bar{x} = -\delta A^{-1}e$.

Для вектора u^* оптимальных значений двойственных переменных задачи

(3.17), (3.18) имеем $A'u^* = -c$, $u^* = -(A')^{-1}c$. Функция $h^*(x)$ линейна на любом

луче, выходящем из точки x^* , и $h^*(x^*) = 0$. Поэтому, как нетрудно видеть,

$$-h^*(\bar{x}, p(\bar{x}, x^*)) \|x^* - \bar{x}\| = \delta. \text{ Откуда}$$

$$\bar{r} = r^*(\bar{x}, x^*) = \frac{\langle c, x^* - \bar{x} \rangle}{-h^*(\bar{x}, p(\bar{x}, x^*)) \|x^* - \bar{x}\|} = \frac{\langle c, -\delta A^{-1}e \rangle}{\delta} = \langle -(A')^{-1}c, e \rangle = \langle u^*, e \rangle.$$

Поскольку $r(\bar{x}) \leq r^* = \sup\{r(x) : x \in C\} \leq r^*(\bar{x}, x^*)$, то для завершения доказательства достаточно показать, что

$$r^*(\bar{x}, x^*) = r(\bar{x}) = \min \{r(z, \bar{x}) : z = \pi_C(\bar{x}, y), y \in C\}.$$

Пусть $y \in C$, $z = \pi_C(\bar{x}, y)$, $J(z) = \{i : h_i(z) = 0, i = 1, \dots, m\}$ – множество активных ограничений в точке z , $G_h(z) = \text{conv}\{a_i, i \in J(z)\}$ – субдифференциал функции h в точке z . Можно показать, что при достаточно малом значении величины δ будет выполняться $J(z) \subseteq I^*$.

Обозначим $\bar{g}(z) = \arg \max_{g \in G_h(z)} \langle g, p(z, \bar{x}) \rangle$, тогда

$$h'(z, p(z, \bar{x})) = \max_{g \in G_h(\bar{x})} \langle g, p(z, \bar{x}) \rangle = \langle \bar{g}(z), p(z, \bar{x}) \rangle \text{ и}$$

$$h'(x^*, p(x^*, \bar{x})) = \max_{g \in G_h(x^*)} \langle g, p(x^*, \bar{x}) \rangle \geq \langle \bar{g}(z), p(\bar{x}, x^*) \rangle.$$

Как и ранее, можно показать, что $\langle \bar{g}(z), x^* \rangle = \langle \bar{g}(z), z \rangle$, откуда

$$\begin{aligned} r(x^*, \bar{x}) - r(z, \bar{x}) &= \frac{-\langle c, \bar{x} - x^* \rangle}{h'(x^*, p(x^*, \bar{x})) \|x^* - \bar{x}\|} - \frac{-\langle c, \bar{x} - z \rangle}{h'(z, p(z, \bar{x})) \|y' - \bar{x}\|} \leq \\ &\leq \frac{-\langle c, \bar{x} - x^* \rangle}{\langle \bar{g}(z), p(\bar{x}, x^*) \rangle \|x^* - \bar{x}\|} - \frac{-\langle c, \bar{x} - z \rangle}{\langle \bar{g}(z), p(z, \bar{x}) \rangle \|z - \bar{x}\|} = \frac{-\langle c, \bar{x} - x^* \rangle}{\langle \bar{g}(z), \bar{x} - x^* \rangle} - \frac{-\langle c, \bar{x} - z \rangle}{\langle \bar{g}(z), \bar{x} - z \rangle} = \\ &= \frac{1}{\langle \bar{g}(z), \bar{x} - z \rangle} \left(\langle c, x^* - \bar{x} \rangle - \langle c, z - \bar{x} \rangle \right) \leq 0. \end{aligned}$$

То есть $x^* = \arg \min \{r(z, \bar{x}) : z = \pi_C(\bar{x}, y), y \in C\}$ и $r^* = r(x^*, \bar{x}) = \langle u^*, c \rangle$.

Теорема доказана. ■

Предложенный подход позволяет строить достаточно эффективные процедуры автоматического определения значений штрафных коэффициентов по ходу работы алгоритмов безусловной оптимизации. При этом каждой точке x^k , генерируемой на итерации k алгоритма, должна ставиться в соответствие некоторая точка y^k из допустимого множества C исходной задачи. Для точек x^k и y^k выполняется проверка условия (3.6). Простейшая процедура заключается в использовании начальной точки y^0 ,

такой что $h(y^0) < 0$, на всех итерациях алгоритма, т.е. $y^k = y^0$.

Показано, что наилучший выбор точек y^k при решении задачи линейного программирования приводит к известному результату – для штрафного

коэффициента r^* выполняется $r^* = \sum_{i=1}^m u_i^*$, где u_i^* – оптимальные значения

двойственных переменных. Для повышения эффективности предлагаемого подхода необходимо разрабатывать правила (желательно простые) формирования допустимых точек y^k .

3.2. Точные штрафные функции и невыпуклые задачи оптимизации

Для невыпуклых задач оптимизации использование точных штрафных функций связано с существенными проблемами. Допустимая область исходной задачи может быть многосвязной, штрафная добавка и штрафная функция в целом могут иметь локальные минимумы вне допустимой области исходной задачи. В связи с этим известные методы не всегда могут гарантировать сходимость к допустимому решению. Условия, при которых обеспечивается сходимость методов к допустимой точке, трудно проверять при решении практических задач, часто эти задачи не удовлетворяют сформулированным условиям сходимости. Такие патологические задачи могут быть весьма простыми. Достаточно рассмотреть задачи с квадратичными невыпуклыми ограничениями.

В данном подразделе предлагается подход, позволяющий преодолеть некоторые из указанных проблем. При этом должна быть известна допустимая точка исходной задачи. Схема решения в целом приводится для гладких задач, и может интерпретироваться как последовательное решение совокупности регуляризованных подзадач.

3.2.1. Постановка задачи и некоторые свойства штрафных функций

Рассматривается задача: найти

$$f^* = \min \{f(x) : x \in \Omega\}, \quad (3.23)$$

где $\Omega = \{x : h_i(x) \leq 0, i = 1, \dots, m, x \in R^n\}$ $f, h_i : R^n \rightarrow R$ – непрерывные, дифференцируемые по направлениям (в смысле Адамара [77]) функции, $i = 1, \dots, m$.

Функция f называется дифференцируемой в точке x по направлению p в смысле Адамара, если существует конечный предел [77]

$$f'_H(x, p) = \lim_{[\alpha, p'] \rightarrow [+0, p]} \frac{f(x + \alpha p') - f(x)}{\alpha}, \quad (3.24)$$

величина $f'_H(x, p)$ – производная Адамара функции f в точке x по направлению p . Если предел в (3.24) существует и конечен для любого $p \in R^n$, то функция f называется H -дифференцируемой в точке x по направлениям.

В дальнейшем, если это не будет вызывать неоднозначностей, вместо $f'_H(x, p)$ будем использовать обозначение $f'(x, p)$ и называть производную Адамара просто производной функции f в точке x по направлению p .

Положим $h(x) = \max \{0, h_i(x) : i = 1, \dots, m\}$. Будем рассматривать штрафные функции вида

$$F_\lambda(x) = f(x) + \lambda \cdot h(x), \quad \lambda \in R, \lambda \geq 0, \quad (3.25)$$

и задачи: найти

$$F_\lambda^* = \min \{F_\lambda(x) : x \in \Omega_\delta\}, \quad (3.26)$$

где $\Omega_\delta = \{x : h(x) \leq \delta, x \in R^n\}$, $\delta > 0$.

Обозначим $F'_\lambda(x, p)$, $f'(x, p)$, $h'(x, p)$ – производные функций F_λ , f , h в точке $x \in R^n$ по направлению p . Следуя [77], определим $h^\downarrow(x)$, скорость наискорейшего спуска функции h в точке x ,

$$h^\downarrow(x) = \min \{h'(x, p) : \|p\| = 1\}.$$

Пусть для чисел $\delta > 0$ и $a > 0$ выполняются условия

$$h^\downarrow(x) \leq -a, \quad \forall x \in \Omega_\delta \setminus \Omega, \quad (3.27)$$

а функция f липшицева на $\Omega_\delta \setminus \Omega$. Тогда существует число $\lambda^* \geq 0$ такое, что при $\lambda > \lambda^*$ все точки локального минимума функции $F_\lambda(x)$, принадлежащие множеству Ω_δ , являются и точками локального минимума функции f на Ω (подробнее см. в [77]). Вне множества Ω_δ штрафная добавка и, соответственно, штрафная функция $F_\lambda(x)$ могут иметь дополнительные локальные минимумы, не принадлежащие множеству Ω .

В случае, когда $f, h_i : R^n \rightarrow R, i = 1, \dots, m$ – выпуклые функции и для задачи (3.23) выполняются условия Слейтера, величину $a > 0$ можно выбрать так, что условие (3.27) будет выполняться при любом $\delta > 0$. То есть при $\lambda > \lambda^*$ задача (3.26) есть задача безусловной оптимизации ($\Omega_\delta = R^n$), точки минимума в задачах (3.23) и (3.26) совпадают, для решения задачи (3.26) могут использоваться эффективные методы безусловной оптимизации.

В невыпуклом случае положение существенно усложняется. Тем не менее, точные штрафные функции широко используются в методах линеаризации [155, 75], в методах последовательного квадратичного программирования [70, 16, 17]. Одна из основных проблем при этом – определение значений штрафных коэффициентов. Для этого используются решения специальных вспомогательных задач, а также различные эвристики. Целью большинства работ в этом направлении является достижение сверхлинейной скорости сходимости для тех или других классов задач. Методы сходятся к стационарным точкам штрафной функции в допустимой

области Ω или в недопустимой области. Вопросы о том, как избежать сходимости к стационарной точке штрафной функции в недопустимой области, как правило, не рассматриваются.

В настоящем подразделе рассматривается подход, обеспечивающий сходимость к допустимой стационарной точке исходной задачи, и позволяющий сравнительно просто определять значения штрафных коэффициентов. Предлагаемая схема сводится к введению некоторым образом специальной доверительной области, радиус которой определяется автоматически по ходу работы оптимизационного алгоритма.

Процедура определения штрафных коэффициентов является обобщением результатов, изложенных в предыдущем разделе для выпуклых задач оптимизации.

3.2.2. Дополнительные свойства и доверительные области

Лемма 3.3. Пусть $\Phi: R^n \rightarrow R$ – непрерывная, дифференцируемая по направлениям функция, $x^k, p^k \in R^n, \|p^k\| = 1, k = 1, 2, \dots$, – сходящаяся последовательности, $x^k \rightarrow \tilde{x}, p^k \rightarrow \tilde{p}, k \rightarrow \infty$, для некоторых чисел $\delta > 0$, $\varepsilon > 0$ выполняются условия

$$\Phi'(x^k + tp^k, p^k) \leq -\varepsilon, t \in [0, \delta], k = 1, 2, \dots \quad (3.28)$$

Тогда $\Phi'(\tilde{x}, \tilde{p}) \leq -\varepsilon$.

Доказательство. Покажем, что для любого $\sigma > 0$ выполняется неравенство

$$\Phi(x^k + tp^k) \leq \Phi(x^k) - \varepsilon \cdot t + \sigma \cdot t, t \in [0, \delta]. \quad (3.29)$$

Для краткости обозначим $\varphi(t) = \Phi(x^k + tp^k)$, $\varphi'(t) = \Phi'(x^k + tp^k, p^k)$. По определению производной по направлению имеем $\varphi(t + \Delta t) = \varphi(t) + \varphi'(t)\Delta t + o(t, \Delta t)$, $\Delta t \geq 0$, где

$$\frac{o(t, \Delta t)}{\Delta t} \rightarrow 0 \text{ при } \Delta t \rightarrow +0. \quad (3.30)$$

Пусть значение переменной t зафиксировано, $t \in [0, \delta]$.

Разобьем интервал $[0, t]$ на совокупность подинтервалов

$[t_0, t_1], [t_1, t_2], \dots, [t_{S-1}, t_S]$, где $0 = t_0 < t_1 < \dots < t_{S-1} < t_S = t$. Обозначим

$\Delta_s = t_{s+1} - t_s$, $s = 0, 1, \dots, S-1$. Очевидно,

$\varphi(t_{s+1}) = \varphi(t_s + \Delta_s) = \varphi(t_s) + \varphi'(t_s)\Delta_s + o(t_s, \Delta_s)$ и

$$\varphi(t) = \varphi(t_S) = \varphi(t_0) + \sum_{s=0}^{S-1} \varphi'(t_s)\Delta_s + \sum_{s=0}^{S-1} o(t_s, \Delta_s).$$

В силу (3.30) всегда можно выбрать такое разбиение интервала $[0, t]$, что для каждого подинтервала $[t_s, t_{s+1}]$, будет выполняться $|o(t_s, \Delta_s)| \leq \sigma \cdot \Delta_s$, $s = 0, \dots, S-1$, Откуда, учитывая (3.28), получаем

$$\varphi(t) \leq \varphi(t_0) - \varepsilon \sum_{s=0}^{S-1} \Delta_s + \sigma \sum_{s=0}^{S-1} \Delta_s = \varphi(t_0) - \varepsilon \cdot t + \sigma \cdot t, \text{ что эквивалентно (3.29).}$$

Покажем, что $\Phi'(\tilde{x}, \tilde{p}) \leq -\varepsilon$. Предположим противное, т.е. $\Phi'(\tilde{x}, \tilde{p}) > -\varepsilon$. Выберем число $\gamma > 0$ так, что $\Phi'(\tilde{x}, \tilde{p}) = -\varepsilon + \gamma$. В силу дифференцируемости по направлению для $t \geq 0$ имеет место разложение $\Phi(\tilde{x} + t\tilde{p}) = \Phi(\tilde{x}) + \Phi'(\tilde{x}, \tilde{p}) \cdot t + o(t) = \Phi(\tilde{x}) - \varepsilon \cdot t + \gamma \cdot t + o(t)$. Очевидно, что всегда можно выбрать $\tau \in [0, \delta]$ так, что $\gamma \cdot \tau + o(\tau) > 0$. Обозначим $\rho = \gamma \cdot \tau + o(\tau)$. Тогда

$$\Phi(\tilde{x} + \tau \cdot \tilde{p}) = \Phi(\tilde{x}) - \varepsilon \cdot \tau + \rho, \rho > 0 \quad (3.31)$$

В силу непрерывности Φ существует K , такое что $|\Phi(\tilde{x} + \tau \cdot \tilde{p}) - \Phi(x^k + \tau \cdot p^k)| \leq \frac{\rho}{4}$, $\forall k > K$, т.е. $\Phi(\tilde{x} + \tau \cdot \tilde{p}) \leq \Phi(x^k + \tau \cdot p^k) + \frac{\rho}{4}$.

Выберем σ так, что $\sigma \cdot \tau = \frac{\rho}{4}$. Тогда, учитывая (3.29), получаем

$\Phi(\tilde{x} + \tau \cdot \tilde{p}) \leq \Phi(x^k) - \varepsilon \cdot \tau + \frac{\rho}{2}$, $\forall k > K$. Также, очевидно, существует $K_1 > K$,

такое что $|\Phi(\tilde{x}) - \Phi(x^k)| \leq \frac{\rho}{4}$, $\forall k > K_1$, т.е. $\Phi(x^k) \leq \Phi(\tilde{x}) + \frac{\rho}{4}$. Откуда

$\Phi(\tilde{x} + \tau \cdot \tilde{p}) \leq \Phi(\tilde{x}) - \varepsilon \cdot \tau + \frac{3\rho}{4}$, что противоречит (3.31). Таким

образом, предположение $\Phi'(\tilde{x}, \tilde{p}) > -\varepsilon$ неверно. Лемма доказана. ■

Обозначим $p(x, y)$ вектор, определяющий направление из точки x в точку y , $y \neq x$,

$$p(x, y) = (y - x) / \|y - x\|. \quad (3.32)$$

Положим $T_\delta(x, y) = [x, x + \delta p(x, y)] \cap [x, y]$, где $\delta > 0$. Множество $T_\delta(x, y)$ есть δ -окрестность точки x на отрезке $[x, y]$. Очевидно, что $T_\delta(x, y) = [x, y]$, если $\delta = +\infty$.

Теорема 3.5. Пусть $\Phi: R^n \rightarrow R$ – непрерывная, дифференцируемая по направлениям функция; \tilde{x} – локальный минимум функции Φ на R^n ; Ω_B – ограниченное замкнутое множество, $\Omega_B \subset R^n$, $\text{int } \Omega_B \neq \emptyset$; заданы точка $y^0 \in \text{int } \Omega_B$ и последовательность $x^k \in R^n$, $k = 1, 2, \dots$, сходящаяся к \tilde{x} , для некоторых чисел $\delta > 0$, $\varepsilon > 0$ выполняются условия

$$\Phi'(x, p(x^k, y^0)) \leq -\varepsilon, \quad \forall x \in T_\delta(x^k, y^0) \setminus \Omega_B, \quad (3.33)$$

Тогда для предельной точки \tilde{x} имеет место $\tilde{x} \in \Omega_B$.

Доказательство. Предположим $\tilde{x} \notin \Omega_B$. Без ограничения общности можно считать, что $x^k \notin \Omega_B$, $k, k = 0, 1, \dots$. Обозначим $\tilde{p} = p(\tilde{x}, y^0)$ (заметим, что $\tilde{p} = \lim_{k \rightarrow \infty} p^k$),

$$\tilde{t} = \min \{t : t > 0, \tilde{x} + t\tilde{p} \in \Omega_B\}. \quad (3.34)$$

Точка $\tilde{x} + \tilde{t} \cdot \tilde{p}$ лежит на границе множества Ω_B . Положим $\tilde{\delta} = \min(\delta, \tilde{t})$.

Покажем, что существует номер K такой, что $x^k + tp^k \notin \Omega_B$, при $t \in \left[0, \frac{\tilde{\delta}}{2}\right]$, $k > K$. Предположим противное, т.е. для любого i найдется $k_i > i$,

для которого существует точка $t_{k_i} \in \left[0, \frac{\tilde{\delta}}{2}\right]$ такая, что $x^{k_i} + t_{k_i} p^{k_i} \in \Omega_B$. Без

ограничения общности можно считать, что последовательность

$t_{k_i}, i=1, \dots$, сходится. Обозначим $\bar{t} = \lim_{i \rightarrow \infty} t_{k_i}$. Имеем $\bar{t} \leq \frac{\tilde{\delta}}{2} < \tilde{t}$, $\lim_{i \rightarrow \infty} p^{k_i} = \tilde{p}$,

$x^{k_i} + t_{k_i} p^{k_i} \rightarrow \tilde{x} + \bar{t} \cdot \tilde{p}, i \rightarrow \infty$. Откуда, в силу замкнутости множества Ω_B , имеем $\tilde{x} + \bar{t} \cdot \tilde{p} \in \Omega_B$, что противоречит (3.34).

Таким образом, имеет место $\Phi'(x^k + tp^k, p^k) \leq -\varepsilon$, $t \in \left[0, \frac{\tilde{\delta}}{2}\right]$, $k > K$.

Отсюда, в силу леммы 3.3, следует $\Phi'(\tilde{x}, \tilde{p}) \leq -\varepsilon$, что противоречит утверждению о том, что \tilde{x} локальный минимум функции Φ . То есть предположение $\tilde{x} \notin \Omega_B$ неверно. Теорема доказана. ■

В настоящее время для построения устойчивых алгоритмов, обеспечения глобальной сходимости широко применяется понятие доверительной области (trust region, см., например, [16, 17, 70]). Будем использовать аналогичный подход для обеспечения сходимости к допустимой точке исходной задачи.

Предполагается заданной точка $y^0 \in \Omega$. Обозначим $B_\rho(y^0) = \{x : \|x - y^0\| \leq \rho, x \in R^n\}$. Очевидно, что если выполняются условия (3.27), то существует $\rho > 0$ такое, что

$$h^\downarrow(x) \leq -a, \quad \forall x \in B_\rho(y^0) \setminus \Omega. \quad (3.35)$$

Положим $\eta_\rho(x) = \max\{0, \|x - y^0\| - \rho\}$. Рассмотрим штрафную функцию

$$\Phi_{\lambda\beta\rho}(x) = f(x) + \lambda \cdot h(x) + \beta \cdot \eta_\rho(x) \quad (3.36)$$

и задачи: найти

$$\Phi_{\lambda\beta\rho}^* = \min\{\Phi_{\lambda\beta\rho}(x) : x \in R^n\}, \quad (3.37)$$

$$f_\rho^* = \min\{f(x) : x \in B_\rho(y^0) \cap \Omega\}. \quad (3.38)$$

Утверждение 1. Пусть для множества $B_\rho(y^0)$ выполняется условие (3.35), f, h – липшицевы функции на R^n . Тогда существуют число $\lambda^* > 0$ и

функция $\beta(\lambda)$, $\beta(\lambda) > 0$, такие, что при $\lambda \geq \lambda^*$ и $\beta \geq \beta(\lambda)$ все точки локального минимума функции $\Phi_{\lambda\beta\rho}(x)$ в пространстве R^n являются точками локального минимума функции f на $\Omega \cap B_\rho(y^0)$. То есть задачи (3.37) и (3.38) эквивалентны.

В самом деле, $\Phi_{\lambda\beta\rho}(x) = f(x) + \lambda \cdot h(x)$ для $x \in B_\rho(y^0)$. В силу условия (3.35) существует $\lambda^* > 0$ такое, что при $\lambda \geq \lambda^*$ все точки локального минимума функции $\Phi_{\lambda\beta\rho}(x)$ на множестве $B_\rho(y^0)$ являются точками локального минимума функции f на $\Omega \cap B_\rho(y^0)$. Функция $f(x) + \lambda \cdot h(x)$ – липшицева на R^n , а для $\eta_\rho(x)$ выполняется $\eta_\rho^\downarrow(x) = -1, \forall x \notin B_\rho(y^0)$, откуда следует существование числа $\beta(\lambda) > 0$ такого, что все точки локального минимума функции $\Phi_{\lambda\beta\rho}(x)$ в пространстве R^n являются точками локального минимума функции $f(x) + \lambda \cdot h(x)$ на множестве $B_\rho(y^0)$, и, как было показано, точками локального минимума функции f на $\Omega \cap B_\rho(y^0)$.

Теорема 3.6. Пусть заданы числа $\rho > 0$, $\sigma > 0$, функции f, h липшицевы на R^n и выполняются условия

$$h'(x, p(x, y^0)) \leq -\sigma, \forall x \in B_\rho(y^0) \setminus \Omega, \quad (3.39)$$

Тогда для любых $\delta > 0$, $\varepsilon > 0$ существуют числа $\lambda(\varepsilon) > 0$ и функция $\beta(\lambda) > 0$, такие что при $\lambda > \lambda(\varepsilon)$ и $\beta > \beta(\lambda)$ для любой последовательности точек x^k , $k, k = 0, 1, \dots$, сходящейся к локальному минимуму \tilde{x} задачи (3.37), выполняются условия теоремы 3.5, где $\Phi(x) = \Phi_{\lambda\beta\rho}(x)$, $\Omega_B = B_\rho(y^0) \cap \Omega$. То есть \tilde{x} – решение задачи (3.38).

Доказательство. Обозначим L_f константу Липшица для функции f .

Очевидно, что $f'(x, p) \leq L_f$. Положим $\lambda(\varepsilon) = \frac{L_f + \varepsilon}{\sigma}$, тогда для любой точки

$x \in B_\rho(y^0) \setminus \Omega$ при $\lambda > \lambda(\varepsilon)$ выполняется $\Phi'_{\lambda\beta\rho}(x, p) \leq -\varepsilon$.

Пусть $x \notin B_\rho(y^0)$. Очевидно $\eta'_\rho(x, p) = -1$. Обозначим L_h константу Липшица для функции h , $h'(x, p) \leq L_h$. Пусть λ удовлетворяет условию $\lambda > \lambda(\varepsilon)$. Положим $\beta(\lambda) = L_f + \lambda L_h + \varepsilon$. Тогда для любой точки $x \notin B_\rho(y^0)$ при $\beta > \beta(\lambda)$ имеет место $\Phi'_{\lambda\beta\rho}(x, p) \leq -\varepsilon$. Таким образом, условия (3.33) выполняются для любой точки $x \notin \Omega_B = \Omega \cap B_\rho(y^0)$. Теорема доказана.

■

Пусть для решения задачи (3.37) используется некоторый алгоритм A безусловной минимизации, обеспечивающий сходимость к локальному минимуму функции $\Phi_{\lambda\beta\rho}(x)$. Теоремы 3.5, 3.6 позволяют сформировать модификацию алгоритма A (алгоритм A_B) для решения задачи (3.37), которая для заданных $\delta > 0$, $\varepsilon > 0$, $\sigma > 0$ и базовой точки $y^0 \in \text{int}\Omega$ определяет радиус ρ шара $B_\rho(y^0)$, удовлетворяющий условиям теорем, и находит точку локального минимума функции f на $\Omega \cap B_\rho(y^0)$.

В алгоритме A_B на каждой итерации k алгоритма A выполняются следующие дополнительные операции.

1) Для текущей точки x_k проверяется условие

$$h'(x, p(x^k, y^0)) \leq -\sigma, \quad \forall x \in T_\delta(x^k, y^0) \cap \{B_\rho(y^0) \setminus \Omega\}, \quad (3.40)$$

если это условие не выполняется, радиус ρ уменьшается;

2) Для текущей точки x_k проверяется условие (3.33) в два этапа:

этап 1 – для $\forall x \in T_\delta(x^k, y^0) \cap \{B_\rho(y^0) \setminus \Omega\}$, если условие (3.33) не выполняется, увеличивается коэффициент λ ;

этап 2 – для $\forall x \in T_\delta(x^k, y^0) \setminus B_\rho(y^0)$, если условие (3.33) не выполняется, увеличивается коэффициент β .

Под увеличением (уменьшением) параметра понимается такое его изменение, после которого соответствующее условие выполняется. При этом

изменение должно быть не меньше некоторой фиксированной величины. В силу конечности чисел $\lambda(\varepsilon)$, $\beta(\lambda)$ число изменений параметров λ , β , ρ будет конечно, после чего алгоритм сходится к локальному минимуму \tilde{x} функции $\Phi_{\lambda\beta\rho}(x)$ в пространстве R^n и $\tilde{x} \in \Omega \cap B_\rho(y^0)$.

В приведенной схеме алгоритма A_B необходимо конкретизировать процедуры проверки условий (3.33), (3.40). Нетрудно видеть, что каждое множество, на котором эти условия проверяются, имеет вид $X = \{x : x = x^k + tp(x^k, y^0), 0 \leq t \leq \bar{t}\}$ либо $X = \{x : x = x^k + tp(x^k, y^0), 0 \leq t < \bar{t}\}$. Во втором случае предельная точка $x = x^k + \bar{t} \cdot p(x^k, y^0)$ не принадлежит множеству X . Проверка заключается в анализе поведения функции $\varphi(t) = \Phi_{\lambda\beta\rho}(x^k + t \cdot p(x^k, y^0))$ или $\eta(t) = h(x^k + t \cdot p(x^k, y^0))$ в интервале $0 \leq t < \bar{t}$. Обозначим $\varphi'_+(t)$ производную функции $\varphi(t)$ в точке t в направлении увеличения переменной t , $\varphi'_-(t)$ – производная в направлении уменьшения переменной t . Аналогично определим производные $\eta'_+(t)$, $\eta'_-(t)$.

Рассмотрим случай, когда функции $f, h_i, i = 1, \dots, m$, выпуклые. Тогда $\varphi(t)$ и $\eta(t)$ тоже выпуклые функции. Нетрудно видеть, что в силу выпуклости имеем $\varphi'_+(t_1) \leq \varphi'_+(t_2) \leq -\varphi'_-(\bar{t})$ для любых $t_1, t_2 \in [0, \bar{t}]$, $t_1 < t_2$. Откуда следует, что проверку условий (3.33), (3.40) достаточно проводить только в точке \bar{t} .

Рассмотрим случай, когда функции $f, h_i, i = 1, \dots, m$, дифференцируемые, величина δ достаточно мала. Тогда в δ -окрестности точки x^k эти функции достаточно хорошо приближаются их линейной аппроксимацией в этой точке, а функции $\varphi(t)$ и $\eta(t)$ для линейных аппроксимаций будут выпуклыми. Таким образом, проверку условий (3.33), (3.40) также достаточно проводить только в точке \bar{t} (для \bar{t} всегда

выполняется $\bar{t} \leq \delta$). В случае больших δ проверку достаточно проводить в отдельных точках интервала $[0, \bar{t}]$ с некоторым небольшим шагом.

В приведенном алгоритме A_B выбор параметра $\delta > 0$ достаточно произволен. При больших значениях этого параметра в выпуклом случае достаточно проверять условие (3.33) в единственной точке, являющейся пересечением отрезка $[x^k, y^0]$ с границей множества Ω . При выборе $\delta = 0$ предельная точка \tilde{x} может не принадлежать допустимому множеству исходной задачи. В общем случае проверка условий (3.33), (3.40) является достаточно сложной проблемой. Эффективные процедуры такой проверки могут быть сформулированы для специальных случаев.

Выбор больших значений параметра $\sigma > 0$ приводит к уменьшению радиуса шара ρ . При малых значениях параметра σ по ходу работы оптимизационного алгоритма могут получаться большие значения штрафных коэффициентов.

3.2.3. Общая схема решения задачи

Штрафная функция (3.36) – негладкая. Поэтому в общем случае для непосредственного решения задачи (3.37) должны использоваться методы негладкой оптимизации, например, r -алгоритм [184]. В случае, когда функции $f, h_i, i = 1, \dots, m$ дифференцируемы, естественным является применение методов гладкой оптимизации, в частности, методов линеаризации [155, 75], методов последовательного квадратичного программирования (см., например, [70, 16, 17]).

Предлагаемая схема решения задачи (3.23) в целом заключается в последовательности этапов $s = 0, 1, \dots$, на каждом из которых приближенно решается вспомогательная задача вида (3.37) с использованием алгоритма A_B

для заданной допустимой точки y_s (полагается $y^0 = y_s$) и формируется допустимая точка y_{s+1} для следующего этапа.

В случае, когда функции $f, h_i, i=1, \dots, m$ дифференцируемы, для формирования точки y_{s+1} могут использоваться различные процедуры. Приведем одну из них. Пусть $\tilde{x} \in \Omega$ – приближенное решение задачи (3.37), полученное на этапе s , задано число $\gamma > 0$, $I_\gamma(\tilde{x}) = \{i : h_i(\tilde{x}) \geq -\gamma\}$ – множество γ -активных ограничений. Положим $x = \tilde{x} + v$, представим линеаризованные γ -активные ограничения в виде системы линейных ограничений $Av \leq b$, и рассмотрим задачу поиска точки наиболее удаленной от всех граней многогранника $\{v : Av \leq b, cv \geq d\}$, где неравенство $cv \geq d$ соответствует линеаризованному неравенству $f(\tilde{x} + v) \geq f(\tilde{x})$. Эту задачу можно представить в виде

$$\max z \quad (3.41)$$

при ограничениях

$$Av + e \cdot z \leq b, \quad (3.42)$$

$$cv - z \geq d, \quad (3.43)$$

$$\|v\| \leq 1, \quad (3.44)$$

где $e = (1, \dots, 1)^T$, $z \in R$. Здесь ограничения (3.44) добавлены в задачу, поскольку многогранник $\{v : Av \leq b, cv \geq c^0\}$ может быть неограниченным.

Вместо (3.44) могут использоваться ограничения

$$|v^i| \leq 1, i = 1, 2, \dots, n. \quad (3.45)$$

Пусть v^*, z^* – оптимальное решение задачи (3.41)–(3.44). Рассмотрим $R = \{x : x = \tilde{x} + tv^*, t > 0\}$ – луч, выходящий из точки \tilde{x} . Обозначим \bar{t} минимальное значение параметра t , при котором луч R пересекает границу множества Ω . Для формирования точки y_{s+1} может применяться следующее правило

$$y_{s+1} = \tilde{x} + \frac{\bar{t}}{2} v^*. \quad (3.46)$$

Предлагаемая схема может интерпретироваться как одновременное использование методов внешних штрафов и внутренних точек.

Рассмотренный подход позволяет для методов оптимизации, основанным на использовании точных штрафных функций, преодолеть проблему сходимости к стационарным точкам, не принадлежащим допустимой области исходной невыпуклой задачи. Также сравнительно просто решаются вопросы выбора значений штрафных коэффициентов. Рассмотренный подход применим, если известна начальная допустимая точка исходной задачи. В предлагаемой схеме используется специальная доверительная область, радиус которой определяется автоматически по ходу работы оптимизационного алгоритма. Решение исходной задачи сводится к последовательному решению совокупности регуляризованных подзадач.

3.3. Обобщенный подход к определению штрафных коэффициентов

Рассматривается задача: найти

$$f^* = \min \{ f_0(x) : x \in C \}, \quad (3.47)$$

где $C = \{ x : f_i(x) \leq 0, i = 1, \dots, m, x \in R^n \}$, $f_i : R^n \rightarrow R, i = 0, \dots, m$ – выпуклые функции, принимающие конечные значения при любых значениях переменных.

Положим $\Phi_\beta(x) = f_0(x) + \sum_{i=1}^m \beta_i f_i^+(x)$, $F_\lambda(x) = f_0(x) + \lambda h^+(x)$, где

$$h(x) = \max \{ f_i(x), i = 1, \dots, m \}, \quad f^+(x) = \max \{ 0, f(x) \},$$

$$\Phi_\beta^* = \min \{ \Phi_\beta(x) : x \in R^n \}, \quad (3.48)$$

$$F_\lambda^* = \min \{ F_\lambda(x) : x \in R^n \}. \quad (3.49)$$

$\Phi_\beta(x)$ ($F_\lambda(x)$) называется точной штрафной функцией, если решения задач (3.47) и (3.48) (соответственно (3.49)) совпадают.

Известно (см., например, [184], теоремы 26, 27), что если $u_i^*, i = 1, \dots, m$ – оптимальные значения множителей Лагранжа задачи (3.47), то $\Phi_\beta(x)$ – точная штрафная функция при $\beta_i > u_i^*, i = 1, \dots, m$, $F_\lambda(x)$ – точная штрафная функция при $\lambda > \sum_{i=1}^m u_i^*$.

Обычно для определения множителей Лагранжа на каждой итерации решаются некоторые вспомогательные задачи (например, линеаризация исходной задачи) [184, 17, 75].

В предлагаемом ниже подходе также формулируется вспомогательная задача, однако для ее решения используются эвристические алгоритмы.

Сформулируем в виде леммы следующее простое утверждение.

Лемма 3.4. Пусть множество C ограничено и замкнуто, значения штрафных коэффициентов фиксированы, заданы число $\varepsilon > 0$ и последовательность точек $x_k, k = 1, 2, \dots$, сходящаяся к решению \tilde{x} задачи (3.48) (или решению задачи (3.49)). Пусть каждой x_k по некоторому правилу P поставлена в соответствие точка $z_k = P(x_k), z_k \in C, k = 1, 2, \dots$, и выполняются неравенства

$$\varphi(x_k) \geq f_0(z_k) + \varepsilon \|z_k - x_k\|, \text{ для точек } x_k \notin C, k = 1, 2, \dots, \quad (3.50)$$

где $\varphi(x) = \Phi_\beta(x)$ (соответственно $\varphi(x) = F_\lambda(x)$). Тогда $\tilde{x} \in C$.

Доказательство очевидно, поскольку без ограничения общности можно считать, что для последовательности $z_k, k = 1, 2, \dots$, существует предельная точка $z^* \in C$ и если $\tilde{x} \notin C$, то $f_0(z^*) = \varphi(z^*) < \varphi(\tilde{x})$, т.е. \tilde{x} не есть решение задачи (3.48).

3.3.1. Использование функции F_λ . Пусть заданы $x \notin C$, правило $P: R^n \rightarrow C$. Обозначим

$$\lambda_P(x, \varepsilon) = \max \left(0, \frac{f_0(z) + \varepsilon \|z - x\| - f_0(x)}{h^+(x)} \right), \text{ где } z = P(x), \quad (3.51)$$

$$\lambda_P(\varepsilon) = \sup \{ \lambda_P(x, \varepsilon) : x \in C \} \quad (3.52)$$

Пусть для решения задачи (3.49) применяется некоторый сходящийся алгоритм A . Поскольку точные штрафные коэффициенты заранее не известны, их значения будем уточнять (увеличивать) по ходу работы алгоритма. Обозначим λ_k значение коэффициента λ на итерации k . Для $k=1$ значение $\lambda_1 > 0$ считается заданным. При построении алгоритмом A точки x_k на итерации k используется значение λ_k . Если в точке x_k неравенство (3.50) выполняется при $\lambda = \lambda_k$, полагаем $\lambda_{k+1} = \lambda_k$, в противном случае

$\lambda_{k+1} = \lambda_P(x_k, \varepsilon) + R$, где $R > 0$ – заданный фиксированный параметр.

Очевидно, что соотношения (3.50) выполняются, если $\lambda > \lambda_P(\varepsilon)$ и число уточнений коэффициентов λ_k будет конечным, если $\lambda_P(\varepsilon) < \infty$.

Использование штрафных функций с существенно завышенными значениями штрафных коэффициентов приводит к проблемам, связанным с ошибками округления, ухудшением сходимости оптимизационных алгоритмов. Поэтому важной характеристикой правила P является величина $\lambda_P(\varepsilon)$. В дальнейшем будут рассматриваться различные подходы к формированию правил P .

Представляет интерес оценка величина $\lambda_P(\varepsilon)$, которая может быть достигнута при выборе подходящего правила P . Рассмотрим случай, когда исходная задача (3.47) является задачей линейного программирования:

$$f^* = \min \langle c, x \rangle, \quad (3.53)$$

$$\langle a_i, x \rangle + b_i \leq 0, \quad i = 1, \dots, m. \quad (3.54)$$

Лемма 3.5. Пусть задача (3.53)–(3.54) имеет единственное оптимальное решение x^* , $P = P^* : R^n \rightarrow x^*$, ограничения, активные в точке

x^* , линейно независимы, (u_1^*, \dots, u_m^*) – оптимальные значения

двойственных переменных задачи (3.53)–(3.54). Тогда $\lambda_P(0) = \sum_{i=1}^m u_i^*$.

Доказательство. Нетрудно проверить, что из выпуклости функций f_0, h следует, что $\lambda_P(x, 0)$ не возрастает вдоль любого направления, исходящего из точки x^* и лежащего вне допустимой области. Отсюда следует, что для любого $\bar{\delta} > 0$ имеет место

$$\begin{aligned} \lambda_P(0) &= \sup \{ \lambda_P(x, 0) : 0 < h(x) \leq \bar{\delta} \} = \max_{0 \leq \delta \leq \bar{\delta}} \max \{ \lambda_P(x, 0) : h(x) = \delta \} = \\ &= \max_{0 \leq \delta \leq \bar{\delta}} \left\{ \frac{1}{\delta} \max \{ f_0(x^*) - f_0(x) : h(x) = \delta, f_0(x^*) - f_0(x) \geq 0 \} \right\} = \max_{0 \leq \delta \leq \bar{\delta}} \frac{1}{\delta} [f^* - f_\delta^*], \end{aligned}$$

где $f_\delta^* = \min \{ f_0(x) : h(x) = \delta, f_0(x^*) - f_0(x) \geq 0 \} = \min \{ f_0(x) : h(x) \leq \delta \}$.

Последнее равенство имеет место, поскольку минимум достигается на границе допустимого множества.

Рассматривая f_δ^* как функцию от значений правых частей ограничений (3.54), при достаточно малых значениях δ получаем $f_\delta^* = f^* - \delta \sum_{i=1}^m u_i^*$. Откуда следует утверждение леммы. ■

Понятно, что величина $\lambda_P(\varepsilon)$ существенно зависит от правила P . Для того, чтобы минимизировать $\lambda_P(\varepsilon)$, нужно использовать в качестве правила P для каждой точки x_k решение вспомогательной задачи (построение оптимального правила P)

$$z_k = P(x_k) = \arg \min_z \{ f_0(z) + \varepsilon \|z - x_k\| : z \in C \}. \quad (3.55)$$

Эта задача, конечно, не проще исходной. Но поскольку получаемое решение (точка z_k) используется для уточнения штрафного коэффициента (при нарушении условий (3.50)), можно ограничиться грубыми

приближениями, и использовать эвристические алгоритмы. Это приведет к завышенному значению $\lambda_P(\varepsilon)$.

Если точка z_k лежит на границе множества C то ограничение (3.50) можно представить в виде

$$f_0'(z_k, p_k) + \lambda h'(z_k, p_k) \geq \varepsilon - o(\|z_k - x_k\|), \quad (3.56)$$

где $f_0'(z_k, p_k)$, $h'(z_k, p_k)$ – производные по направлению p_k функций f_0, h в точке z_k , $p_k = (x_k - z_k) / \|x_k - z_k\|$, $o(\|z_k - x_k\|) \geq 0$. Последнее неравенство следует из выпуклости функций f_0, h .

Рассмотрим возможные подходы к определению правила P . Пусть $x \notin C$, $y_0 \in C$, $\pi_C(x, y_0)$ – точка пересечения отрезка $[x, y_0]$ с границей множества C .

Теорема 3.7. Пусть множество C ограничено, функция f_0 липшицева на C , задана точка $y_0 \in C$, $h(y_0) < 0$, $P(x) = \pi_C(x, y_0)$, для $x \notin C$. Тогда $\lambda_P(\varepsilon) < \infty$.

Доказательство. Обозначим M – константа Липшица функции f , $r = \max\{\|y_0 - z\| : z \in C\}$. Пусть $x \notin C$. Можно показать, что в силу выпуклости функций f, h имеют место неравенства

$$f_0(x) \geq f_0(z) - \left[\frac{f_0(y_0) - f_0(z)}{\|y_0 - z\|} \right] \cdot \|z - x\|, \quad h(x) \geq \frac{-h(y_0)}{\|y_0 - z\|} \|z - x\| \quad \text{где } z = \pi_C(x, y_0).$$

Учитывая,

что

$$f_0(z) + \varepsilon \|z - x\| - f_0(x) \leq \left(\frac{f_0(y_0) - f_0(z)}{\|y_0 - z\|} + \varepsilon \right) \cdot \|z - x\| \leq (M + \varepsilon) \|z - x\|, \quad \text{получаем}$$

$$\lambda_P(x, \varepsilon) \leq \frac{(M + \varepsilon) \|z - x\|}{h(x)} \leq \frac{(M + \varepsilon)}{|h(y_0)|} \|y_0 - z\| \leq r \frac{(M + \varepsilon)}{|h(y_0)|}. \quad \text{Откуда следует}$$

утверждение теоремы.

Использование правила $P(x) = \pi_C(x, y_0)$ позволяет строить достаточно эффективные процедуры уточнения штрафного коэффициента, однако при неудачном выборе точки y_0 значения коэффициентов могут устанавливаться достаточно большими.

Для улучшения качества рассмотренного правила P возможны два подхода. Первый заключается в том, чтобы использовать точку $z_k = \pi_C(x_k, y_0)$ как начальную для уточнения решения задачи (3.55). Более подробно это будет рассматриваться при описании применения предложенного подхода в схемах декомпозиции.

Второй подход – для каждой точки $x_k, k=1,2,\dots$, вместо начальной точки $y_0 \in C$ использовать некоторую вспомогательную точку $y_k \in C$. В этом случае полагаем $z_k = P(x_k) = \pi_C(x_k, y_k)$. Вопросы выбора вспомогательных точек $y_k \in C$ должны рассматриваться отдельно.

3.3.2. Использование функции Φ_β . Рассмотрим схему уточнения штрафных коэффициентов для функции Φ_β . Пусть последовательность точек $x_k, k=1,2,\dots$, генерируется сходящимся алгоритмом A при решении задачи (3.48). Точки z_k формируются в соответствии с правилом P : $z_k = P(x_k), z_k \in C, k=1,2,\dots$. Коэффициенты β заранее неизвестны и уточняются (увеличиваются) по ходу работы алгоритма. Обозначим $\beta^k = (\beta_i^k, i=1,\dots,m)$ значения штрафных коэффициентов на итерации k . При построении алгоритмом A точки x_k на итерации k используются значения β^k . Для точки x_k проверяется неравенство (3.50) при значениях $\beta = \beta^k$. Это неравенство можно представить в виде

$$f_0(x_k) + \sum_{i=1}^m \beta_i^k f_i^+(x_k) - f_0(z_k) \geq \varepsilon \|z_k - x_k\|. \quad (3.57)$$

Если (3.57) выполняется, полагаем $\beta_i^{k+1} = \beta_i^k, i=1,\dots,m$. В противном случае обозначим

$$\beta_P(x_k, \varepsilon) = \min_{\beta} \left\{ \max_i \beta_i : f_0(x_k) + \sum_{i=1}^m \beta_i f_i^+(x_k) - f_0(z_k) \geq \varepsilon \|z_k - x_k\|, \beta \geq 0 \right\}$$

, что эквивалентно

$$\beta_P(x_k, \varepsilon) = \max \left\{ 0, \frac{f_0(z_k) + \varepsilon \|z_k - x_k\| - f_0(x_k)}{\sum_{i=1}^m f_i^+(x_k)} \right\}, \quad (3.58)$$

и определим

$$\beta_i^{k+1} = \begin{cases} \beta_i^k, & \text{если } \beta_i^k \geq \beta_P(x_k, \varepsilon) \text{ или } f_i^+(x_k) = 0, \\ \beta_P(x_k, \varepsilon) + R, & \text{в противном случае,} \end{cases} \quad i = 1, \dots, m, \quad (3.59)$$

где $R > 0$ – заданный фиксированный параметр.

Сравнивая (3.51) и (3.58), получаем $\beta_P(x_k, \varepsilon) \leq \lambda_P(x_k, \varepsilon)$. Т.е., число уточнений коэффициентов β_i по ходу решения задачи (3.48) со штрафной функцией $\Phi_{\beta}(x)$ будет конечным, если $\lambda_P(\varepsilon) < \infty$. Возможности улучшения правила P , рассмотренные для функции F_{λ} , применимы также и для функции Φ_{β} .

3.3.3. Параметризация ограничений. Операция построения точки $z_k = \pi_C(x_k, y_0)$ для заданных x_k, y_0 может быть достаточно трудоемкой. Для упрощения процедуры уточнения штрафных коэффициентов в целом представим исходную задачу (3.1) в параметризованном виде: найти

$$\min \{ f_0(x) : x \in C(b) \}, \quad (3.60)$$

где $C(b) = \{ x : f_i(x) \leq b_i, i = 1, \dots, m, x \in R^n \}$, $b_i \geq 0, i = 0, \dots, m$.

Пусть начальная допустимая точка $y_0 \in C$ фиксирована и задано некоторое $\delta > 0$. Обозначим $z_{\delta}(x)$ точку, получаемую при сдвиге на величину δ из x в направлении на y_0 ,

$$z_{\delta}(x) = x + \delta p, \text{ где } p = (y_0 - x) / \|y_0 - x\|; \quad (3.61)$$

$$C_{\delta}(x) = C(b), \text{ где } b_i = f_i^+(z_{\delta}(x)), i = 1, \dots, m. \quad (3.62)$$

Пусть последовательность точек x_k , $k=1,2,\dots$, генерируется сходящимся алгоритмом при решении задачи (3.48) или (3.49). Нетрудно видеть, что если $z_\delta(x_k) \notin C$, то $z_\delta(x_k) = \pi_{C_\delta(x_k)}(x_k, y_0)$. Учитывая последнее соотношение, будем на итерации k уточнять штрафные коэффициенты не для задачи $\min\{f_0(x) : x \in C\}$, а для задачи $\min\{f_0(x) : x \in C_\delta(x_k)\}$. При этом не нужно искать точку пересечения отрезка $[x_k, y_0]$ с границей множества C .

Рассмотрим штрафную функцию $\Phi_\beta^\delta(x, x_k) = f_0(x) + \sum_{i=1}^m \beta_i (f_i(x) - f_i^+(z_\delta(x_k)))^+$

для множества $C_\delta(x_k)$ и функцию $F_\lambda^\delta(x, x_k) = f_0(x) + \lambda(h(x) - h(z_\delta(x_k)))^+$.

Заметим, что если $z_\delta(x_k) \notin C$, то

$$f_0(z_\delta(x_k)) = \Phi_\beta^\delta(z_\delta(x_k), x_k) = F_\lambda^\delta(z_\delta(x_k), x_k). \quad (3.63)$$

Условия сходимости к решению исходной задачи приведены в следующей теореме.

Теорема 3.8. Пусть множество C компактно и замкнуто, значения штрафных коэффициентов фиксированы, заданы: точка $y_0 \in C$, удовлетворяющая условию Слейтера, числа $\varepsilon > 0$, $\delta > 0$, последовательность точек x_k , $k=1,2,\dots$, сходящаяся к решению \tilde{x} задачи (3.48) (или решению задачи (3.49)). Пусть для точек $x_k \notin C$, $k=1,2,\dots$, выполняются неравенства

$$\varphi^\delta(x_k, x_k) \geq f_0(\tilde{z}_k) + \varepsilon \|\tilde{z}_k - x_k\|, \text{ если } \tilde{z}_k \notin C, \quad (3.64)$$

$$\varphi(x_k) \geq f_0(z_k) + \varepsilon \|z_k - x_k\|, \text{ если } \tilde{z}_k \in C, \quad (3.65)$$

где $\varphi(x) = \Phi_\beta(x)$, $\varphi^\delta(x, x_k) = \Phi_\beta^\delta(x, x_k)$, (соответственно $\varphi(x) = F_\lambda(x)$,

$\varphi^\delta(x, x_k) = F_\lambda^\delta(x, x_k)$), $\tilde{z}_k = z_\delta(x_k)$, $z_k = \pi_C(x_k, x_0)$. Тогда имеет место $\tilde{x} \in C$.

Доказательство. Предположим, что $\tilde{x} \notin C$. Обозначим $\tilde{z} = z_\delta(\tilde{x})$. Рассмотрим случай $\tilde{z} \notin C$. Очевидно, последовательность \tilde{z}_k сходится к \tilde{z} . Нетрудно проверить, что $\varphi(\tilde{x}) - \varphi(\tilde{z}) = \varphi^\delta(\tilde{x}, \tilde{x}) - \varphi^\delta(\tilde{z}, \tilde{x})$. Из (3.64), (3.63)

следует $\varphi^\delta(\tilde{x}, \tilde{x}) > f_0(\tilde{z}) = \varphi^\delta(\tilde{z}, \tilde{x})$. Откуда следует $\varphi(\tilde{x}) - \varphi(\tilde{z}) > 0$, что противоречит условию оптимальности точки \tilde{x} .

Случай $\tilde{z} \in C$ соответствует лемме 3.4. Теорема доказана. ■

3.4. Выводы

В разделе предложены новые методики определения значений штрафных коэффициентов точных штрафных функций, не требующие решения сложных вспомогательных задач оптимизации. Штрафные коэффициенты уточняются по ходу решения задачи безусловной оптимизации, определяемой точной штрафной функцией. Для каждой точки x_k , генерируемой алгоритмом безусловной оптимизации, определяется по некоторому правилу P допустимая точка y_k исходной задачи.

В подразделе 3.1 оценка штрафных коэффициентов производится на основании анализа значений производных целевой функции и функций ограничений по направлению, соединяющему точки y_k и x_k . Производные вычисляются в точке, лежащей на границе допустимого множества между точками y_k и x_k (если x_k не является допустимой точкой). Формулируются простые условия, при выполнении которых гарантируется сходимость к оптимальной точке исходной задачи. При нарушении этих условий штрафные коэффициенты должны увеличиваться. Приводятся условия, обеспечивающие конечность штрафных коэффициентов.

В подразделе 3.2 рассматриваются задачи невыпуклого программирования. Показано, что для уточнения штрафных коэффициентов можно ограничиться анализом поведения (значений производных по направлению) целевой функции и функций ограничений в δ -окрестности точки x_k на отрезке $[x_k, y_k]$, здесь $\delta > 0$ – некоторый фиксированный параметр. При заданном алгоритме безусловной оптимизации формулируется процедура решения исходной задачи с дополнительным ограничением $B_\rho(y^0) = \{x : \|x - y^0\| \leq \rho, x \in R^n\}$, где y^0 – начальная допустимая точка.

Величина ρ , значения штрафных коэффициентов определяются автоматически по ходу решения. Исходная задача с дополнительным ограничением решается приближенно. После этого уточняется начальная точка y^0 и процедура повторяется.

В подразделе 3.3 обобщаются предложенные подходы. Формулируются условия для штрафных коэффициентов, которые не требуют вычисления производных. Формулируется задача построения оптимального правила выбора допустимой точки, которая используется для оценки штрафных коэффициентов. Показано, что при использовании оптимального правила выбора допустимой точки устанавливаются значения штрафных коэффициентов, совпадающие с известными классическими оценками. Для приближенного решения задачи построения оптимального правила выбора допустимой точки могут использоваться приемы, предложенные в предыдущих подразделах.

РАЗДЕЛ 4. СХЕМЫ ДЕКОМПОЗИЦИИ ПО ПЕРЕМЕННЫМ В СТРУКТУРИРОВАННЫХ ЗАДАЧАХ ВЫПУКЛОЙ ОПТИМИЗАЦИИ

Применение схем декомпозиции по переменным для решения структурированных нелинейных задач выпуклой оптимизации связано с преодолением следующих проблем:

- точные решения отдельных подзадач получены быть не могут, необходимо использовать приближенные решения;
- подзадачи могут не иметь допустимых решений при некоторых значениях связывающих переменных, что требует разработки специальных приемов регуляризации исходной задачи;
- для точных штрафных функций, которые могут использоваться при регуляризации исходной задачи, невозможно определить значения штрафных коэффициентов существующими методами – трудоемкость вспомогательных задач сравнима с трудоемкостью исходной задачи.

В разделе рассматриваются новые подходы к решению указанных проблем и вопросы, связанные построением эффективных алгоритмов декомпозиции для специальных классов вложенных блочных задач линейного программирования. Предложенные алгоритмы применяются для решения задач перспективного планирования в электроэнергетике.

4.1. Декомпозиция по переменным и ε -субградиенты

Рассмотрим задачу математического программирования: найти

$$\min_{x,y} \left\{ \sum_{q=1}^Q f_q^0(x, y^q) : f_q^i(x, y^q) \leq 0, i=1, \dots, I_q, q=1, \dots, Q \right\} \quad (4.1)$$

где $f_q^i(x, y^q)$ – выпуклые собственные функции $[L + N_q]$ -мерного вектора (x, y^q) , $x \in R^L$, $y^q \in R^{N_q}$, $i=0, \dots, I_q$, $q=1, \dots, Q$.

Говорят, что задача (4.1) имеет блочную структуру со связывающими переменными.

Пусть связывающие переменные x фиксированы. Обозначим $D_q(x) = \{y^q \in R^{N_q} : f_q^i(x, y^q) \leq 0, i = 1, \dots, I_q\}$ и определим функцию $\Phi^q(x)$

$$\Phi^q(x) = \begin{cases} \min \{ f_q^0(x, y^q) : y^q \in D_q(x) \}, & x \in W_q, \\ +\infty, & x \notin W_q, \end{cases} \quad (4.2)$$

где W_q множество тех значений вектора x , для которых решение оптимизационной задачи в (4.2) существует.

В схемах декомпозиции решается следующая координирующая задача, которая эквивалентна исходной задаче (4.1): найти

$$\min \left\{ \sum_{q=1}^Q \Phi^q(x) : x \in E^L \right\}. \quad (4.3)$$

Свойства функций, входящих в координирующую задачу, исследовались в разных работах, в частности, в [184]. Показано, что W_q – выпуклое множество, а $\Phi^q(x)$ – выпуклые собственные функции. Там же были получены соотношения, позволяющие вычислять субградиенты функций $\Phi^q(x)$. При этом необходимо использовать точные решения подзадач (4.2). Вопросам использования приближенных решений подзадач в схемах декомпозиции посвящены работы [104, 105, 106].

Далее будут рассмотрены процедуры, позволяющие вычислять ε -субградиенты функций $\Phi^q(x)$ на основании приближенного решения оптимизационных задачи (4.2). В дальнейшем, если это не вызывает неоднозначность, индекс q будем опускать. Рассмотрим задачу: найти

$$\Phi(x) = \min \{ f^0(x, y) : y \in D(x) \}, \quad (4.4)$$

где $D(x) = \{y : f^i(x, y) \leq 0, i = 1, \dots, I\}$, $x \in R^L$, $y \in R^N$.

Положим $W = \{x : D(x) \neq \emptyset\}$, $\Phi(x) = \infty$, если $x \notin W$.

Обозначим $D = \{(x, y) : f^i(x, y) \leq 0, i = 1, \dots, I\}$.

Теорема 4.1. Пусть D – ограниченное множество, $D \in \text{int dom } f^0$, в точке (\bar{x}, \bar{y}) вычислены значения и ε_i -субградиенты $g^i = (g_x^i, g_y^i)$ функций f^i , $\varepsilon_i \geq 0, i = 0, 1, \dots, I$, такие, что для некоторых чисел $\bar{u}_i, i = 1, \dots, I$, выполняется

$$g_y^0 + \sum_{i=1}^I \bar{u}_i g_y^i = 0, \quad (4.5)$$

$$\bar{u}_i \geq 0, i = 1, \dots, I. \quad (4.6)$$

Тогда

$$1) \Phi(\bar{x}) \geq f^0(\bar{x}, \bar{y}) - \bar{\varepsilon},$$

2) если $\bar{y} \in D(\bar{x})$, то $\bar{\varepsilon}$ -субградиент функции $\Phi(x)$ в точке $x = \bar{x}$ может быть вычислен по формуле

$$g_{\Phi}^{\bar{\varepsilon}}(\bar{x}) = g_x^0 + \sum_{i=1}^I \bar{u}_i g_x^i, \quad (4.7)$$

где

$$\bar{\varepsilon} = \varepsilon_0 + \sum_{i=1}^I \bar{u}_i (\varepsilon_i - f^i(\bar{x}, \bar{y})). \quad (4.8)$$

Доказательство. По определению ε -субградиента имеем

$$f^0(x, y) \geq f^0(\bar{x}, \bar{y}) + (g_x^0, x - \bar{x}) + (g_y^0, y - \bar{y}) - \varepsilon^0, \quad (4.9)$$

$$f^i(x, y) \geq f^i(\bar{x}, \bar{y}) + (g_x^i, x - \bar{x}) + (g_y^i, y - \bar{y}) - \varepsilon^i, i = 1, \dots, I. \quad (4.10)$$

Умножая (4.10) на $\bar{u}_i \geq 0, i = 1, \dots, I$, складывая с (4.9) и учитывая (4.5), получаем

$$f^0(x, y) + \sum_{i=1}^I \bar{u}_i f^i(x, y) \geq f^0(\bar{x}, \bar{y}) + \sum_{i=1}^I \bar{u}_i f^i(\bar{x}, \bar{y}) +$$

$$\begin{aligned}
& + (g_x^0 + \sum_{i=1}^I \bar{u}_i g_x^i, x - \bar{x}) + (g_y^0 + \sum_{i=1}^I \bar{u}_i g_y^i, y - \bar{y}) - \varepsilon^0 - \sum_{i=1}^I \bar{u}_i \varepsilon^i = \\
& = f^0(\bar{x}, \bar{y}) + \sum_{i=1}^I \bar{u}_i f^i(\bar{x}, \bar{y}) + (g_x^0 + \sum_{i=1}^I \bar{u}_i g_x^i, x - \bar{x}) - \varepsilon^0 - \sum_{i=1}^I \bar{u}_i \varepsilon^i. \quad (4.11)
\end{aligned}$$

Пусть $x \in W$. Множество $D(x)$ ограничено (поскольку ограничено D), и задача (4.4) имеет решение, которое обозначим $y(x)$. Очевидно,

$$\Phi(x) \geq f^0(x, y(x)) + \sum_{i=1}^I \bar{u}_i f^i(x, y(x)), \text{ и, учитывая (4.11), получаем}$$

$$\Phi(x) \geq f^0(\bar{x}, \bar{y}) + \sum_{i=1}^I \bar{u}_i f^i(\bar{x}, \bar{y}) + (g_x^0 + \sum_{i=1}^I \bar{u}_i g_x^i, x - \bar{x}) - \varepsilon^0 - \sum_{i=1}^I \bar{u}_i \varepsilon^i. \quad (4.12)$$

При $x = \bar{x}$ получаем первое утверждение теоремы (если $\bar{x} \notin W$, то $\Phi(\bar{x}) = \infty$).

Пусть $\bar{y} \in D(\bar{x})$ (\bar{y} – допустимая точка). Тогда, очевидно, $f^0(\bar{x}, \bar{y}) \geq \Phi(\bar{x})$, и $\Phi(x) \geq \Phi(\bar{x}) + (g_x^0 + \sum_{i=1}^I \bar{u}_i g_x^i, x - \bar{x}) - \bar{\varepsilon}$. Отсюда следует

второе утверждение. Теорема доказана. ■

Рассмотрим случай, когда каждая компонента вектора y изменяется в некоторых пределах

$$L \leq y \leq U, \quad (4.13)$$

и ограничения (4.13) входят в общие ограничения задачи (4.4).

Теорема 4.2. [108]. Пусть D – ограниченное множество, $D \in \text{int dom } f^0$, в точке (\bar{x}, \bar{y}) вычислены значения и ε_i -субградиенты $g^i = (g_x^i, g_y^i)$ функций f^i , $\varepsilon_i \geq 0, i = 0, 1, \dots, I$, для вектора y выполняются условия (4.13), заданы некоторые числа $\bar{u}_i \geq 0, i = 1, \dots, I$. Тогда имеют место утверждения 1, 2 теоремы 4.1, где

$$\bar{\varepsilon} = \varepsilon_0 + \sum_{i=1}^I \bar{u}_i (\varepsilon_i - f^i(\bar{x}, \bar{y})) + \sum_{j=1}^N \theta_j p_j. \quad (4.14)$$

$$p = - \left(g_y^0 + \sum_{i=1}^I \bar{u}_i g_y^i \right), \quad (4.15)$$

$$\theta_j = \begin{cases} U_j - \bar{y}_j, & \text{если } p_j \geq 0, \\ L_j - \bar{y}_j, & \text{если } p_j < 0, \end{cases} \quad (4.16)$$

\bar{y}_j, p_j – компоненты векторов \bar{y}, p , числа U_j, L_j – верхние и нижние границы компонент \bar{y}_j , $j=1, \dots, N$.

Доказательство. Повторяя рассуждения доказательства предыдущей теоремы, получаем

$$\begin{aligned} f^0(x, y) + \sum_{i=1}^I \bar{u}_i f^i(x, y) &\geq f^0(\bar{x}, \bar{y}) + \sum_{i=1}^I \bar{u}_i f^i(\bar{x}, \bar{y}) + \\ &+ (g_x^0 + \sum_{i=1}^I \bar{u}_i g_x^i, x - \bar{x}) + (g_y^0 + \sum_{i=1}^I \bar{u}_i g_y^i, y - \bar{y}) - \varepsilon^0 - \sum_{i=1}^I \bar{u}_i \varepsilon^i. \end{aligned} \quad (4.17)$$

Заметим, что для вектора y , удовлетворяющего условию (4.13), имеет место неравенство

$$(g_y^0 + \sum_{i=1}^I \bar{u}_i g_y^i, y - \bar{y}) \geq - \sum_{j=1}^N \theta_j p_j. \quad (4.18)$$

Обозначим $y(x)$ решение задачи (4.4) для заданного x . Очевидно, что

$$\Phi(x) \geq f^0(x, y(x)) + \sum_{i=1}^I \bar{u}_i f^i(x, y(x)). \quad (4.19)$$

Подставляя точку $(\bar{x}, y(\bar{x}))$ в (4.17) и учитывая (4.18), (4.19), получаем первое утверждение теоремы.

Пусть \bar{y} – допустимая точка, т.е. $\bar{y} \in D(\bar{x})$. Тогда, очевидно, $f^0(\bar{x}, \bar{y}) \geq \Phi(\bar{x})$. Учитывая последнее неравенство, а также (4.18), (4.19), получаем из (4.17)

$$\Phi(x) \geq \Phi(\bar{x}) + (g_x^0 + \sum_{i=1}^I \bar{u}_i g_x^i, x - \bar{x}) + \sum_{i=1}^I \bar{u}_i f^i(\bar{x}, \bar{y}) - \sum_{j=1}^N \theta_j p_j - \varepsilon^0 - \sum_{i=1}^I \bar{u}_i \varepsilon^i$$

Откуда следует второе утверждение теоремы. ■

Рассмотрим полученные результаты применительно к задаче линейного программирования. Пусть $f^i(x, y) = a_x^i x + a_y^i y + b^i$, $i = 0, \dots, I$, где a_x^i, a_y^i – вектор-строки соответствующих размерностей.

Обозначим

$$A_x = \begin{Bmatrix} a_x^1 \\ \dots \\ a_x^I \end{Bmatrix}, A_y = \begin{Bmatrix} a_y^1 \\ \dots \\ a_y^I \end{Bmatrix}, b = \begin{Bmatrix} b^1 \\ \dots \\ b^I \end{Bmatrix}, b(x) = b + A_x x, b^0(x) = b^0 + a_x^0 x.$$

Тогда задачу линейного программирования в (4.4) и двойственную к ней можно представить в виде

$$\Phi(x) = \min_y \left\{ a_y^0 y + b^0(x) : A_y y + b(x) \leq 0 \right\}, \quad (4.20)$$

$$\max_u \left\{ b(x)^T u + b^0(x) : A_y^T u + a_y^0 = 0, u \geq 0 \right\}. \quad (4.21)$$

Лемма 4.1. Пусть $x = \bar{x}$, \bar{y} – допустимая точка задачи (4.20), \bar{u} – допустимая точка задачи (4.21). Тогда

$$\Phi(\bar{x}) \geq a_y^0 \bar{y} + b^0(\bar{x}) - \bar{\varepsilon} = b(\bar{x})^T \bar{u} + b^0(\bar{x}),$$

а $\bar{\varepsilon}$ -субградиент функции $\Phi(x)$ в точке $x = \bar{x}$ равен

$$g_{\Phi}^{\bar{\varepsilon}}(\bar{x}) = a_x^0 + A_x^T \bar{u}, \quad (4.22)$$

где $\bar{\varepsilon} = a_y^0 \bar{y} - b(\bar{x})^T \bar{u}$.

Доказательство. Поскольку $g^i = (a_x^i, a_y^i)$ – субградиенты функций f^i в произвольной точке, то $\varepsilon_i = 0$, $i = 0, 1, \dots, I$. Соотношение (4.22) очевидно.

Из (4.8) получаем,
$$\bar{\varepsilon} = -\sum_{i=1}^I \bar{u}_i f^i(\bar{x}, \bar{y}) = -(A_y \bar{y} + b(\bar{x}))^T \bar{u} =$$

$$= a_y^0 \bar{y} - a_y^0 \bar{y} - (A_y \bar{y} + b(\bar{x}))^T \bar{u} = a_y^0 \bar{y} - b(\bar{x})^T \bar{u}. \text{ Лемма доказана. } \blacksquare$$

Таким образом, если \bar{u} есть $\bar{\varepsilon}$ -оптимальное решение двойственной задачи (4.21), то соотношения (4.22) определяют $\bar{\varepsilon}$ -субградиент оптимального значения задачи (4.20).

Пусть задана совокупность $\{y_1, y_2, \dots, y_m\}$ точек из R^N , в каждой точке (\bar{x}, y_k) вычислены значения $f^{ik} = f^i(\bar{x}, y_k)$ и субградиенты $g^{ik} = (g_x^{ik}, g_y^{ik})$ функций f^i , $i = 0, \dots, I$, $k = 1, \dots, m$. Рассмотрим нижнюю линейную аппроксимацию задачи (4.4) при фиксированном $x = \bar{x}$: найти

$$\min_{y, \xi} \xi \quad (4.23)$$

при ограничениях

$$f^{ik} + (g_y^{ik}, y - y_k) \leq \xi, \quad k = 1, \dots, m, i = 0, \quad (4.24)$$

$$f^{ik} + (g_y^{ik}, y - y_k) \leq 0, \quad k = 1, \dots, m, i = 1, \dots, I. \quad (4.25)$$

Двойственная задача имеет вид: найти

$$\max_u \sum_{i=0}^I \sum_{k=1}^m (f^{ik} - (g_y^{ik}, y_k)) u_{ik} \quad (4.26)$$

при ограничениях

$$\sum_{k=1}^m u_{0k} = 1, \quad (4.27)$$

$$\sum_{i=0}^I \sum_{k=1}^m u_{ik} g_y^{ik} = 0, \quad (4.28)$$

$$u_{ik} \geq 0, i = 0, \dots, I, k = 1, \dots, m. \quad (4.29)$$

Теорема 4.3. Пусть задана точка $\bar{y} \in D(\bar{x})$, ξ^*, y^* – оптимальное решение, \bar{u}_{ik} – оптимальные значения двойственных переменных,

$k = 1, \dots, m, i = 0, \dots, I$ задачи (4.23)–(4.25). Тогда

$$\xi^* \leq \Phi(\bar{x}) \leq f^0(\bar{x}, \bar{y}), \quad g_{\Phi}^{\bar{\varepsilon}}(\bar{x}) = \sum_{k=1}^m \sum_{i=0}^I \bar{u}_{ik} g_x^{ik}, \quad \text{где } \bar{\varepsilon} = f^0(\bar{x}, \bar{y}) - \xi^*.$$

Доказательство. Обозначим $\bar{f}^i = f^i(\bar{x}, \bar{y})$. Легко видеть, что g^{ik} есть ε^{ik} -субградиент функции f^i в точке (\bar{x}, \bar{y}) , где

$$\begin{aligned} \varepsilon^{ik} &= \bar{f}^i - f^{ik} - (g_y^{ik}, \bar{y} - y_k) = \bar{f}^i - f^{ik} - (g_y^{ik}, \bar{y} - y_k + y^* - y^*) = \\ &= \bar{f}^i + (g_y^{ik}, y^* - \bar{y}) - [f^{ik} + (g_y^{ik}, y^* - y_k)]. \end{aligned}$$

Заметим, что если $\bar{u}_{ik} > 0$, то

$$f^{ik} + (g_y^{ik}, y^* - y_k) = \begin{cases} \xi^*, & i = 0, \\ 0, & i > 0. \end{cases} \quad (4.30)$$

Пусть $\bar{u}_i = \sum_{k=1}^m \bar{u}_{ik} > 0$. Умножая на \bar{u}_{ik} неравенства

$$f^i(x, y) \geq \bar{f}^i + (g_x^{ik}, x - \bar{x}) + (g_y^{ik}, y - \bar{y}) - \varepsilon^{ik}, \quad k = 1, \dots, m,$$

и суммируя по k , нетрудно видеть, что вектор $g^i = (\bar{u}_i)^{-1} \left(\sum_{k=1}^m \bar{u}_{ik} g^{ik} \right)$ есть

ε^i -субградиент функции f^i в точке (\bar{x}, \bar{y}) , где $\varepsilon^i = (\bar{u}_i)^{-1} \left(\sum_{k=1}^m \bar{u}_{ik} \varepsilon^{ik} \right)$.

Нетрудно проверить, что векторы g^i и числа \bar{u}_i удовлетворяют соотношениям (4.5), (4.6). Подставляя соответствующие выражения в (4.7),

(4.8) и учитывая (4.30), а также, что $\sum_{k=1}^m \bar{u}_{0k} = 1$, получаем

$$\bar{\varepsilon} = \bar{f}^0 - \xi^* + \left(\sum_{k=1}^m \sum_{i=0}^I \bar{u}_{ik} g_y^{ik}, y^* - \bar{y} \right) = \bar{f}^0 - \xi^*. \quad \text{Последнее равенство следует из}$$

(4.28). Теорема доказана. ■

Приведенная теорема позволяет строить правила остановки в алгоритмах приближенного решения задачи (4.4).

В случае, когда в задаче (4.4) целевая функция нелинейная, но линейны ограничения, для ее решения могут использоваться алгоритмы, которые на каждом шаге генерируют допустимые точки, например, метод Келли [41]. В общем случае используемые алгоритмы должны формировать подпоследовательность допустимых точек, сходящуюся к решению задачи (4.4).

При формировании линейной аппроксимации в задачу (4.23)–(4.25) целесообразно включать только наиболее существенные ограничения. Правила, которые при этом используются, являются эвристическими, однако позволяют существенно понизить размерность генерируемой задачи.

Для решения задачи (4.4) в общем случае могут применяться различные методы – r -алгоритм Н.З. Шора с использованием негладких штрафных функций [184], комбинированный метод Б.Н. Пшеничного [156] и другие. Такие методы имеют высокую скорость сходимости, но точки, которые генерируются на каждой итерации, могут не принадлежать допустимой области, только в пределе сходясь к оптимальному решению. Эта особенность затрудняет вычисление $\bar{\varepsilon}$ -субградиентов функции $\Phi(x)$.

При практическом применении негладких штрафных функций в задаче (4.4) вместо $D(x)$ может использоваться множество $D^\delta(x) = \{(x, y) : f^i(x, y) \leq -\delta, i = 1, \dots, I\}$, где параметр $\delta > 0$ уменьшается по ходу работы алгоритма оптимизации и в пределе стремится к нулю. Такой прием позволяет генерировать допустимые точки исходной задачи.

4.2. Регуляризация задач в схемах декомпозиции

При реализации схемы декомпозиции необходимо преодолеть существенные трудности, связанные с неявным описанием множеств W_q , $q = 1, \dots, Q$, на которых определены функции $\Phi^q(x)$.

Рассмотренные подходы к вычислению ε -субградиентов функции $\Phi^q(x)$ применимы, если выполняются условия $D_q(x) \neq \emptyset$. В случае, когда $D_q(x) = \emptyset$, а отдельные подзадачи являются задачами линейного программирования, используются другие приемы [33, 58, 63] и формируются дополнительные отсекающие ограничения в пространстве связывающих переменных. Трудоемкость построения таких дополнительных ограничений соизмерима с затратами на получение точных решений подзадач.

Для нелинейных подзадач ситуация существенно усложняется. Обозначим

$$D_q^m(x) = \left\{ y : f^{ik} + (g_y^{ik}, y - y_k) \leq 0, k = 1, \dots, m, i = 1, \dots, I \right\}.$$

Множество $D_q^m(x)$ есть аппроксимация множества $D_q(x)$. Если $D_q^m(x) = \emptyset$, то можно построить [58] дополнительное ограничение вида $\langle a, x \rangle + b \leq 0$, которому должны удовлетворять связывающие переменные (точки множества W_q). Совокупность таких ограничений при разных значениях вектора x формирует аппроксимацию множеств W_q . Однако, для нелинейных задач в случае, когда $D_q(x) = \emptyset$, построение совокупности точек $\{y_1, y_2, \dots, y_m\}$, для которой выполняется условие $D_q^m(x) = \emptyset$, может быть очень трудоемким.

Рассмотрим ряд подходов для преодоления этих трудностей.

1. Пусть функции $f_q^i(x, y^q)$, $i=0, \dots, I_q$, определены на всем пространстве $E^L \times E^{N_q}$, $q=1, \dots, Q$. Используя точные штрафные функции, заменим [184] исходную задачу (4.1) на следующую: найти

$$\min_{x, y, v} \sum_{q=1}^Q \left\{ f_q^0(x, y^q) + M_q \sum_{i=1}^{I_q} v_q^i \right\} \quad (4.31)$$

при ограничениях

$$f_q^i(x, y^q) - v_q^i \leq 0, v_q^i \geq 0, i=1, \dots, I_q, q=1, \dots, Q, \quad (4.32)$$

где M_q достаточно большие положительные числа. Задача (4.2) заменяется при этом на такую: найти

$$\Psi^q(x) = \min_{y^q, v^q} f_q^0(x, y^q) + M_q \sum_{i=1}^{I_q} v_q^i \quad (4.33)$$

при ограничениях

$$f_q^i(x, y^q) - v_q^i \leq 0, v_q^i \geq 0, i=1, \dots, I_q. \quad (4.34)$$

Ясно, что для любого x задача (4.33)–(4.34) имеет непустое допустимое множество, которое удовлетворяет условию Слейтера, а функция $\Psi^q(x)$ определена для всех x . Таким образом, координирующая задача

$$\min \left\{ \sum_{q=1}^Q \Psi^q(x) : x \in R^L \right\} \quad (4.35)$$

есть задача выпуклого программирования без ограничений.

Количество вспомогательных переменных регуляризации равно числу ограничений в каждой подзадаче.

2. Пусть функции $f_q^i(x, y^q)$, $i=0, \dots, I_q$, определены не на всем пространстве $R^L \times R^{N_q}$, $q=1, \dots, Q$. Для регуляризации исходной задачи (4.1)

введем вспомогательные переменные $v^q = (v_1^q, \dots, v_L^q)$, $q = 1, \dots, Q$, и рассмотрим следующую задачу [105]: найти

$$\min_{x, y, v} \sum_{q=1}^Q \left\{ f_q^0(v^q, y^q) + M_q \sum_{l=1}^L |x_l - v_l^q| \right\} \quad (4.36)$$

при ограничениях

$$f_q^i(v^q, y^q) \leq 0, i = 1, \dots, I_q, q = 1, \dots, Q, \quad (4.37)$$

где $v^q \in R^L$, $q = 1, \dots, Q$.

Если задача (4.1), имеет решение, то при достаточно больших положительных значениях M_q , $q = 1, \dots, Q$, решения задач (4.1) и (4.36), (4.37) совпадают.

Обозначим $z^q = (v^q, y^q)$, $z^q \in R^L \times R^{N_q}$. При фиксированных значениях x задача (4.36), (4.37) распадается на подзадачи

$$\Psi^q(x) = \min_{z^q} \left\{ f_q^0(z^q) + M_q \sum_{l=1}^L |x_l - z_l^q| : f_q^i(z^q) \leq 0, i = 1, \dots, I_q \right\}. \quad (4.38)$$

Пусть системы ограничений задач (4.38) совместны, и эти задачи имеют решение при $M_q = 0$; тогда функции $\Psi^q(x)$ определены при любых x и, соответственно, координирующая задача есть задача выпуклого программирования без ограничений. Количество вспомогательных переменных регуляризации равно числу переменных в каждой подзадаче.

3. Будем предполагать, что для каждого блока q задана точка x_0^q такая, что $x_0^q \in W_q$, $q = 1, \dots, Q$, т.е. решение оптимизационной задачи в (4.2) существует для точки x_0^q . Введем вспомогательные переменные α_q , $q = 1, \dots, Q$, $\alpha_q \in R$, и рассмотрим следующую задачу [108]: найти

$$\min_{x, y, \alpha} \sum_{q=1}^Q f_q^0(x_0^q + \alpha_q(x - x_0^q), y^q) \quad (4.39)$$

при ограничениях

$$f_q^i(x_0^q + \alpha_q(x - x_0^q), y^q) \leq 0, i = 1, \dots, I_q, q = 1, \dots, Q, \quad (4.40)$$

$$0 \leq \alpha_q \leq 1, q = 1, \dots, Q, \quad (4.41)$$

$$1 - \alpha_q \leq 0, q = 1, \dots, Q. \quad (4.42)$$

Очевидно, что задача (4.39)–(4.42) эквивалентна исходной задаче (4.1).

Обозначим $\varphi_q^i(x, \alpha_q, y^q) = f_q^i(x_0^q + \alpha_q(x - x_0^q), y^q)$, $i = 0, \dots, I_q$, $q = 1, \dots, Q$ и рассмотрим задачу

$$\min_{x, y, \alpha_{q=1}^Q} \sum_{q=1}^Q \left\{ \varphi_q^0(x, \alpha_q, y^q) + M_q(1 - \alpha_q)^+ \right\} \quad (4.43)$$

при ограничениях

$$\varphi_q^i(x, \alpha_q, y^q) \leq 0, i = 1, \dots, I_q, q = 1, \dots, Q, \quad (4.44)$$

$$0 \leq \alpha_q \leq 1, q = 1, \dots, Q, \quad (4.45)$$

где функция $(t)^+ = \begin{cases} t, & \text{если } t > 0, \\ 0 & \text{в противном случае.} \end{cases}$

Из свойств негладких штрафных функций следует, что если задача (4.39)–(4.42), имеет решение, то при достаточно больших положительных значениях M_q , $q = 1, \dots, Q$, решения задач (4.39)–(4.42) и (4.43)–(4.45) совпадают.

При фиксированных значениях x задача (4.43)–(4.45) распадается на подзадачи для каждого блока q : найти

$$\Psi^q(x) = \min_{y^q, \alpha_q} \left\{ \varphi_q^0(x, \alpha_q, y^q) + M_q[1 - \alpha_q]^+ \right\} \quad (4.46)$$

при ограничениях

$$\varphi_q^i(x, \alpha_q, y^q) \leq 0, i = 1, \dots, I_q, \quad (4.47)$$

$$0 \leq \alpha_q \leq 1. \quad (4.48)$$

Заметим, что при фиксированном $\alpha_q = 0$ задача (4.46)–(4.48)

имеет решение для любого x . Т.е. функция $\Psi^q(x)$ определена при любых x . Для каждой подзадачи вводится одна вспомогательная переменная регуляризации.

4.3. Точные штрафные функции в схемах декомпозиции по переменным

Существенной проблемой при использовании рассмотренных выше подходов является определение значений штрафных коэффициентов, при которых штрафная функция становится точной. Выбор слишком больших значений приводит к неустойчивости используемых алгоритмов, ошибкам округления, и т.п.

Рассмотрим возможности применения результатов, изложенных в разделе 3, для задачи: найти

$$F^* = \min_x \sum_{q=1}^Q f_q^0(x^0, x^q) \quad (4.49)$$

при ограничениях

$$(x^0, x^q) \in C_q = \{(x^0, x^q) : f_q^1(x^0, x^q) \leq 0\}, \quad q = 1, \dots, Q, \quad (4.50)$$

где $f_q^i(x^0, x^q)$ – выпуклые собственные функции, $i = 0, 1$, $x^q \in R^{N_q}$, $q = 0, 1, \dots, Q$. Предполагается, что C_q – компактные множества, $C_q \subseteq \text{dom } f_q^0 \cap \text{dom } f_q^1$, $q = 1, \dots, Q$.

При условии того, что функции $f_q^i(x^0, x^q)$, $i = 0, 1$, $q = 1, \dots, Q$, определены для любых значений переменных, использование точных штрафных функций позволяет привести задачу (4.49)–(4.50) к виду: найти

$$\min_x \sum_{q=1}^Q [f_q^0(x^0, x^q) + \beta_q f_q^+(x^0, x^q)]. \quad (4.51)$$

Координирующая задача (4.3) и подзадачи (4.2) также оказываются задачами безусловной оптимизации. Здесь

$$f_q^+(x^0, x^q) = \max\{0, f_q^1(x^0, x^q)\}.$$

Для блочной задачи со связывающими переменными (4.49)–(4.50) обозначим:

$$x = (x^q, q = 0, \dots, Q),$$

$$C = \{x : (x^0, x^q) \in C_q, q = 1, \dots, Q\},$$

$$y_0 - \text{начальная точка такая, что } f_q^1(y_0^0, y_0^q) < 0, q = 1, \dots, Q,$$

x_k – точка, полученная на текущей итерации при минимизации штрафной функции исходной блочной задачи (4.51),

$$z_k = P(x_k) - \text{точка, полученная по правилу } P, z_k \in C, k = 1, 2, \dots$$

Неравенство (3.57) принимает вид

$$\sum_{q=1}^Q \left[f_q^0(x_k^0, x_k^q) + \beta_q^k f_q^+(x_k^0, x_k^q) \right] - \sum_{q=1}^Q f_q^0(z_k^0, z_k^q) \geq \varepsilon \|z_k - x_k\|. \quad (4.52)$$

Соответственно соотношение (3.58) переходит в

$$\beta_P(x_k, \varepsilon) = \max \left\{ 0, \frac{\sum_{q=1}^Q f_q^0(z_k^0, z_k^q) + \varepsilon \|z_k - x_k\| - \sum_{q=1}^Q f_q^0(x_k^0, x_k^q)}{\sum_{q=1}^Q f_q^+(x_k^0, x_k^q)} \right\}, \quad (4.53)$$

Задача (3.55) построения оптимального правила P может быть представлена следующим образом

$$z_k = P(x_k) = \arg \min_z \left\{ \sum_{q=1}^Q f_q^0(z^0, z^q) + \varepsilon \|z - x_k\| : (z^0, z^q) \in C_q, q = 1, \dots, Q \right\}. \quad (4.54)$$

Пусть на некоторой итерации при решении координирующей задачи выполнен переход в точку (значения связывающих переменных) x_k^0 .

Обозначим $x_k^q, q=1, \dots, Q$, значения блочных переменных, полученные к моменту перехода в точку $x_k^0, x_k = (x_k^0, x_k^q, q=1, \dots, Q)$. Если $x_k \notin C$, производится построение точки z_k (правила P), которое будем выполнять последовательно по этапам $q=0, \dots, Q$.

На этапе $q=0$ определим $\bar{z}_k = \pi_C(x_k, y_0)$, при этом $f_q^1(\bar{z}_k^0, \bar{z}_k^q) \leq 0, q=1, \dots, Q$. Значения связывающих переменных на последующих этапах изменяться не будут $z_k^0 = \bar{z}_k^0$.

На этапе $q=1, \dots, Q$ выполняется отдельная обработка блока q . Если $f_q^1(z_k^0, \bar{z}_k^q) = 0$, полагаем $z_k^q = \bar{z}_k^q$. В случае $f_q^1(z_k^0, \bar{z}_k^q) < 0$ находится приближенное решение задачи (4.54) для блока q . Связывающие переменные и переменные остальных блоков считаются зафиксированными. Аналогичные действия производятся при решении подзадач в схеме декомпозиции при фиксированных значениях связывающих переменных.

4.4. Ограниченные области определения функций

Если функции $f_q^i(x^0, x^q), i=0,1, q=1, \dots, Q$ определены не для всех значений переменных, необходимо сочетать выпуклое продолжение этих функций на все пространство с использованием точных штрафов. Вопросы выпуклого продолжению функций рассматривались в разделе 2.

Рассматривается блочная задача со связывающими переменными (4.49)–(4.50). Будем считать заданной начальную точку $y_0 = (y_0^q, q=0, \dots, Q)$ такую, что $f_q^1(y_0^0, y_0^q) < 0, q=1, \dots, Q, f_q^0(x^0, x^q)$ – выпуклые липшицевы функции, $q=1, \dots, Q$. Обозначим $\pi_{C_q}(x^0, x^q, y_0^0, y_0^q)$ точку пересечения луча, исходящего из (y_0^0, y_0^q) и проходящего через точку (x^0, x^q) , с границей множества C_q . Определим функции $\chi_{E_q}(x^0, x^q), \psi_{E_q}(x^0, x^q), R_{C_q}(x^0, x^q)$ по аналогии с (2.30), (2.28), (2.31).

Поскольку функция $f_q^1(x^0, x^q)$ определена не при любых значениях переменных, для описания множества C_q будем использовать функцию Минковского $R_{C_q}(x^0, x^q)$. Рассмотрим задачу: найти

$$F^* = \min_x \sum_{q=1}^Q \psi_{E_q}^q(x^0, x^q) \quad (4.55)$$

при ограничениях

$$(x^0, x^q) \in C_q = \left\{ (x^0, x^q) : R_{C_q}(x^0, x^q) \leq 1 \right\}, \quad q = 1, \dots, Q. \quad (4.56)$$

В соответствии с теоремой 2.1 для функции $f_q^0(x^0, x^q)$ и множества C_q существует конечное \bar{E}_q такое, что для всех $E_q \leq \bar{E}_q$ функция $\psi_{E_q}^q(x^0, x^q)$ выпуклая, $q = 1, \dots, Q$. На множестве C_q функции f_q^0 и $\psi_{E_q}^q$ совпадают. Таким образом, при условии $E_q \leq \bar{E}_q$, $q = 1, \dots, Q$, задача (4.55)–(4.56) и исходная задача (4.49)–(4.50) эквивалентны. Для обеспечения условия $E_q \leq \bar{E}_q$ необходимо проверять на каждой итерации аналог неравенства (2.42) теоремы 2.3.

Для задачи (4.55)–(4.56) могут применяться точные штрафные функции, и подходы к определению значений штрафных коэффициентов, рассмотренные ранее.

Рассмотрим задачу (4.55) без ограничений (4.56).

Теорема 4.4. Пусть множества $C_q = \left\{ (x^0, x^q) : f_q^1(x^0, x^q) \leq 0 \right\}$ выпуклы и замкнуты, функции f_q^0 выпуклы и липшицевы на C_q , задана начальная точка $y_0 = (y_0^q, q = 0, \dots, Q)$, для которой $f_q^1(y_0^0, y_0^q) < 0$, $q = 1, \dots, Q$. Тогда существует набор чисел $E^* = (E_q^*, q = 1, \dots, Q)$ такой, что для любых

$E_q, E_q < E_q^*, q=1, \dots, Q$, решения задачи (4.49)–(4.50) и задачи (4.55)

совпадают.

Доказательство. Рассмотрим задачу (4.55)–(4.56). Пусть $E_q = \tilde{E}_q \leq \bar{E}_q$, $q=1, \dots, Q$, т.е. задачи (4.55)–(4.56) и (4.49)–(4.50) эквивалентны. Для функции

$\Phi_\beta(x) = \sum_{q=1}^Q \left(\Psi_{\tilde{E}_q}^q(x^0, x^q) + \beta_q \left[R_{C_q}(x^0, x^q) - 1 \right]^+ \right)$ существуют коэффициенты

$\beta_q^*, q=1, \dots, Q$ такие, что $\Phi_\beta(x)$ – точная штрафная функция, если $\beta_q \geq \beta_q^*, q=1, \dots, Q$.

Покажем, что $\Psi_{\tilde{E}_q}^q(x^0, x^q) + \beta_q \left[R_{C_q}(x^0, x^q) - 1 \right]^+ = \Psi_{\tilde{E}_q - \beta_q}^q(x^0, x^q)$. В самом деле, при $(x^0, x^q) \in C_q$ правая и левая часть соотношения равна $f_q^0(x^0, x^q)$. Пусть $(x^0, x^q) \notin C_q$, тогда

$$\begin{aligned} & \Psi_{\tilde{E}_q}^q(x^0, x^q) + \beta_q \left[R_{C_q}(x^0, x^q) - 1 \right]^+ = \\ & = \tilde{E}_q + \left(f_q^0(\pi_{C_q}(x^0, x^q, y_0^0, y_0^q)) - \tilde{E}_q \right) \frac{\| (x^0, x^q) - (y_0^0, y_0^q) \|}{\| \pi_{C_q}(x^0, x^q, y_0^0, y_0^q) - (y_0^0, y_0^q) \|} + \\ & + \beta_q \left(\frac{\| (x^0, x^q) - (y_0^0, y_0^q) \|}{\| \pi_{C_q}(x^0, x^q, y_0^0, y_0^q) - (y_0^0, y_0^q) \|} - 1 \right) = \Psi_{\tilde{E}_q - \beta_q}^q(x^0, x^q). \end{aligned}$$

Откуда, полагая $E_q^* = \tilde{E}_q - \beta_q^*, q=1, \dots, Q$, получаем утверждение теоремы. ■

4.5. Особенности программной реализации для специального класса задач

Вычислительные эксперименты по использованию предлагаемого подхода проводились при решении тестовых задач, являющихся модификацией известной трудной задачи из [156]. Содержательно рассматриваемые задачи заключаются в поиске минимума потенциальной

энергии для цепочки, подвешенной за крайние звенья. В модифицированной задаче рассматривается несколько цепочек, имеющих одно общее звено. Крайние звенья зафиксированы, требуется найти минимум потенциальной энергии. При фиксированном положении общего звена задача распадается на подзадачи для каждой отдельной цепочки. Рассматриваемая задача является задачей квадратичного программирования.

В первоначальной программной реализации предложенной схемы декомпозиции для решения координирующей задачи (4.3) и подзадач вида (4.4) использовался r -алгоритм. Для решения задач линейного программирования (4.23)–(4.25) (линейной аппроксимации подзадач) использовалась стандартная программа из известной библиотеки IMSL.

В ходе отладки программных средств выяснилось, что при приближении к оптимуму подзадачи (4.4) линейные аппроксимации (4.23)–(4.25) становятся плохо обусловленными, что приводит к большим погрешностям при использовании программных средств линейного программирования. В связи с этим в дальнейшем для решения подзадач (4.4) использовалась программная реализация РНК-метода [156].

Анализ работы созданных программных средств декомпозиции показал, что существенное повышение эффективности алгоритмов может быть достигнуто за счет использования информации, накопленной на предыдущих итерациях координирующей задачи, при решении подзадач (4.4).

Кроме того, эффективность схемы декомпозиции задачи в целом существенно зависит от регулировки точности решения подзадач по ходу решения координирующей задачи: на первых шагах нецелесообразно выбирать высокую точность решения подзадач. Эта точность должна увеличиваться по мере приближения к оптимальному решению координирующей задачи.

4.6. Вложенные блочные задачи линейного программирования

В подразделе рассматриваются задачи линейного программирования большой размерности, являющиеся естественным обобщением блочных задач со связывающими переменными. Такие задачи имеют место, если каждый блок блочной задачи в свою очередь является блочной задачей.

Эти задачи возникают в таких прикладных областях, как моделирование в экономике [142], многоэтапное стохастическое программирование [50, 40, 11] и др., при рассмотрении динамических процессов с дискретным временем в некотором интервале планирования $[0, \dots, T]$. Изменения состояния процесса за период t , $t = 0, \dots, T$, зависят от вектора случайных величин (внешних факторов), значения которых становятся известными в начале периода t , и от вектора управлений, значения которых определяются по состоянию процесса и по значениям внешних факторов в начале периода t .

Качество управления определяется целевым критерием (величина доходов или убытков), который может быть вычислен для любой реализации случайных величин во времени и выбранных значений вектора управлений. Если правило определения значений вектора управлений задано для каждой реализации случайных величин, то качество этого правила определяется как математическое ожидание целевого критерия.

Оптимальным управлением является правило доставляющее минимум (для убытков) или максимум (для доходов) математического ожидания целевого критерия по множеству возможных управлений (правил определения значений вектора управлений).

Для приближенного определения оптимального управления в настоящее время достаточно широко используется сценарный подход, в котором для представления совместного распределения случайных величин (внешних факторов) рассматривается конечное дерево возможных

реализаций, каждой вершине которого поставлена в соответствие условная вероятность реализации данного сценария. Рассматриваемая задача формулируется относительно дерева сценариев и называется детерминированным аналогом исходной задачи.

Для многих прикладных задач детерминированный аналог является вложенной блочной задачей линейного программирования.

Рассматриваемые задачи достигают очень большой размерности. Для их решения разрабатываются специализированные алгоритмы, применяются суперкомпьютеры с большим числом процессоров. В основе этих алгоритмов используются различные подходы, этим проблемам посвящено много работ (см., например, [28]).

Структура рассматриваемых задач описывается совокупностью связанных выпуклых подзадач – деревом подзадач. В предлагаемой схеме декомпозиции для каждой подзадачи строится линейная аппроксимация и обычная блочная задача со связывающими переменными. Алгоритм решения исходной задачи состоит в последовательном решении (декомпозиции) блочных задач для отдельных вершин с регулируемой точностью. При этом увеличивается точность аппроксимаций подзадач. Особенностью алгоритма является возможность согласования точности аппроксимаций для различных подзадач по ходу решения исходной задачи. Вычислительные эксперименты показали эффективность предлагаемого подхода.

Возможности разработанных программных средств исследуются на примере стохастической задачи планирования инвестиций в электроэнергетике, являющейся модификацией модели из [49]. Приводятся результаты вычислительных экспериментов по сравнению разработанных и существующих программных средств.

Рассматривается также задача перспективного планирования в электроэнергетике, в которой должны учитываться суточные неравномерности потребления электроэнергии и особенности

функционирования энергосистемы. Предложена упрощенная математическая модель задачи, которая при дополнительных ограничениях является многоэтапной линейной задачей стохастического программирования. В общем случае задача нелинейна и невыпукла по некоторым переменным.

4.6.1. Описание задач и схема декомпозиции

1. Будем считать заданным некоторое дерево (V, E) . Каждой вершине $q \in V$ поставим в соответствие оптимизационную задачу, зависящую от параметров. Вектор параметров обозначим x^q , вектор переменных, по которым производится оптимизация – y^q . Задачи, соответствующие разным вершинам дерева (V, E) , являются связанными. Связь заключается в том, что если вершина q есть непосредственный потомок вершины p , то

$$x^q = (x^p, y^p). \quad (4.57)$$

Для корневой вершины r будем предполагать, что $x^r = 0$.

Пусть $f^q(x^q)$ – некоторая функция параметров x^q вершины q . В дальнейшем будем считать эквивалентными записи $f^q(x^q)$ и $f^q(x^p, y^p)$, где q есть непосредственный потомок вершины p .

Обозначим $S(q)$ множество непосредственных потомков вершины q . Оптимизационная задача для вершины $q \in V$ имеет вид: найти

$$\varphi^q(x^q) = \min_{y^q} \left\{ c^q y^q + \sum_{s \in S(q)} \varphi^s(x^q, y^q) \right\} \quad (4.58)$$

при ограничениях

$$A^q y^q \leq d^q - B^q x^q, \quad (4.59)$$

где $x^q \in R^{n_q}$, $y^q \in R^{m_q}$, $c^q \in R^{m_q}$, $d^q \in R^{k_q}$, A^q, B^q – матрицы размерности $k_q \times m_q$ и $k_q \times n_q$.

Матрицы A^q, B^q , вектор-строка c^q и вектор d^q считаются заданными для каждой вершины $q \in V$.

Для висячих вершин дерева (V, E) задачи (4.58)–(4.59) являются задачами линейного программирования, для остальных вершин задачи (4.58)–(4.59) определяются рекуррентно. Известно (см., например, [184]), что функции $\varphi^q(x^q)$, $q \in V$, являются выпуклыми.

Для корневой вершины оптимизационная задача от параметров не зависит (или, что эквивалентно, значения параметров зафиксированы).

Совокупность задач (4.58)–(4.59) для всех вершин дерева (V, E) может быть представлена одной задачей линейного программирования большой размерности, обладающей вложенной структурой. В случае, когда дерево состоит из корня и набора висячих вершин, эта задача является блочной задачей со связывающими переменными и с дополнительными ограничениями (4.59). Задача (4.58)–(4.59) для корневой вершины является координирующей задачей в схеме декомпозиции по переменным.

2. Рассмотрим выпуклую параметрическую задачу следующего вида: найти

$$\Phi(x) = \min_y \{(c, y) + F(x, y)\} \quad (4.60)$$

при ограничениях

$$Ay \leq d - Bx. \quad (4.61)$$

Пусть задана некоторая кусочно-линейная нижняя аппроксимация $H(x, y)$ для функции $F(x, y)$: $H(x, y) \leq F(x, y)$,

$$H(x, y) = \max_i \{(h_x^i, x) + (h_y^i, y) + \eta_i, i = 1, \dots, I\}. \quad (4.62)$$

Положим $\xi^*(x) = \min_y \{(c, y) + H(x, y) : Ay \leq d - Bx\}$. Очевидно,

$$\xi^*(x) \leq \Phi(x).$$

Пусть значение x зафиксировано. Нетрудно видеть, что

$$\xi^*(x) = \min_{y, \xi} \xi \quad (4.63)$$

при ограничениях

$$\xi \geq (h_x^i, x) + (c + h_y^i, y) + \eta_i, \quad i = 1, \dots, I, \quad (4.64)$$

$$Ay \leq d - Bx. \quad (4.65)$$

Обозначим

$w_i, i = 1, \dots, I$ – двойственные переменные, соответствующие ограничениям (4.64), $w = (w_1, \dots, w_I)$,

u – вектор двойственных переменных, соответствующих ограничениям (4.65),

G – матрица, строками которой являются векторы $c + h_y^i, i = 1, \dots, I$,

H_x – матрица, строками которой являются векторы $h_x^i, i = 1, \dots, I$,

$\eta = (\eta_1, \dots, \eta_I)$.

Задача, двойственная к (4.63)–(4.65), имеет вид: найти

$$\max_{u, w} \{(H_x x + \eta, u) + (Bx - d, w)\} \quad (4.66)$$

при ограничениях

$$\sum_{i=1}^I u_i = 1, \quad (4.67)$$

$$G^T u + A^T w = 0, \quad (4.68)$$

$$u \geq 0, w \geq 0. \quad (4.69)$$

Нетрудно видеть, что имеет место следующее утверждение.

Утверждение. Пусть u, w – допустимая точка задачи (4.66)–(4.69).

Положим

$$L(x) = (H_x x + \eta, u) + (Bx - d, w). \quad (4.70)$$

Тогда $L(x) \leq \xi^*(x) \leq \Phi(x)$. Если $u = u(\bar{x})$, $w = w(\bar{x})$ – решение задачи (4.66)–(4.69) для $x = \bar{x}$, то линейная функция $L(x)$ является опорной для $\xi^*(x)$ и $L(\bar{x}) = \xi^*(\bar{x})$.

Таким образом, если задана некоторая нижняя аппроксимация $H(x, y)$ для функции $F(x, y)$, то соотношение (4.70) определяет правило построения аппроксимаций для функции $\Phi(x)$.

3. Пусть задана некоторая кусочно-линейная нижняя аппроксимация $H^q(x^q, y^q)$ для функции $\sum_{s \in S(q)} \varphi^s(x^q, y^q)$ в выражении (4.58) для каждого $q \in V$. Для висячих вершин $H^q(x^q, y^q) \equiv 0$, поскольку $\sum_{s \in S(q)} \varphi^s(x^q, y^q) = 0$.

Будем называть l -аппроксимацией задачи (4.58)–(4.59) следующую задачу: найти

$$\psi^q(x^q) = \min_{y^q} \left\{ c^q y^q + H^q(x^q, y^q) \right\} \quad (4.71)$$

при ограничениях

$$A^q y^q \leq d^q - B^q x^q. \quad (4.72)$$

Легко видеть, что для висячих вершин задачи (4.58)–(4.59) и (4.71)–(4.72) совпадают. Для корневой вершины r дерева (V, E) величина ψ^r есть оценка снизу оптимального значения совокупности задач.

ld -аппроксимацией задачи (4.58)–(4.59) будем называть задачу: найти

$$\xi^q(x^q) = \min_{y^q} \left\{ c^q y^q + \sum_{s \in S(q)} \psi^s(x^q, y^q) \right\} \quad (4.73)$$

при ограничениях

$$A^q y^q \leq d^q - B^q x^q. \quad (4.74)$$

Заметим, что ld -аппроксимация для каждой невисячей вершины есть блочная задача линейного программирования со связывающими

переменными и дополнительными ограничениями (4.74), для висячих вершин задачи (4.73)–(4.74) и (4.71)–(4.72) совпадают.

Обозначим I^q – множество индексов параметров вершины $q \in V$, т.е. $x^q = (x_i, i \in I^q)$. Положим

$$x = (x_i, i \in \bigcup_{q \in V} I^q). \quad (4.75)$$

Пусть некоторым образом определены значения компонент вектора \bar{x} . Тем самым определены значения переменных \bar{y}^q для каждой вершины $q \in V$, не являющейся висячей.

Вектор \bar{x} будем называть допустимым, если для каждой невисячей вершины выполняются ограничения (4.59), а для висячих вершин ограничения (4.59) определяют непустые множества.

Обозначим V^* множество висячих вершин дерева (V, E) . Пусть задан набор неотрицательных чисел $\varepsilon = (\varepsilon^q, q \in V \setminus V^*)$.

Аппроксимации $H^q(x^q, y^q), q \in V$ назовем согласованными, если $\psi^q(x^q) \leq \xi^q(x^q), q \in V$.

Согласованные аппроксимации $H^q(x^q, y^q), q \in V \setminus V^*$ будем называть ε -сбалансированными в допустимой точке \bar{x} , если

$$0 \leq c^q \bar{y}^q + \sum_{s \in S(q)} \psi^s(\bar{x}^q, \bar{y}^q) - \psi^q(\bar{x}^q) \leq \varepsilon^q, q \in V \setminus V^*. \quad (4.76)$$

Заметим, что соотношение (4.76) для фиксированного q есть условие ε^q -оптимальности решения \bar{y}^q блочной задачи (4.73)–(4.74) при $x^q = \bar{x}^q$.

Лемма 4.2. Пусть аппроксимации $H^q(x^q, y^q), q \in V \setminus V^*$ являются ε -сбалансированными в точке \bar{x} . Тогда \bar{x} есть σ -оптимальное решение

совокупности задач (4.58)–(4.59), поставленных в соответствие вершинам дерева (V, E) , где $\sigma = \sum_{q \in V \setminus V^*} \varepsilon^q$.

Доказательство. Для висячих вершин q обозначим $\vartheta^q(\bar{x}^q)$ оптимальное значение задачи линейного программирования (4.58)–(4.59). Для остальных вершин величины $\vartheta^q(\bar{x}^q)$ определим рекуррентно:

$$\vartheta^q(\bar{x}^q) = (c^q, \bar{y}^q) + \sum_{s \in S(q)} \vartheta^s(\bar{x}^q, \bar{y}^q). \quad (4.77)$$

Нетрудно видеть, что ϑ^r есть значение целевой функции на приближенном решении \bar{x} , где r – корень дерева (V, E) . Обозначим V^q множество всех потомков вершины q , $q \in V^q$, $\sigma^q = \sum_{s \in V^q \setminus V^*} \varepsilon^s$. Покажем, что

$$\vartheta^q(\bar{x}^q) - \psi^q(\bar{x}^q) \leq \sigma^q. \quad (4.78)$$

Пусть все потомки вершины q – висячие вершины. Тогда с учетом (4.77) условия (4.76) и (4.78) эквивалентны: \bar{y}^q является ε^q -оптимальным решением задачи (4.73)–(4.74) при $x^q = \bar{x}^q$. Пусть неравенство (4.78) выполняется для всех потомков вершины q . Покажем, что тогда это неравенство справедливо и для вершины q . Учитывая (4.77), (4.76), получаем

$$\vartheta^q(\bar{x}^q) - \psi^q(\bar{x}^q) \leq \varepsilon^q + \sum_{s \in S(q)} \vartheta^s(\bar{x}^q, \bar{y}^q) - \sum_{s \in S(q)} \psi^s(\bar{x}^q, \bar{y}^q) \leq \varepsilon^q + \sum_{s \in S(q)} \sigma^s = \sigma^q.$$

Следовательно, при $q = r$ (r – корень дерева (V, E)) получаем утверждение леммы. ■

4. При использовании схем декомпозиции для решения блочных задач со связывающими переменными возникают определенные трудности в ситуациях, когда при некоторых значениях связывающих переменных не все

подзадачи имеют решения. Для преодоления этих трудностей в [184] предложена специальная регуляризация исходной задачи, заключающаяся во введении вспомогательных переменных для каждого ограничения подзадач. Вспомогательные переменные вводятся в целевую функцию с большими штрафными коэффициентами, что обеспечивает эквивалентность регуляризованной и исходной задач. Для регуляризованной задачи все подзадачи имеют решения при любых значениях связывающих переменных.

В дальнейшем предполагается, что аналогичная регуляризация проведена для совокупности задач (4.58)–(4.59).

Обозначим $\xi^q(x^q, y^q)$ значение целевой функции задачи (4.73)–(4.74) в допустимой точке y^q

$$\xi^q(x^q, y^q) = c^q y^q + \sum_{s \in S(q)} \psi^s(x^q, y^q). \quad (4.79)$$

$P(\varepsilon^q)$ -процедурой решения ld -аппроксимации (блочной задачи (4.73)–(4.74)) для заданного $q \in V$ и фиксированного x^q назовем процедуру поиска ε^q -оптимального решения y^q этой задачи и построения (уточнения) аппроксимации $H^q(x^q, y^q)$, для которых выполняется

$$0 \leq \xi^q(x^q, y^q) - \psi^q(x^q) \leq \varepsilon^q, \quad q \in V. \quad (4.80)$$

Для реализации $P(\varepsilon^q)$ -процедуры в рамках схемы декомпозиции могут использоваться различные подходы – методы негладкой оптимизации [184], методы секущих плоскостей [50, 40, 11] и др. Поскольку подзадачи являются задачами линейного программирования небольшой размерности, они решаются точно. Правила вычисления субградиентов (отсекающих гиперплоскостей), рассмотренные в предыдущих подразделах, упрощаются и сводятся к известным ранее. При этом необходимо находить оптимальные значения двойственных переменных для подзадач блочной задачи. Для

программной реализации более простыми являются схемы, основанные на методах секущих плоскостей и современных программных средствах линейного программирования.

В дальнейшем будем предполагать, что в нашем распоряжении имеется некоторая реализация $P(\varepsilon^q)$ -процедуры.

Пусть задан вектор $\varepsilon = (\varepsilon^q, q \in V)$. Рассмотрим $A(\varepsilon)$ -алгоритм поиска допустимой точки x и построения ε -сбалансированных аппроксимаций $H^q(x^q, y^q), q \in V$ в этой точке. Будем считать заданными некоторые начальные аппроксимации $H_0^q(x^q, y^q), q \in V$.

Пусть для каждой вершины $q \in V$ задана некоторая текущая аппроксимация $H^q(x^q, y^q)$. Рассмотрим связанное поддереву $B = (V_b, E_b)$ дерева (V, E) , содержащее корневую вершину r . Обозначим $x(B)$ совокупность параметров и переменных задач поддерева B .

Поддереву B будем называть ε -сбалансированным для аппроксимаций $H^q(x^q, y^q), q \in V_b$ в допустимой точке $x(B)$, если условие (4.80) выполняется для всех $q \in V_b$.

На каждой итерации рассматриваемого $A(\varepsilon)$ -алгоритма анализируется и модифицируется некоторое текущее ε -сбалансированное поддереву B (если $V_b = \emptyset$, то поддереву B будем называть пустым), текущие точка $x(B)$, аппроксимация $H^q(x^q, y^q)$ для всех $q \in V$.

Считается заданным параметр алгоритма $\gamma, 0 < \gamma < 1$, некоторая начальная точка $\tilde{x}(V, E)$.

Приведем описание $A(\varepsilon)$ -алгоритма.

Шаг 0. Положить $V_b = \emptyset$ (B – пустое поддереву).

Шаг 1. Выбрать вершину $q \in V \setminus V_b$, для непосредственного предка p которой выполняется $p \in V_b$. Если $V_b = \emptyset$, выбрать корневую вершину дерева.

Шаг 2. Если для текущей точки (x^q, y^q) выполняется условие (4.80), добавить вершину q в поддереву B и перейти на шаг 5.

Шаг 3. Применить $P(\gamma\varepsilon^q)$ -процедуру к вершине q , пусть y^q – полученное ε^q -оптимальное решение.

Шаг 4. Исключить непосредственного предка p и всех его потомков из поддерева B .

Шаг 5. Если поддерево B совпадает с исходным деревом (V, E) , закончить процесс вычислений (искомая точка x и ε -сбалансированные аппроксимации построены), иначе перейти на шаг 1.

Приведем некоторые пояснения шагов 3 и 4. Применение $P(\gamma\varepsilon^q)$ -процедуры к вершине q приводит к уточнению аппроксимации $H^q(x^q, y^q)$, что в свою очередь приводит к увеличению разности $\xi^p(x^p, y^p) - \psi^p(x^p)$ для непосредственного предка p вершины q . При нарушении условия (4.80) для вершины p необходимо повторно уточнять аппроксимацию для этой вершины и искать уточненное решение y^p , т.е. применять $P(\gamma\varepsilon^p)$ -процедуру. В случае $0 < \gamma < 1$ при малом увеличении разности $\xi^p(x^p, y^p) - \psi^p(x^p)$ не приходится повторно применять $P(\gamma\varepsilon^p)$ -процедуру.

В качестве начальных аппроксимаций выбираются функции $H_0^q(x^q, y^q) \equiv -\bar{H}$, $q \in V$, где \bar{H} – достаточно большое положительное число.

Алгоритм решения совокупности задач (4.58)–(4.59) для всех вершин дерева (V, E) состоит в итеративном применении $A(\varepsilon)$ -алгоритма. На первой итерации определяется начальное значение вектора $\varepsilon = (\varepsilon^q, q \in V)$. На каждой

итерации уточняются (уменьшаются) значения компонент вектора ε и применяется $A(\varepsilon)$ -алгоритм.

Процесс прекращается при достижении требуемой точности.

Характер поведения $A(\varepsilon)$ -алгоритма зависит от выбора вектора ε и правил его уточнения на каждой итерации. Нетрудно видеть, что при относительно малых ε^q для вершин дерева близких к корню (по сравнению с удаленными от корня вершинами) соответствующие ld -аппроксимационные задачи решаются часто и с большой точностью, при этом используются грубые аппроксимации $H^s(x^s, y^s)$ для их потомков. В случае, когда выбираются относительно малые ε^q для вершин дерева близких к висячим, чаще решаются ld -аппроксимационные задачи для этих вершин и используются более точные аппроксимации $H^s(x^s, y^s)$ для их потомков. В последнем случае точки, относительно которых строятся точные аппроксимации, могут быть достаточно далеки от оптимальных.

5. Описанный алгоритм был реализован средствами языка программирования C++ с использованием библиотеки классов линейного программирования Soplex [65]. Для реализации $P(\varepsilon^q)$ -процедуры был выбран алгоритм Келли.

Начальные значения вектора $\varepsilon = (\varepsilon^q, q \in V)$ задавались следующим образом: $\varepsilon^q = \varepsilon_0 * (\text{Pr}_q)^s$, где ε_0 , s – параметры алгоритма, Pr_q – вероятность реализации вершины q дерева (V, E) .

При итеративном применении $A(\varepsilon)$ -алгоритма величины ε^q изменялись следующим образом $\varepsilon^q = t * \varepsilon^q$, где $0 < t < 1$ – еще один параметр алгоритма.

4.6.2. Стохастическая модель планирования инвестиций в электроэнергетике

В качестве основы для рассматриваемой модели взята постановка задачи из [49].

На каждом этапе планирования t интервала планирования $[0, \dots, T]$ производятся инвестиции в n различных технологий. С каждой технологией i связаны соответствующие значения случайных удельных капитальных затрат α^{it} , случайных удельных эксплуатационных затрат (затрат на обслуживание) γ^{it} .

Введем обозначения:

g_i^t – динамика изменения мощностей технологии i , которые существовали в начале процесса планирования,

X^{it} – новые мощности технологии i , которые планируется создать в момент времени t ,

Δ_i – период введения новой мощности технологии i в эксплуатацию.

Выбор значений $X^{it}, t = 0, \dots, T$ производится в начале интервала планирования при составлении плана инвестиций. В дальнейшем, в начале каждого этапа планирования t , когда известна информация о реализации случайных величин предыдущего этапа, принятых решениях на предыдущих этапах и появляется дополнительная информация о реализации случайных величин текущего этапа, производится корректировка плана. Такие корректировки приводят к определенным потерям, связанным с расторганием инвестиционных контрактов, заключением новых контрактов на менее выгодных условиях.

Положим

W^{it} – корректировка (уменьшение) плана инвестиций для технологии i в момент времени t , $W^{it} \leq 0$,

U^{it} – корректировка (увеличение) плана инвестиций для технологии i в момент времени t , $U^{it} \geq 0$,

β^{it} – случайное удельное сокращение средств при уменьшении (корректировке) плана инвестиций для технологии i в момент времени t (уменьшение затрат равно $\beta^{it}W^{it}$),

χ^{it} – случайные удельные затраты при увеличении (корректировке) плана инвестиций для технологии i в момент времени t (увеличение затрат равно $\chi^{it}U^{it}$).

После введения в эксплуатацию новых мощностей их величина не изменяется. Обозначим G^{it} новые наличные мощности, доступные в момент времени t , тогда

$$G^{it} = \sum_{\tau=0}^{t-\Delta_i} (X^{i\tau} + W^{i\tau} + U^{i\tau}) = G^{i(t-1)} + X^{i(t-\Delta_i)} + W^{i(t-\Delta_i)} + U^{i(t-\Delta_i)}, \quad t \geq \Delta_i,$$

$$G^{it} = 0, \quad t < \Delta_i.$$

Случайные запросы на электроэнергию могут приходиться от различных потребителей, и удовлетворение этих запросов должно быть полностью выполнено на каждом временном этапе. В случае если наличных мощностей не хватает, имеется возможность закупки электроэнергии на внешнем рынке (по более высоким ценам).

Обозначим: d^t – запрос на электроэнергию на этапе t , Z^t величину закупки электроэнергии на внешнем рынке, Y^{it} количество генерируемой электроэнергии по технологии $i = 1, \dots, n$, тогда

$$\sum_{i=1}^n Y^{it} + Z^t \geq d^t, \quad t = 0, \dots, T.$$

Должны также выполняться ограничения на генерируемые мощности

$$Y^{it} \leq G^{it} + g_i^t, \quad i = 1, \dots, n, \quad t = 0, \dots, T.$$

В задаче необходимо найти значения переменных, которые минимизируют математическое ожидание затрат, представленных в виде суммы капитальных и эксплуатационных затрат. Случайными параметрами являются запросы на электроэнергию d^t , значения удельных капитальных и эксплуатационных затрат α^{it} , γ^{it} , удельных затрат на корректировку плана β^{it} , χ^{it} , $i=1, \dots, n$, а также цена λ^t электроэнергии, закупаемой на внешнем рынке.

Обозначим ξ^t векторную случайную переменную с элементами $\{\alpha^{it}, \beta^{it}, \chi^{it}, \gamma^{it}, \lambda^t, d^t, \forall i, t\}$.

Рассматриваемая задача имеет вид: найти

$$\varphi^* = \min_{x,y,z,w,u} E_{\xi} \left[\sum_{t=1}^T \left(\sum_{i=1}^n (\alpha^{it} X^{it} + \beta^{it} W^{it} + \chi^{it} U^{it} + \gamma^{it} Y^{it}) + \lambda^t Z^t \right) \right]$$

при ограничениях

$$\sum_{i=1}^n Y^{it} + Z^t \geq d^t, \quad t=0, \dots, T,$$

$$G^{it} = G^{i(t-1)} + X^{i(t-\Delta_i)} + W^{i(t-\Delta_i)} + U^{i(t-\Delta_i)}, \quad t \geq \Delta_i, i=1, \dots, n,$$

$$G^{it} = 0, \quad t < \Delta_i,$$

$$Y^{it} \leq g_i^t + G^{it}, \quad i=1, \dots, n, t=0, \dots, T,$$

$$X^{it} + W^{it} + U^{it} \geq 0, \quad i=1, \dots, n, t=0, \dots, T,$$

$$X^{it} \geq 0, Y^{it} \geq 0, i=0, \dots, n, Z^t \geq 0, t=0, \dots, T,$$

$$W^{it} \leq 0, U^{it} \geq 0, i=0, \dots, n, t=0, \dots, T.$$

Для описанной модели проводился вычислительный эксперимент для задач двух-, трех-, четырех- и пятиэтапного планирования. Каждая невисячая вершина дерева сценариев имела десять потомков (в некоторых случаях – 20). Случайными величинами считались потребности в электроэнергии.

Предполагалось, что эти величины независимы для каждого этапа и дискретно равномерно распределены на отрезке.

Исходные данные для экземпляров задач формировались в форматах MPS и SMPS с использованием системы моделирования SLP-IOR [50]. Это позволило для вычислительных экспериментов использовать следующие существующие программные средства – Soplex, MSLiP [32].

Поскольку для рассматриваемой задачи в MSLiP выявились программные ошибки, то сравнение проводилось только с пакетом Soplex.

С увеличением размерности увеличивалась разница времени решения задач разработанными средствами и системой Soplex. Для максимальной по размерности задачи пятиэтапного планирования (20 000 сценариев, число ограничений – 211 104, число переменных – 337 773) результаты следующие:

общее время решения Soplex-ом – 925.91 сек. (процессорное время – 651.02 сек.), оптимальное значение – $3.4887500e+03$;

общее время решения разработанными программными средствами ~ 65 сек. (варьируется в зависимости от выбора параметров алгоритма), по ходу вычислительного процесса генерируются приближенные решения, точность которых увеличивается.

Поскольку по ходу решения исходной задачи используются и уточняются кусочно-линейные аппроксимации задач (4.58)–(4.59), важной характеристикой алгоритма является число строк, добавленных к исходной задаче. При этом добавленные строки, которые длительное время остаются неактивными в решениях задач (4.71)–(4.72), удаляются из задачи. Финальное число строк равно 233 286 (исходное – 211 104) при использовании параметров алгоритма $p = 0.9$, $q = 0.7$. Такое относительно небольшое количество добавленных строк объясняется тем, что такие строки добавляются только к задачам, соответствующим невисячим вершинам дерева задач.

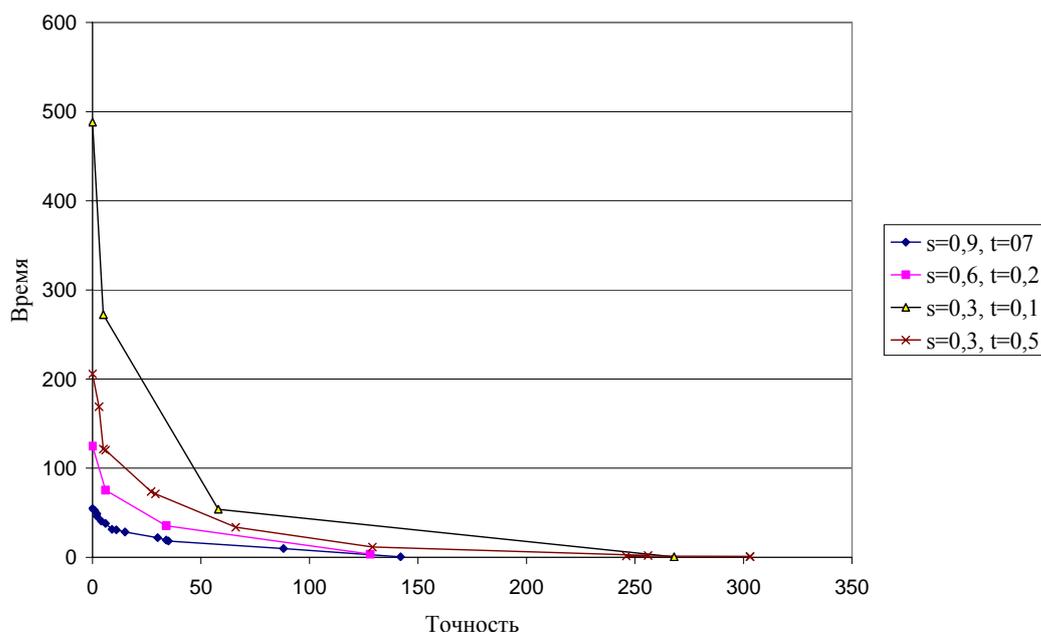


Рис. 4.1. Зависимость времени (сек.) поиска приближенного решения от точности

Динамика вычислительного процесса в зависимости от выбора параметров алгоритма приведена на рисунке и подтверждает предположение о существенной зависимости эффективности алгоритма от согласования точности решения блочных задач (4.73)–(4.74) в ходе вычислений.

4.6.3. Задача перспективного планирования в электроэнергетике с учетом суточных неравномерностей потребления электроэнергии

Рассматриваются энергосистемы, которые описываются совокупностью электрогенерирующих и управляющих объектов. В качестве управляющих используются энергоаккумулирующие объекты (ГАЭС), маневренные энергоблоки пылеугольных и газомазутных ТЭС, газопаротурбинные установки, а также электротермические потребители-регуляторы, исследование возможностей которых интенсивно ведется в последнее время [80, 81, 99].

Существенной особенностью рассматриваемых задач является необходимость учитывать суточные неравномерности потребления электроэнергии. Обычно учет суточных графиков потребления

электроэнергии выполняется при решении задач оперативного планирования-управления. Для этих задач разработаны различные математические модели [138], но они являются слишком сложными для использования в задачах перспективного планирования.

Предложена упрощенная математическая модель энергосистемы, позволяющая при решении задач перспективного планирования учитывать суточные неравномерности потребления электроэнергии. Эта модель является задачей многоэтапного стохастического программирования, обладает характерными особенностями и может в дальнейшем уточняться и обобщаться. Для решения таких задач используются различные схемы декомпозиции [117,184].

Приводятся результаты вычислительных экспериментов по применению разработанных программных средств [117] для решения специальных классов задач, которые описываются предложенными математическими моделями.

4.6.3.1. Особенности функционирования энергосистемы и задач перспективного планирования

В энергосистеме в целом должен поддерживаться баланс производства и потребления электроэнергии с учетом суточных и сезонных неравномерностей потребления (нагрузок), возможностей аварийного отключения энергоблоков. Регулирование энергосистемы осуществляется за счет изменения мощностей включенных энергоблоков, включения и отключения некоторых энергоблоков, использования энергоаккумулирующих объектов и электротермических потребителей-регуляторов. Расход топлива и количество произведенной электроэнергии зависят от режима (мощности) эксплуатации каждого энергоблока. В состоянии горячего резерва топливо расходуется, а электроэнергия не производится.

Для объектов энергосистемы важной характеристикой является скорость изменения используемой мощности, которая существенно зависит от типа объекта:

- атомные энергоблоки работают при постоянной нагрузке,
- для пылеугольных ТЭС скорость изменения мощности не должна превышать 2,5% за минуту,
- для газомазутных ТЭС – 8% за минуту,
- для ГЭС и энергоаккумулирующих объектов – 1,5–2,5% за секунду,
- для электротермических потребителей-регуляторов – до 500 МВт за секунду.

Эффективность функционирования энергосистемы определяется рациональным оперативным управлением и перспективным планированием работ по поддержанию существующих мощностей и созданию новых.

Затраты на генерирование электроэнергии (ЭЭ) существенно зависят от цен на топливо, эксплуатационных затрат, капитальных затрат на поддержание существующих мощностей и создание новых. При перспективном планировании эти характеристики, а также объемы потребления ЭЭ, являются случайными величинами.

Для планового периода, состоящего из нескольких этапов (интервалов), эта информация считается известной для первого интервала, а для последующих интервалов могут использоваться только их совместные распределения.

Приближенное решение таких задач может быть получено на пути использования сценарного подхода, в котором для представления совместного распределения случайных величин рассматривается конечное дерево возможных реализаций (сценариев) по интервалам планирования и каждой вершине этого дерева поставлена в соответствие условная вероятность реализации данного сценария. Рассматриваемая задача формулируется относительно дерева сценариев и называется

детерминированным аналогом исходной задачи. Такой подход широко используется в настоящее время.

4.6.3.2. Принятые упрощения

Энергосистему будем описывать совокупностью электрогенерирующих и управляющих объектов. Каждый объект энергосистемы реализует одну из возможных технологий. Существует три группы технологий – электрогенерирующие (энергоблоки атомных станций, ГЭС, газомазутные энергоблоки, пылеугольные энергоблоки, газопаротурбинные установки), энергоаккумулирующие и электротермические потребители-регуляторы. Количество используемых технологий фиксировано для каждой группы.

Под перспективным планированием в энергетике понимается планирование работ по поддержанию (ремонту и реконструкции) существующих объектов энергосистемы и созданию новых объектов. Плановый период состоит из нескольких интервалов планирования. В качестве интервала планирования может рассматриваться год, квартал или другая величина.

Для упрощения математической модели будем характеризовать каждую технологию суммарной мощностью ее уже существующих объектов и графиком изменения (выработки ресурса) этой мощности по интервалам планового периода. Будем считать, что работы по поддержанию существующих объектов и созданию новых объектов каждой технологии приводят к созданию новых мощностей, которые уже не изменяются в течение планового периода.

Существующие и новые мощности каждой технологии энергосистемы используются в течение всего планового периода для обеспечения требуемого потребления электроэнергии, которое задается суточными графиками потребления ЭЭ. При заданных суточных графиках решается задача определения оптимальных почасовых нагрузок (мощностей) всех объектов энергосистемы в течение суток (задача оперативного

планирования). Такие задачи рассматривались в разных работах (см., например, [138]).

В дальнейшем предполагается, что для каждого интервала планирования суточные почасовые графики потребления ЭЭ одинаковы и описываются кусочно-линейными функциями специального вида. Количество суток в интервале планирования фиксировано.

Поскольку скорости изменения режима генерирования ЭЭ электрогенерирующих объектов ограничены, используются энергоаккумулирующие объекты и электротермические потребители-регуляторы для сглаживания заданного графика потребления ЭЭ. На рис. 4.2 приведены примеры графиков $p(t)$ потребления и $g(t)$ генерирования ЭЭ. В интервале $(0, \zeta)$ генерируется избыток ЭЭ, что позволяет уменьшить скорости изменения режима генерирования ЭЭ. Избыток ЭЭ используется энергоаккумулирующими объектами для накопления ЭЭ и электротермическими потребителями-регуляторами. Предполагается, что скорости изменения режимов этих объектов неограничены (ограничены мощности). В интервале (ζ, τ_3) недостаток ЭЭ компенсируется ЭЭ, накопленной энергоаккумулирующими объектами в интервале $(0, \zeta)$. В интервале (τ_3, T) выполняется $p(t) = g(t)$.

Графики потребления $p(t)$ и генерирования $g(t)$ электроэнергии будем задавать параметрами $(p_{\max}, p_{\min}, t_1, t_2, t_3)$ и $(g_{\max}, g_{\min}, \tau_1, \tau_2, \tau_3)$, где

t_1, t_2, t_3 (τ_1, τ_2, τ_3) – точки излома графика $p(t)$ (соответственно графика $g(t)$),

p_{\max}, p_{\min} – максимальный и минимальный уровни потребления ЭЭ,

g_{\max}, g_{\min} – максимальный и минимальный уровни генерирования ЭЭ,

$p_{\max} \geq p_{\min} \geq 0, 0 < t_1 \leq t_2, t_3 = t_1 + t_2 \leq T, t_1 \leq \tau_1 \leq \tau_2 \leq t_2, \tau_3 = \tau_1 + \tau_2 \geq t_3,$

$g_{\max} = p_{\max}, g_{\min} \geq p_{\min},$

T – длительность суточного интервала планирования.

Скорости уменьшения и увеличения потребления ЭЭ считаются одинаковыми, т.е. $t_1 = t_3 - t_2$. Аналогично для графика $g(t)$ – $\tau_1 = \tau_3 - \tau_2$.

Параметры $(p_{\max}, p_{\min}, t_1, t_2, t_3)$ являются внешними для рассматриваемых задач, параметры $(g_{\max}, g_{\min}, \tau_1, \tau_2, \tau_3)$ определяются при решении задач оперативного планирования.

Рассмотрим графики потребления и генерирования электроэнергии. Обозначим

b – скорость уменьшения (увеличения) потребления ЭЭ,

z – скорость уменьшения (увеличения) генерирования ЭЭ,

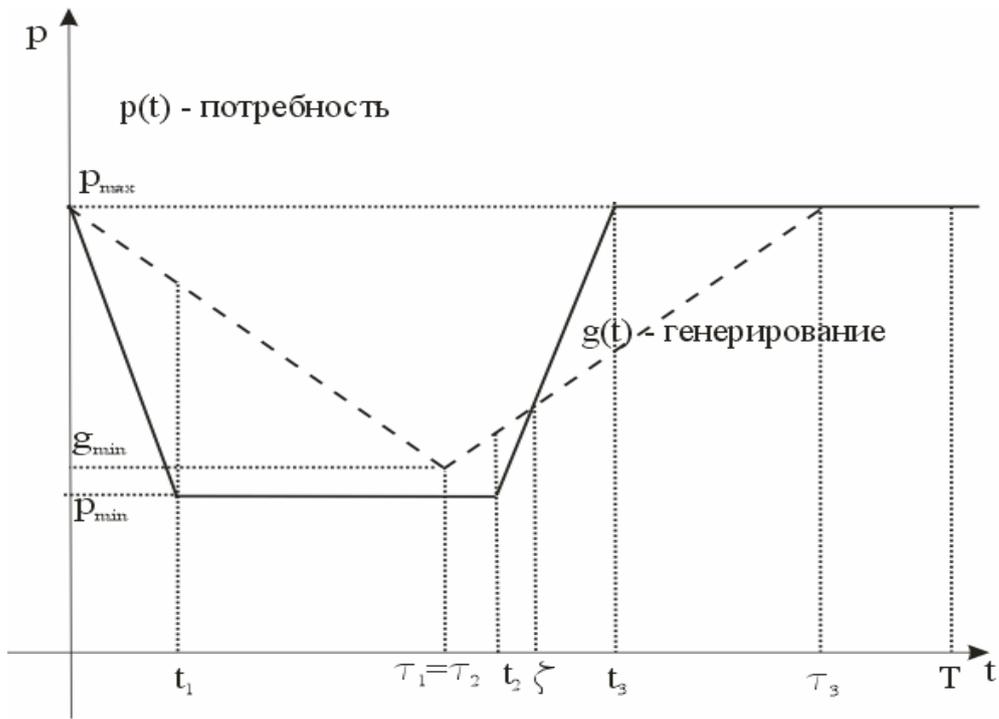
$$b = (p_{\max} - p_{\min})/t_1,$$

$$z = (g_{\max} - g_{\min})/\tau_1, \quad 0 \leq z \leq b.$$

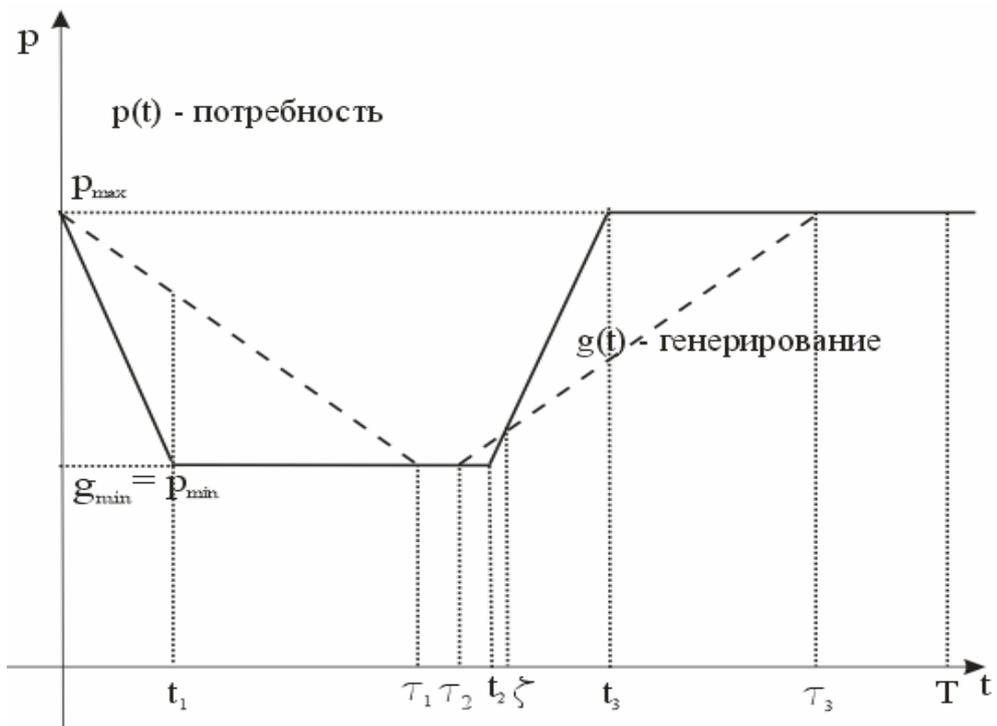
Значение g_{\min} рассчитывается, $g_{\min} = \max\{p_{\min}, p_{\max} - z t_3/2\}$. Если $g_{\min} > p_{\min}$, то $\tau_1 = \tau_2 = \tau_3/2$.

В случае, когда в энергосистеме используются энергоаккумулирующие объекты, выполняется $\tau_3 > t_3$. В противном случае $\tau_3 = t_3$.

Пусть в энергосистеме используются энергоаккумулирующие объекты. Тогда $\tau_3 > t_3$. Обозначим $x = t_3 - \zeta$, $y = \tau_3 - t_3$. Будем рассматривать величины x, y как независимые переменные, полностью описывающие график $g(t)$ при заданном графике $p(t)$. Для x, y должны выполняться условия $0 \leq x \leq t_3 - t_2$, $0 < y \leq T - t_3$. Можно показать, что $z = b(t_3 - \zeta)/(\tau_3 - \zeta) = bx/(x + y)$ и выразить все параметры графика $g(t)$ через величины x, y .



а)



б)

Рис. 4.2. Графики потребления (сплошная линия) $p(t)$ электроэнергии в энергосистеме и генерирования (штрихпунктирная линия) $g(t)$ в группе электрогенерирующих объектов

В случае, когда энергоаккумулирующие объекты не используются, имеет место $\tau_3 = t_3$, график $g(t)$ однозначно определяется значением z – скоростью уменьшения (увеличения) генерирования ЭЭ, $\tau_1 = \min\{t_3/2, (p_{\max} - p_{\min})/z\}$, $\tau_2 = t_3 - \tau_1$.

Примеры графика $u(t)$ компенсации дисбаланса потребления и генерирования ЭЭ приведены на рис. 4.3.

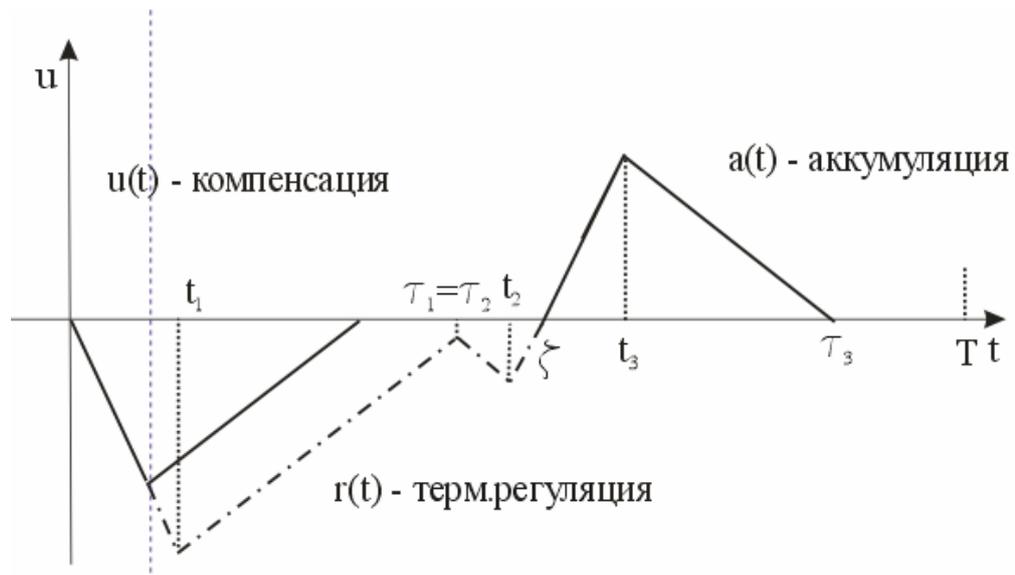


Рис. 4.3. Графики (вариант а) распределения компенсации $u(t) = p(t) - g(t)$ между энергоаккумулирующими объектами $a(t)$ (сплошная линия) и электротермическими потребителями-регуляторами $r(t)$ (штрихпунктиром обозначена сумма $a(t) + r(t) = u(t)$).

График $u(t) = p(t) - g(t)$ определяется значениями функции $u(t)$ в точках $0, t_1, \tau_1, \tau_2, t_2, t_3, \tau_3$:

$$\tau_1 = \tau_3 - \tau_2,$$

$$u(0) = 0,$$

$$u(t_1) = p_{\min} - p_{\max} + z t_1,$$

$$u(\tau_1) = u(\tau_2) = p_{\min} - g_{\min},$$

$$u(t_2) = p_{\min} - g_{\min} - z(t_2 - \tau_2),$$

$$u(t_3) = p_{\max} - g_{\min} - z(t_3 - \tau_2),$$

$$u(\tau_3) = 0.$$

Если энергоаккумулирующие объекты не используются, то $a(t) = 0$, $u(t_1) = u(t_2)$, $t_3 = \tau_3$ и z является независимой переменной, определяющей график $u(t)$.

В дальнейшем будем рассматривать случай, когда энергоаккумулирующие объекты не используются. Тогда график $r(t)$ режима работы электротермических потребителей-регуляторов определяется соотношением $r(t) = p(t) - g(t)$, а также имеет место $\tau_3 = t_3$, $\tau_2 = t_3 - \tau_1$.

Ограничение по мощности электротермических потребителей-регуляторов имеет вид

$$\max \{|r(t)| : 0 \leq t \leq t_3\} = g(t_1) - p(t_1) = p_{\max} - z t_1 - p_{\min} \leq R_r, \quad (4.81)$$

где R_r – максимальная мощность.

Функционирование электрогенерирующих объектов i -ой технологии в каждом интервале планирования будем описывать суммарной мощностью r_i включенных объектов (не изменяется в интервале планирования) и графиком $g_i(t)$ генерирования ЭЭ, $g_i(t) \leq r_i$, $r_i \leq R_i$, где R_i – суммарная производственная мощность всех объектов i -ой технологии.

Пусть в некотором интервале планирования объектами i -ой технологии сгенерирован объем $E_i = \int_0^T g_i(t) dt$ электроэнергии. Затраты Z_i на генерирование ЭЭ будем определять по упрощенному правилу $Z_i = c_i E_i + c_i^g r_i$, где c_i – удельные затраты на генерирование ЭЭ, c_i^g – удельные затраты на поддержание мощностей во включенном состоянии.

Пусть $(g_{\max}, g_{\min}, \tau_1, \tau_2, \tau_3)$ – параметры, задающие суммарный график $g(t)$, тогда график $g_i(t)$ определяется набором $(g_{\max}^i, g_{\min}^i, \tau_1, \tau_2, \tau_3)$, где значения g_{\max}^i, g_{\min}^i определяются при распределении нагрузок между разными технологиями.

4.6.3.3. Обозначения

Положим

$$\langle a, b \rangle = \sum_{i=1}^n a_i b_i \text{ – скалярное произведение векторов } a, b \in R^n,$$

I – совокупность индексов всех ресурсов технологий энергосистемы (энергоаккумулирующие технологии характеризуются двумя ресурсами – преобразования и накопления энергии, остальные технологии – одним ресурсом),

$\{1, \dots, m\}$ – совокупность индексов ресурсов генерирующих технологий,
 $\{1, \dots, m\} \subset I$,

$(0, 1, \dots, S)$ – плановый период, $s, s = 0, 1, \dots, S$ – интервалы планирования.

Рассмотрим дерево сценариев (V, E) для задачи перспективного планирования. Обозначим q^0 корневую вершину дерева сценариев. Каждой вершине $q \in V$ соответствует определенный сценарий – последовательность реализаций случайных величин, определяющих функционирование энергосистемы и условия создания новых объектов. Сценарий для вершины q задается совокупностью вершин дерева (V, E) , лежащих на пути из корня q^0 в вершину q .

Рангом $\tau(q)$ вершины $q \in V$ будем называть длину пути плюс 1 от корня q^0 до вершины q . Вершины, для которых $\tau(q) = s$, соответствуют интервалу планирования s . Для корня q^0 имеем $\tau(q^0) = 1$.

Обозначим

ρ_i^s – значение на интервале планирования s мощности ресурса i , которая существовала до начала планового периода, $s=0,1,\dots,S$, $i \in I$,
 $\rho^s = \{\rho_i^s, i \in I\}$,

η_i^q – новые мощности ресурса i , создаваемые в соответствии со сценарием, определяемым вершиной q , $\eta^q = (\eta_i^q, i \in I)$,

$X^q = \{X_i^q, i \in I\}$ – объемы работ по увеличению мощностей энергосистемы в вершине q ,

$\alpha^q = (\alpha_i^q, i \in I)$ – удельные затраты при выполнении работ X^q , общие затраты на выполнение этих работ равны $\langle \alpha^q, X^q \rangle$.

Q_{\max}^q – объемы возможных инвестиций.

Суммарные производственные мощности $R^q = (R_i^q, i \in I)$, доступные в вершине q , используются для обеспечения суточного графика потребления ЭЭ, соответствующего этой вершине. Загрузка мощностей энергосистемы для обеспечения суточного графика потребления ЭЭ определяется решением упрощенной задачи оперативного планирования для вершины q .

Объекты i -ой генерирующей технологии характеризуются максимальной скоростью α_i изменения нагрузки (мощности), $i=1,\dots,m$. Эти величины одинаковы для всех вершин дерева сценариев.

Каждой вершине q поставлены в соответствие:

$p_{\max}^q, p_{\min}^q, t_1^q, t_2^q, t_3^q$ – параметры (реализация случайных величин), задающие суточный график потребления ЭЭ,

c_i^q – удельные затраты (цена топлива) на производство ЭЭ i -ой технологии,

c_i^{qg} – удельные затраты на поддержку мощности включенных объектов i -ой технологии (\sim горячий резерв),

c_r^q – удельная цена продукта электротермических потребителей-регуляторов при потреблении единицы ЭЭ.

Переменными, определяющими загрузку мощностей энергосистемы для обеспечения суточного графика потребления ЭЭ для вершины q , являются:

$\tau_1^q, \tau_2^q, \tau_3^q, g_{\min}^q, g_{\max}^q$ – параметры суммарного графика генерирования ЭЭ,

z^q – скорость уменьшения (увеличения) генерирования ЭЭ,

$\tau_1^q, \tau_2^q, \tau_3^q, g_{\min}^{qi}, g_{\max}^{qi}$ – параметры графика генерирования ЭЭ i -ой технологией, $i = 1, \dots, m$, (точки излома графика генерирования ЭЭ i -ой технологией совпадают с точками излома суммарного графика),

r_i^q – производственная мощность включенных объектов i -ой технологии,

R_i^q – производственная мощность всех объектов i -ой технологии,

R_r – максимальная мощность электротермических потребителей-регуляторов.

4.6.3.4. Математическая модель задачи перспективного планирования

При перспективном планировании решается задача минимизации суммарных затрат по обеспечению суточных графиков потребления ЭЭ для всех вершин дерева сценариев и затрат по увеличению мощностей объектов энергосистемы. Реализации случайных величин, определяющих функционирование энергосистемы и условия создания новых объектов, а также условные вероятности этих реализаций считаются заданными для всех вершин дерева V .

Пусть вершина q является непосредственным потомком вершины p . Рассмотрим оптимизационную задачу P_q , соответствующую вершине q . Обозначим $\varphi^q(\eta^p)$ оптимальное значение задачи P_q (минимальные ожидаемые затраты для вершины q и совокупности сценариев, являющихся продолжением вершины q).

Переменными задачи P_q , поставленной в соответствие вершине q , являются новые мощности η^q и объемы работ X^q , а также переменные, определяющие загрузку мощностей энергосистемы для обеспечения суточного графика потребления ЭЭ вершины q . Параметрами этой задачи являются значения переменных η^p задачи P_p , соответствующей вершине-предку p .

Обозначим μ количество суток в рассматриваемых интервалах планирования, $S(q)$ – множество непосредственных потомков вершины q . Задача P_q может быть представлена в виде: найти

$$\varphi^q(\eta^p) = \min \left\{ \mu \Phi^q + \langle \alpha^q, X^q \rangle + \sum_{k \in S(q)} p_k \varphi^k(\eta^q, X) \right\} \quad (4.82)$$

при ограничениях

$$\langle \alpha^q, X^q \rangle \leq Q_{\max}^q, \quad (4.83)$$

$$X^q \geq 0, \quad (4.84)$$

$$\eta^q = \eta^p + X^q, \quad (4.85)$$

$$R^q = \eta^p + \rho^{\tau(q)}, \quad (4.86)$$

$$\Phi^q = \sum_{i=1}^m \left(c_i^q E_i^q + c_i^{qs} r_i^q \right) - c_r^q \left(E^q - E_{\text{ном}p}^q \right), \quad (4.87)$$

$$E_i^q = g_{\min}^{qi} t_3^q + (g_{\max}^{qi} - g_{\min}^{qi}) \tau_1^q + g_{\max}^{qi} (T - t_3^q), \quad i = 1, \dots, m, \quad (4.88)$$

$$E^q = g_{\min}^q t_3^q + (g_{\max}^q - g_{\min}^q) \tau_1^q + g_{\max}^q (T - t_3^q), \quad (4.89)$$

$$\sum_{i=1}^m g_{\min}^{qi} = g_{\min}^q, \quad (4.90)$$

$$\sum_{i=1}^m g_{\max}^{qi} = g_{\max}^q, \quad (4.91)$$

$$0 \leq g_{\min}^{qi} \leq g_{\max}^{qi} \leq r_i^q, \quad i = 1, \dots, m, \quad (4.92)$$

$$r_i^q \leq R_i^q, \quad i = 1, \dots, m, \quad (4.93)$$

$$g_{\max}^{qi} - g_{\min}^{qi} \leq \alpha_i^q r_i^q \tau_1^q, \quad i = 1, \dots, m, \quad (4.94)$$

$$p_{\max}^q - z^q t_1^q - p_{\min}^q \leq R_r^q, \quad (4.95)$$

$$g_{\min}^q = \max \left\{ p_{\min}^q, p_{\max}^q - z^q t_3^q / 2 \right\}, \quad (4.96)$$

$$\tau_1^q = \min \left\{ t_3^q / 2, \frac{p_{\max}^q - p_{\min}^q}{z^q} \right\}, \quad (4.97)$$

$$0 \leq z^q \leq \frac{p_{\max}^q - p_{\min}^q}{t_1^q}, \quad (4.98)$$

$$g_{\max}^q = p_{\max}^q. \quad (4.99)$$

Здесь:

$$\Phi^q = \sum_{i=1}^m (c_i^q E_i^q + c_i^{qg} r_i^q) - c_r^q (E^q - E_{номр}^q) \text{ затраты на генерирование ЭЭ (в}$$

том числе избыточной, которая используется электротермическими потребителями-регуляторами) для обеспечения суточных графиков потребления ЭЭ.

Величина $c_r^q (E^q - E_{номр}^q)$ есть стоимость продукции, произведенной электротермическими потребителями-регуляторами,

E_i^q – объем ЭЭ, сгенерированной объектами i -ой технологии,

$c_i^{qs} r_i^q$ – затраты на поддержку мощности r_i^q включенных объектов

i -ой технологии,

E^q – суммарный объем сгенерированной ЭЭ,

$E_{номр}^q$ – объем суточной потребляемой ЭЭ,

$$E_{номр}^q = (p_{\max}^q + p_{\min}^q) t_1^q + p_{\min}^q (t_2^q - t_1^q) + p_{\max}^q (T - t_3^q),$$

ограничения (4.90), (4.91) определяют распределение нагрузок по разным технологиям генерирования ЭЭ,

ограничения (4.92) означают, что нагрузки не должны превышать мощность включенных объектов,

(4.93) означают, что мощность включенных объектов не должны превышать производственной мощности всех объектов для каждой технологии,

(4.94) определяют ограничения по скорости изменения нагрузок,

(4.95) определяют ограничения по мощности электротермических потребителей-регуляторов,

(4.96), (4.97) определяют зависимость переменных g_{\min}^q , τ_1^q от независимой переменной z^q .

Сформулированная задача линейна по всем переменным, кроме переменных, определяющих графики генерирования ЭЭ. Величина τ_1^q как функция от z^q невыпуклая и невыпуклые ограничения (4.90) (если их выразить через z^q), хотя величина g_{\min}^q как функция от z^q выпукла. Также невыпуклые ограничения (4.94), а величины E^q, E_i^q являются билинейными функциями от переменных, определяющих графики генерирования ЭЭ.

Для параметров η^p задачи P_q для корневой вершины q^0 дерева сценариев имеет место $\eta^p = 0$.

4.6.3.5. Частные случаи и возможные методы решения задач планирования

Рассмотрим задачу (4.82)–(4.99) при дополнительных ограничениях.

1. Пусть $0 \leq z^q \leq 2(p_{\max}^q - p_{\min}^q)/t_3^q$. Тогда $\tau_1^q = t_3^q/2$, $g_{\min}^q = p_{\max}^q - z^q t_3^q/2$ – линейная функция, и задача (4.82)–(4.99) является задачей линейного программирования.

2. Пусть $2(p_{\max}^q - p_{\min}^q)/t_3^q \leq z^q \leq (p_{\max}^q - p_{\min}^q)/t_1^q$. Тогда $g_{\min}^q = p_{\min}^q$, $\tau_1^q = (p_{\max}^q - p_{\min}^q)/z^q$ – выпуклая функция. Возьмем величину τ_1^q в качестве независимой переменной. Ограничение (4.95) приводится к виду $(p_{\max}^q - p_{\min}^q)(\tau_1^q - t_1^q) \leq R_r^q \tau_1^q$ и является выпуклым, ограничение (4.94) – также нелинейно, но выпукло, величины E_i^q (соотношения (4.88)) есть билинейные функции от переменных задачи g_{\max}^{qi} , g_{\min}^{qi} , τ_1^q . Таким образом, задачу (4.82)–(4.99) можно представить в виде задачи с билинейной целевой функцией при выпуклых ограничениях.

В первом случае, $0 \leq z^q \leq 2(p_{\max}^q - p_{\min}^q)/t_3^q$, могут использоваться схемы декомпозиции для вложенных блочных задач линейного программирования со связывающими переменными [117]. Во втором случае должны разрабатываться специальные средства решения задачи, либо можно ограничиться расчетами (решением задач линейного программирования) при некоторых фиксированных значениях величин τ_1^q .

4.6.3.6. Результаты вычислительных экспериментов

Для предложенных математических моделей исследовались возможности применения ранее разработанных программных средств [117]. Тестирование проводилось на задачах, которые рассматривались в пункте 4.6.3, при дополнительных ограничениях $0 \leq z^q \leq 2(p_{\max}^q - p_{\min}^q)/t_3^q$. Тестовые задачи генерировались случайным образом. В качестве случайных

рассматривались графики потребления ЭЭ (величины p_{\max}^q, p_{\min}^q), остальные параметры считались фиксированными.

При применении схем декомпозиции существенные проблемы возникают в ситуациях, когда подзадачи не имеют допустимых решений при некоторых значениях связывающих переменных. Для исключения таких ситуаций в состав энергогенерирующих объектов включались объекты с неограниченными мощностями и высокими ценами использования, что соответствует закупке ЭЭ на внешнем рынке. Также считались неограниченными мощности электротермических потребителей-регуляторов и не учитывались ограничения по объемам инвестиций.

Рассматривались задачи двух-, трех-, четырех- и пятиэтапного планирования. При формировании дерева сценариев для каждой невисячей вершины генерировалось от пяти до двадцати потомков. Все случайные величины считались независимыми и равномерно распределенными на отрезке.

В ходе вычислительных экспериментов проводилось сравнение разработанных программных средств с известным пакетом программ Soplex (<http://soplex.zib.de/>). С увеличением размерности увеличивалась разница времени решения задач разработанными средствами и системой Soplex. Для максимальной по размерности задачи пятиэтапного планирования (20 000 сценариев, число ограничений – 211 104, число переменных – 337 773) общее время решения Soplex-ом – 925.91 сек., время решения задачи разработанными программными средствами ~ 65 сек. (варьируется в зависимости от выбора параметров алгоритма), по ходу вычислительного процесса генерируются приближенные решения, точность которых увеличивается.

4.7. Выводы

В подразделе 4.1 сформулированы условия, при выполнении которых может быть вычислен ε -субградиент оптимального значения подзадачи при

заданных значениях связывающих переменных. Для таких вычислений должна быть задана допустимая точка подзадачи (приближенное решение) и выполняться определенные соотношения для ε -субградиентов функций подзадач в заданной точке.

Полученные результаты используются при вычислении ε -субградиента оптимального значения на основе кусочно-линейной аппроксимации подзадачи, полученной в процессе решения этой подзадачи. Допустимые точки подзадач генерируются по ходу решения, процесс решения завершается при достижении заданной точности.

Существенные проблемы возникают, если подзадачи не имеют решений при некоторых значениях связывающих переменных. Для преодоления этих проблем в подразделе 4.2 рассматриваются различные приемы регуляризации в схемах декомпозиции. Все эти подходы основаны на использовании точных штрафных функций.

Раздел 4.3 посвящен вопросам определения значений штрафных коэффициентов при использовании точных штрафных функций в схемах декомпозиции. Детализируются подходы, изложенные в разделе 3, применительно к схемам декомпозиции. Для задач с ограниченными областями определения функций предлагается (подраздел 4.4) использовать методы выпуклого продолжения, описанные в разделе 2. Функции подзадач должны быть липшицевыми.

В подразделе 4.5 описываются особенности программной реализации схемы декомпозиции для специального класса задач, приводятся результаты вычислительных экспериментов, обсуждаются возможности повышения эффективности разрабатываемых алгоритмов.

Подраздел 4.6 посвящен естественному обобщению блочных задач со связывающими переменными – задачам линейного программирования с вложенной структурой. Такие задачи имеют место, если каждая подзадача в свою очередь является блочной задачей. Эти задачи возникают в прикладных

областях, связанных с моделированием в экономике [142], многоэтапным стохастическим программированием [50, 40, 11] и др. В подразделе описываются схемы декомпозиции для задач такого вида, особенности программной реализации и результаты вычислительных экспериментов для задач планирования инвестиций в электроэнергетике.

РАЗДЕЛ 5. ИСПОЛЬЗОВАНИЕ НЕГЛАДКОЙ ОПТИМИЗАЦИИ В ЗАДАЧАХ ПОСТРОЕНИЯ АЛГОРИТМОВ КЛАССИФИКАЦИИ

Задачам построения линейных классификаторов посвящено большое количество работ (см., например, [62, 74, 72, 137, 174]). Наиболее исследованными являются классификаторы для разделения двух классов. Наиболее сложными являются задачи построения классификаторов для линейно неразделимых обучающих выборок.

В разделе приводится описание новых подходов к построению классификаторов, проводится анализ и теоретическое сравнение предлагаемых подходов с существующими. Для решения задач построения классификаторов предлагается использовать методы негладкой оптимизации. Описываются результаты вычислительных экспериментов.

5.1. Задачи построения линейных классификаторов

Рассматриваются различные формулировки задач построения классификаторов, определяется понятие и анализируются условия линейной разделимости множеств. Для линейно разделимых множеств задачи построения оптимальных линейных классификаторов решаются достаточно эффективно. Понятие оптимальности для двух линейно разделимых множеств имеет простой геометрический смысл – оптимальный классификатор определяет полосу максимальной ширины, разделяющую эти множества.

В случае линейно неразделимых множеств одним из критериев выбора классификатора является минимизация эмпирического риска. Рассматривается новая частично целочисленная формулировка задачи минимизация эмпирического риска. Проводится сравнение предложенного подхода с существующими.

Другим критерием выбора классификатора в случае линейно неразделимых множеств является минимизация ширины полосы, в которой

классификация производится неправильно. В общем случае такой подход приводит к невыпуклым многоэкстремальным задачам математического программирования, для которых неизвестны эффективные методы поиска глобального минимума. Рассматриваются случаи полиномиальной разрешимости таких задач.

Описываются особенности использования методов негладкой оптимизации для решения сформулированных задач.

5.1.1. Формулировка задачи и линейная разделимость

Считается заданной совокупность функций $f_i(x, W^i)$, где $x \in R^n$ – вектор признаков, $W^i \in R^{l_i}$ – вектор параметров, $i = 1, \dots, m$. Обозначим $W = (W^1, \dots, W^m)$, $W \in R^L$, где $L = \sum_{i=1}^m l_i$.

Функции $f_i(x, W^i)$, $i = 1, \dots, m$, будем называть дискриминантными функциями. Дискриминантным классификатором для случая m классов назовем отображение следующего вида

$$a(x, W) = \arg \max_i \{ f_i(x, W^i) : i = 1, \dots, m \}, \quad x \in R^n, \quad W \in R^L. \quad (5.1)$$

В данном подразделе будет предполагаться, что $f_i(x, W^i)$ – линейные функции, $f_i(x, W^i) = (w^i, x) + w_0^i$, $i = 1, \dots, m$, $W^i = (w_0^i, w^i) \in R^{n+1}$. Классификатор $a(x, W)$, определяемый линейными дискриминантными функциями, будем называть линейным классификатором.

В случае $m = 2$ линейный классификатор описывается одной линейной функцией $l(x, w, w_0) = f_1(x, W^1) - f_2(x, W^2) = (w, x) + w_0$, где $w = w^1 - w^2$, $w_0 = w_0^1 - w_0^2$,

$$a(x, W) = \begin{cases} 1, & \text{если } (w, x) + w_0 \geq 0, \\ 2, & \text{если } (w, x) + w_0 < 0. \end{cases} \quad (5.2)$$

Классификатор $a(x, W)$ определяет разбиение пространства R^n на подмножества $K_i = \{x \in R^n : a(x, W) = i\}$, $i = 1, \dots, m$.

Класс линейных классификаторов является наиболее исследованным. В литературе рассматривается, в основном, случай $m = 2$ [72, 137, 174].

Считается заданной совокупность конечных непересекающихся множеств $\Omega_i, i = 1, \dots, m$. Будем говорить, что классификатор $a(x, W)$ правильно разделяет точки из $\Omega_i, i = 1, \dots, m$, если $a(x, W) = i$, для всех $x \in \Omega_i, i = 1, \dots, m$.

Множества $\Omega_i, i = 1, \dots, m$, называются линейно разделимыми, если существует линейный классификатор, правильно разделяющий точки из этих множеств.

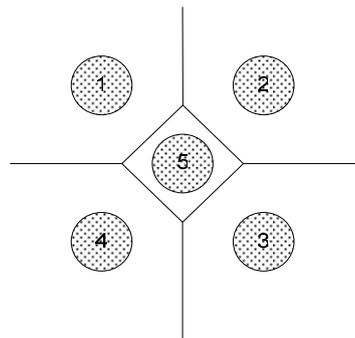


Рис. 5.1.

На рис. 5.1 приведен пример множеств в R^2 и деление плоскости на классы линейным классификатором. Множества 1, 2, ..., 5 – круги радиуса 1, размещенные соответственно в точках $(-2, 2)$, $(2, 2)$, $(2, -2)$, $(-2, -2)$, $(0, 0)$. Линейные функции $f_i(x, W^i) = (w^i, x) + w_0^i$: $w^1 = (-1, 1)$, $w^2 = (1, 1)$, $w^3 = (1, -1)$, $w^4 = (-1, -1)$, $w^5 = (0, 0)$; $w_0^i = 0, i = 1, \dots, 4, w_0^5 = 2$.

Известно, что два множества линейно разделимы, если выпуклые оболочки этих множеств не пересекаются (см., например, [137]). В случае $m > 2$ это условие недостаточно. На рис. 5.2 приведен контрпример. Аналогичный контрпример приведен в [147].

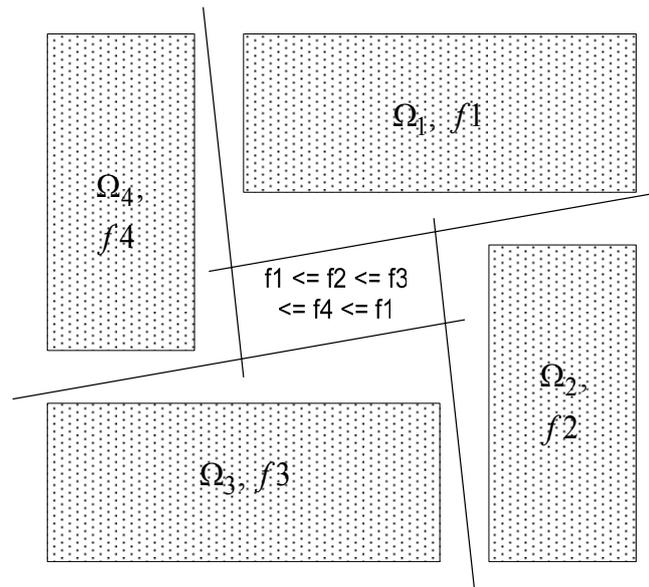


Рис. 5.2. Пример линейно неразделимых множеств

Рассмотрим условия линейной разделимости множеств для произвольного m .

Лемма 5.1. Пусть каждое множество Ω_i состоит из одной точки, $i = 1, \dots, m$. Тогда существует линейный классификатор $a(x, W)$, правильно разделяющий точки из $\Omega_i, i = 1, \dots, m$ (множества $\Omega_i, i = 1, \dots, m$, линейно разделимы).

Доказательство. Пусть $F(x)$ – строго выпуклая дифференцируемая функция такая, что все точки из $\Omega_i, i = 1, \dots, m$ принадлежат области определения $F(x)$. Каждому множеству $\Omega_i = \{x^i\}$ поставим в соответствие функцию $f_i(x, W^i) = (w^i, x) + w_0^i$ так, что $w^i = \nabla F(x^i)$, $w_0^i = F(x^i) - (\nabla F(x^i), x^i)$, $i = 1, \dots, m$. Тогда $f_i(x, W^i) = F(x^i) + (\nabla F(x^i), x - x^i)$. В силу строгой выпуклости имеем

$$f_i(x^i, W^i) = F(x^i) > F(x^j) + (\nabla F(x^j), x^i - x^j) = f_j(x^j, W^j), \quad j \neq i.$$

Отсюда следует, что линейный классификатор $a(x, W)$, правильно разделяет точки из $\Omega_i, i = 1, \dots, m$. Лемма доказана. ■

Теорема 5.1. Пусть вокруг каждого множества Ω_i может быть построен шар S_i , $i=1, \dots, m$, так, что $S_i \cap S_j = \emptyset$, $i \neq j$. Тогда множества Ω_i , $i=1, \dots, m$ линейно разделимы.

Доказательство. Положим

$$B_r = \{x : \|x\| \leq r, x \in R^n\}, \quad F_r(x) = \begin{cases} -\sqrt{r^2 - \|x\|^2}, & \|x\| \leq r, \\ +\infty, & \|x\| > r. \end{cases}$$

Функция $F_r(x)$ строго выпуклая, ее график есть полусфера радиуса r в пространстве R^{n+1} , если $x \in B_r$.

Будем рассматривать r такие, что $S_i \subseteq B_r$, $i=1, \dots, m$. Для каждого множества Ω_i будем подбирать линейную функцию $f_i(x, W^i) = (w^i, x) + w_0^i$.

Обозначим $E_i = \{x \in R^n : (w^i, x) + w_0^i \geq F_r(x)\}$. Нетрудно видеть, что E_i — эллипс, если $E_i \subset \text{int } B_r$. В этом случае множество E_i есть проекция пересечения плоскости и полушара в R^{n+1} на пространство R^n . Пусть для функции $f_i(x, W^i) = (w^i, x) + w_0^i$ выполняется условие $E_i \subset \text{int } B_r$. Нетрудно видеть, что, если радиус r достаточно большой, то всегда можно подобрать функцию $f_i(x, W^i) = (w^i, x) + w_0^i$ так, что будет выполняться $S_i \subseteq E_i$. Будем выбирать функции $f_i(x, W^i)$ так, чтобы им соответствовали эллипсы E_i минимального размера (с минимальной максимальной осью) и выполнялось $S_i \subseteq E_i$. Понятно, что, увеличивая радиус r полусферы, всегда можно добиться того, чтобы эллипсы E_i , $i=1, \dots, m$ не пересекались.

Пусть построены такие функции $f_i(x, W^i) = (w^i, x) + w_0^i$, что соответствующие им эллипсы E_i не пересекаются и выполняется $S_i \subseteq E_i$, $i=1, \dots, m$. Легко видеть, что по построению $F_r(x) > f_j(x, W^j)$, $x \notin E_j$ и $f_i(x, W^i) \geq F_r(x) > f_j(x, W^j)$, $x \in E_i$, $i \neq j$. Таким образом

$f_i(x, W^i) > f_j(x, W^j)$, $x \in S_i$, $i \neq j$, т.е. линейный классификатор $a(x, W)$ правильно разделяет точки из Ω_i , $i = 1, \dots, m$. Теорема доказана. ■

Близкая по содержанию теорема сформулирована в [147]. Приведенное доказательство более простое, чем в [147], и получено независимо.

Другие условия линейной разделимости приведены в [158]. Следует отметить, что условия сформулированных теорем достаточности весьма жесткие. Можно привести много примеров, когда эти условия не выполняются, но, тем не менее, существует линейный классификатор, правильно разделяющий точки из Ω_i , $i = 1, \dots, m$.

Каждое множество Ω_i , $i = 1, \dots, m$ есть обучающая выборка точек из некоторого класса $\bar{\Omega}_i$, известного только на элементах выборки. Обучение классификатора $a(x, W)$ заключается в подборе значений параметров W , при котором классы $\bar{\Omega}_i$, $i = 1, \dots, m$, разделяются «наилучшим» образом. Для определения качества разделения существуют различные подходы.

Обозначим $\Omega = \bigcup_{i=1}^m \Omega_i$. Пусть точки множества Ω перенумерованы, T – совокупность индексов, $\Omega = \{x^t : t \in T\}$, T_i – совокупность индексов множества Ω_i , $\Omega_i = \{x^t : t \in T_i\}$, $T = \bigcup_{i=1}^m T_i$. Положим $i(t)$ – номер множества Ω_i , которому принадлежит точка x^t , $t \in T$. Величина

$$\begin{aligned} g^t(W) &= \min \{ f_i(x^t, W^i) - f_j(x^t, W^j) : j \in \{1, \dots, m\} \setminus i, i = i(t) \} = \\ &= \min \{ (w^i - w^j, x^t) + w_0^i - w_0^j : j \in \{1, \dots, m\} \setminus i, i = i(t) \} \end{aligned} \quad (5.3)$$

называется отступом (margin) или зазором (gap) классификатора $a(x, W)$ на точке x^t , $t \in T$.

В случае $m = 2$ зазором классификатора на точке x^t является величина

$$g^t(W) = \begin{cases} (w, x^t) + w_0, & \text{если } t \in T_1, \\ -((w, x^t) + w_0), & \text{если } t \in T_2. \end{cases} \quad (5.4)$$

Классификатор $a(x, W)$ допускает ошибку на точке x^t тогда и только тогда, когда зазор $g^t(W)$ отрицателен. Величина $g(W) = \min\{g^t(W) : t \in T\}$ называется зазором классификатора $a(x, W)$ на совокупности множеств $\Omega_i, i = 1, \dots, m$. Классификатор $a(x, W)$ правильно разделяет точки из $\Omega_i, i = 1, \dots, m$, если $g(W) > 0$.

Замечание 5.1. Классификатор $a(x, W)$ инвариантен относительно умножения всех функций f_i (векторов W^i) на положительное число, зазор $g(W)$ линеен относительно такой операции умножения. Классификатор $a(x, W)$ и зазор $g(W)$ инвариантны относительно добавления произвольного числа ко всем f_i .

Величину $g(W)$ можно использовать как критерий качества классификатора $a(x, W)$ (чем больше значение $g(W)$, тем надёжнее разделяются точки из $\Omega_i, i = 1, \dots, m$), однако при этом должна выполняться некоторая нормировка совокупности векторов W , которую обозначим $\eta(W)$ и будем называть нормой классификатора $a(x, W)$.

В качестве нормы $\eta(W)$ классификатора могут рассматриваться, например, следующие функции:

$$\eta(W) = \sqrt{\sum_{i=1}^m \|W^i\|^2}, \quad (5.5)$$

$$\eta(W) = \max\{\|W^i\| : i = 1, \dots, m\}, \quad (5.6)$$

$$\eta(W) = \max\{|w_j^i| : i = 1, \dots, m, j = 1, \dots, n\}, \quad (5.7)$$

где $\|W^i\| = \sqrt{\sum_{j=1}^n (w_j^i)^2}$.

Пусть задана совокупность множеств $\Omega_i, i=1, \dots, m$. С учетом введенных обозначений задачу построения оптимального классификатора (определения значений параметров W) запишем в следующем виде: найти

$$g^* = \max_W \{g(W) : \eta(W) = 1, W \in R^L\}. \quad (5.8)$$

Функция $g(W)$ – вогнутая, $\eta(W)$ – выпуклая.

В целом задача (5.8) является задачей невыпуклого программирования, но в случае линейной разделимости множеств $\Omega_i, i=1, \dots, m$, ограничение $\eta(W) = 1$ можно заменить на $\eta(W) \leq 1$, и эта задача становится задачей выпуклого программирования, для решения которой существуют эффективные алгоритмы.

В случае линейной неразделимости множеств $\Omega_i, i=1, \dots, m$, для решения задачи (5.8) не существует эффективных алгоритмов и для поиска приближенных решений используются различные эвристические алгоритмы. Наиболее известным и широко используемым является метод опорных векторов (SVM). В последующих подразделах приводится сравнительный анализ различных подходов к построению приближенных решений задачи (5.8).

5.1.2. Классификаторы для линейно разделимых множеств

Задача построения линейных классификаторов имеет вид: найти

$$g^* = \max_W \{g(W) : \eta(W) \leq 1, W \in R^L\}. \quad (5.9)$$

Поскольку вектор $W = 0$ является допустимым, то задача (5.9) имеет решение всегда и $g^* \geq g(0) = 0$. Заметим, что $g^* > 0$, если множества $\Omega_i, i = 1, \dots, m$, линейно разделимы. Рассмотрим также задачу: найти

$$\eta^* = \min_V \{ \eta(V) : g(V) \geq 1, V \in R^L \}. \quad (5.10)$$

Близкие по смыслу задачи рассматривались разными авторами (см., например, [174, 138]).

Лемма 5.2. Пусть W^* – оптимальное решение задачи (5.9). Тогда:

1) если $g^* > 0$, то задача (5.10) также имеет оптимальное решение

$$V^* \text{ и } V^* = W^* / g^*, \quad \eta^* = 1 / g^*;$$

2) если $g^* = 0$, то задача (5.10) не имеет допустимых решений.

Доказательство.

1) Пусть $g^* > 0$. Понятно, что $\eta(W^*) = 1$. В противном случае ($\eta(W^*) < 1$), полагая $\bar{W} = qW^*$, где $q = 1/\eta(W^*) > 1$, можно было бы увеличить значение g^* .

Положим $V^* = W^* / g^*$. Вектор V^* есть допустимое решение задачи (5.10), т.к. $g(V^*) = 1$. Покажем, что V^* – оптимальное решение задачи (5.10). Предположим противное, т.е. существует вектор \bar{V} такой, что $\eta(\bar{V}) < \eta(V^*) = 1/g^*$, $g(\bar{V}) = 1$. Тогда, полагая $\bar{W} = \bar{V} / \eta(\bar{V})$, получаем допустимый вектор задачи (5.9) такой, что $g(\bar{W}) > g^*$, а это противоречит предположению об оптимальности значения g^* . Т.е. V^* – оптимальное решение задачи (5.10).

2) Пусть $g^* = 0$. Предположим, что задача (5.10) имеет допустимое решение \bar{V} и $g(\bar{V}) = 1$. Тогда $\eta(\bar{V}) > 0$. Полагая $\bar{W} = \bar{V} / \eta(\bar{V})$, получаем

допустимый вектор задачи (5.9), для которого $g(\overline{W}) > 0$, что противоречит условию $g^* = 0$. Лемма доказана. ■

Рассмотрим более подробно задачи построения линейных классификаторов для заданной совокупности множеств $\Omega_i = \{x^t, t \in T_i\}$, $i = 1, \dots, m$. В качестве нормы $\eta(W)$ будем использовать евклидову норму (5.5). Нетрудно видеть, что задача (5.9) может быть представлена в форме задачи линейного программирования с дополнительным квадратичным ограничением: найти

$$g^* = \max_{w, \delta} \delta \quad (5.11)$$

при ограничениях

$$(w^i - w^j, x^t) + w_0^i - w_0^j \geq \delta, \quad j \in \{1, \dots, m\} \setminus i, \quad t \in T_i, \quad i = 1, \dots, m, \quad (5.12)$$

$$\sum_{i=1}^m \sum_{j=1}^n (w_j^i)^2 \leq 1. \quad (5.13)$$

Соответственно задача (5.10) есть задача квадратичного программирования: найти

$$\eta^* = \min_v \sum_{i=1}^m \sum_{j=1}^n (v_j^i)^2 \quad (5.14)$$

при ограничениях

$$(v^i - v^j, x^t) + v_0^i - v_0^j \geq 1, \quad j \in \{1, \dots, m\} \setminus i, \quad t \in T_i, \quad i = 1, \dots, m. \quad (5.15)$$

Заметим, что в случае $m = 2$ задача (5.14), (5.15) эквивалентна задаче, которая используется для построения полосы максимальной ширины, разделяющей линейно разделимые множества Ω_1, Ω_2 , и записывается (см. например, [72]) в следующем виде: найти значения параметров $u \in R^n$, $u_0 \in R$ такие, что

$$(u, u) \rightarrow \min \quad (5.16)$$

при ограничениях

$$(u, x^t) - u_0 \geq 1, t \in T_1 \quad (5.17)$$

$$(u, x^t) - u_0 \leq -1, t \in T_2. \quad (5.18)$$

Пусть $\bar{v}^1, \bar{v}_0^1, \bar{v}^2, \bar{v}_0^2$ – оптимальное решение задачи (5.14), (5.15), u^*, u_0^* – оптимальное решение задачи (5.16)–(5.18). Тогда $\bar{v}^1 = -\bar{v}^2, u^* = \bar{v}^1 - \bar{v}^2, u_0^* = \bar{v}_0^1 - \bar{v}_0^2$.

5.1.3. Минимизация полосы неправильной классификации

Рассмотрим случай двух классов ($m = 2$). В случае линейно неразделимых множеств (обучающей выборки) для построения классификаторов используются самые разные критерии и математические модели.

Широко используемым критерием является минимизация эмпирического риска, т.е. числа точек обучающей выборки, которые классифицируются неправильно. Задача, возникающая при этом, является NP-полной, поэтому для практического построения классификаторов используются различные упрощенные математические модели. Эти вопросы рассматриваются в последующих разделах.

Для анализа другого возможного критерия рассмотрим задачу: найти

$$g^* = \max \{g(W) : \eta(W) = 1\}. \quad (5.19)$$

Пусть в качестве нормы $\eta(W)$ используется евклидова норма

$$\eta(W) = \sqrt{\sum_{j=1}^n (w_j)^2}, \quad W^* \text{ – решение задачи (5.19). Для линейно неразделимой}$$

выборки имеет место неравенство $g^* \leq 0$. При этом точки x^t такие, что

$g^t(W^*) < 0$, классифицируются неправильно, а значение $|g^t(W^*)|$ есть

расстояние от гиперплоскости $f(x, W^*) = \langle w^*, x \rangle + w_0^* = 0$ до точки x^t .

Значение $|g^*|$ есть расстояние до наиболее удаленной от гиперплоскости $f(x, W^*) = \langle w^*, x \rangle + w_0^* = 0$ точки, которая классифицируется неправильно.

Таким образом, (5.19) есть задача построения гиперплоскости $f(x, W^*) = \langle w^*, x \rangle + w_0^* = 0$ и полосы минимальной ширины, вне которой точки обучающей выборки классифицируются правильно.

Вместо евклидовой нормы в ограничении задачи (5.19) могут также использоваться другие нормы, в частности $\eta(W) = \max \{|w_j| : j = 1, \dots, n\}$. Выбор нормы зависит от содержания прикладной задачи. Эти вопросы рассматриваются, в частности, в [7].

Задача (5.19) невыпукла, поскольку невыпукло ограничение $\eta(W) = 1$. Трудоемкость решения этой задачи существенно зависит от используемой нормы. Так в случае, когда $\eta(W)$ – евклидова норма, для этой задачи неизвестны эффективные алгоритмы поиска точного решения, по-видимому, эта задача NP-полная.

Однако при специальном выборе нормы $\eta(W)$ имеет место

Теорема 5.2. Пусть множества Ω_1, Ω_2 линейно неразделимы, в задаче (5.19) используется норма $\eta(W) = \max \{|w_j| : j = 1, \dots, n\}$. Тогда задача (5.19) полиномиально разрешима (достаточно решить $2n$ задач линейного программирования).

Доказательство. Множество $\Omega = \{w : \eta(W) = 1, w \in R^n\}$, где $\eta(W) = \max \{|w_j| : j = 1, \dots, n\}$, есть поверхность гиперкуба в пространстве R^n и состоит из совокупности граней гиперкуба $\Omega = \bigcup_{j=1}^n (C_j^+ \cup C_j^-)$, здесь $C_j^+ = \{w : w_j = 1, -1 \leq w_i \leq 1, i = 1, \dots, n, i \neq j\}$,

$$C_j^- = \{w: w_j = -1, -1 \leq w_i \leq 1, i = 1, \dots, n, i \neq j\}.$$

$$\text{Обозначим } g_j^+ = \max \{g(W) : w \in C_j^+\}, \quad g_j^- = \max \{g(W) : w \in C_j^-\}.$$

Понятно, что

$$g^* = \max \{g_j^+, g_j^-, j = 1, \dots, n\}. \quad (5.20)$$

Задача для отдельной грани C_j^+ есть задача линейного программирования и имеет вид: найти

$$g_j^+ = \max_{\xi, w, w_0} \xi \quad (5.21)$$

при ограничениях

$$\xi \leq \langle w, x^t \rangle + w_0, t \in T_1, \quad (5.22)$$

$$\xi \leq -\langle w, x^t \rangle - w_0, t \in T_2, \quad (5.23)$$

$$-1 \leq w_i \leq 1, i = 1, \dots, n, i \neq j, \quad (5.24)$$

$$w_j = 1. \quad (5.25)$$

Таким образом, для вычисления $g^* = \max \{g_j^+, g_j^-, j = 1, \dots, n\}$ необходимо решить $2n$ задач линейного программирования вида (5.21)–(5.25). Откуда следует полиномиальная разрешимость задачи (5.19).■

5.1.4. Минимизация эмпирического риска для линейно неразделимых множеств

Задачи (5.11)–(5.13) и (5.14), (5.15) позволяют находить оптимальный линейный классификатор только для линейно разделимых множеств. Для линейно неразделимых множеств задачи должны формулироваться иначе.

Для случая двух множеств: Ω_1, Ω_2 , в ряде работ (см., например, [9, 72, 88]) предлагаются формулировки задач, позволяющие находить линейные классификаторы в случае линейно неразделимых множеств. В других работах [174] для этого предлагаются специальные алгоритмы. Наиболее

известным в этой связи является метод опорных векторов (см., например, [72]).

В случае линейно неразделимой выборки естественным критерием выбора классификатора является минимизация эмпирического риска, т.е. числа точек обучающей выборки, которые классификатор разделяет неправильно.

Будем считать, что задан некоторый параметр $\delta > 0$ надежности разделения точек обучающей выборки $\Omega_i, i=1, \dots, m$. Будем говорить, что точки $x^t, t \in T$, для которых величина зазора $g^t(W) < \delta$, разделяются классификатором $a(x, W)$ ненадежно.

В дальнейшем значение эмпирического риска будем определять с учетом надежности, определяемой параметром δ – эмпирический риск равен числу точек обучающей выборки, которые классификатор разделяет неправильно или ненадежно.

Лемма 5.3. Пусть $x^\alpha \in \Omega_i, x^\beta \in \Omega_j$, классификатор $a(x, W)$ правильно разделяет эти точки, для нормы классификатора выполняется ограничение (5.13). Тогда существует $R > 0$ такое, что

$$-R \leq w_0^i - w_0^j \leq R. \quad (5.26)$$

Доказательство. Для точки x^α имеем $(w^i, x^\alpha) + w_0^i > (w^j, x^\alpha) + w_0^j$, откуда $w_0^i - w_0^j > (w^j, x^\alpha) - (w^i, x^\alpha) = (w^j - w^i, x^\alpha)$, аналогично для x^β – $w_0^j - w_0^i > (w^i - w^j, x^\beta)$. Таким образом,

$$(w^j - w^i, x^\alpha) < w_0^i - w_0^j < (w^j - w^i, x^\beta).$$

Учитывая (5.13), можно показать, что существует $r_{\alpha\beta} \geq 0$, такое что $-r_{\alpha\beta} \|x^\alpha\| \leq w_0^i - w_0^j \leq r_{\alpha\beta} \|x^\beta\|$. Откуда следует утверждение леммы. ■

Рассматриваемая задача заключается в определении минимального количества точек, которые нужно исключить из обучающей выборки, чтобы оставшиеся точки разделялись надежно. Естественно требовать, чтобы после

исключения в каждом классе оставалась хотя бы одна точка. Это возможно, если

$$\delta < \max \left\{ \|x^\tau - x^s\| : \tau \in T_1, s \in T_2 \right\}. \quad (5.27)$$

В дальнейшем будем предполагать выполнение этого условия. В силу леммы 5.3 существуют достаточно большие положительные числа $B_t, t \in T$, при которых задача минимизации эмпирического риска с учетом надежности δ представима в виде: найти

$$Q^* = \min_{w,y} \left\{ \sum_{t \in T} y_t \right\} \quad (5.28)$$

при ограничениях

$$(w^i - w^j, x^t) + w_0^i - w_0^j \geq \delta - B_t \cdot y_t, \quad j \in \{1, \dots, m\} \setminus i, \quad t \in T_i, \quad i = 1, \dots, m, \quad (5.29)$$

$$\eta(W) \leq 1, \quad (5.30)$$

$$\sum_{t \in T_i} y_t \leq |T_i| - 1, \quad i = 1, \dots, m, \quad (5.31)$$

$$0 \leq y_t \leq 1, \quad t \in T, \quad (5.32)$$

$$y_t = 0 \vee 1, \quad t \in T. \quad (5.33)$$

Переменная y_t определяет, учитывается ли точка x^t при формулировке задачи. Будем говорить, что числа $B_t, t \in T$, удовлетворяют **условию корректности**, если при $y_t = 1$ точка x^t исключается из обучающей выборки, т.е. ограничения (5.29) выполняются для x^t при любых допустимых значениях остальных переменных задачи.

Ограничения (5.31) определяют условие того, что, по крайней мере, одна точка из каждого множества Ω_i должна быть включена в задачу.

Оптимальное значение Q^* равно минимальному эмпирическому риску с учетом надежности δ . При формулировке задачи минимизации

эмпирического риска могут учитываться дополнительные условия и информация:

- цена ошибки на разных точках может быть разной, т.е. целевой функцией может быть взвешенная сумма ошибок,
- на допустимую область значений векторов W могут накладываться дополнительные ограничения.

Задача (5.28)–(5.33) – *NP*-полная. В связи с этим для практического использования становится актуальной разработка приближенных алгоритмов решения такой задачи. При небольших значениях размерности задачи могут применяться существующие программные средства оптимизации общего назначения (возможности такого использования будут рассматриваться далее).

В качестве приближенных могут рассматриваться алгоритмы, основанные на идеях направленного перебора (последовательного анализа вариантов, метода ветвей и границ), локального поиска. При разработке таких алгоритмов важным является наличие эффективных процедур вычисления оценок снизу величины Q^* и построения допустимых решений задачи (5.28)–(5.33). Для реализации этих процедур будем использовать непрерывную релаксацию задачи (5.28)–(5.33). Понятно, что все целочисленные формулировки задачи (5.28)–(5.33) при достаточно больших значениях величин B_i (удовлетворяющих условию корректности) эквивалентны. Однако непрерывная релаксация этой задачи и значение оценки снизу для Q^* существенно зависят от значений B_i , поскольку с увеличением B_i расширяется область допустимых решений непрерывной релаксации задачи (5.28)–(5.33). Для получения наилучшей оценки для Q^* необходимо использовать наименьшие возможные значения для B_i .

Рассмотрим непрерывную релаксацию сформулированной задачи – задачу (5.28)–(5.32). Оптимальное значение релаксированной задачи

обозначим q^* . Пусть переменные W зафиксированы. Учитывая (5.3), задачу минимизации по переменным y представим в следующем виде: найти

$$q(W) = \min_y \left\{ \sum_{t \in T} y_t \right\} \quad (5.34)$$

при ограничениях

$$y_t \geq \frac{1}{B_t} (\delta - g^t(W)), t \in T, \quad (5.35)$$

$$\eta(W) \leq 1, \quad (5.36)$$

$$\sum_{t \in T_i} y_t \leq |T_i| - 1, i = 1, \dots, m, \quad (5.37)$$

$$0 \leq y_t \leq 1, t \in T. \quad (5.38)$$

Обозначим $d^t(W) = \max \left(0, \frac{1}{B_t} (\delta - g^t(W)) \right)$. Очевидно, что если задача (5.34)–(5.38) имеет решение, то $y^t = d^t(W)$. Откуда получаем задачу минимизации по переменным W : найти

$$q^* = \min \sum_{t \in T} d^t(W) \quad (5.39)$$

при ограничениях

$$\eta(W) \leq 1, \quad (5.40)$$

$$\sum_{t \in T_i} d^t(W) \leq |T_i| - 1, i = 1, \dots, m, \quad (5.41)$$

$$d^t(W) \leq 1, t \in T. \quad (5.42)$$

Функции $d^t(W)$ – выпуклые кусочно-линейные, $\eta(W)$ – выпуклая. Для решения задачи (5.39)–(5.42) целесообразно применять эффективные методы негладкой оптимизации [184].

В дальнейшем будем рассматривать случай $m = 2$. Пусть $t \in T$, $s \in T_1$, $\tau \in T_2, s, \tau \neq t$. Для выбора наименьших возможных значений B_t рассмотрим задачу: найти

$$\beta_t^{s\tau} = \max \left\{ \delta - g^t(W) \right\}, \quad (5.43)$$

$$g^j(W) \geq \delta, \quad j = s, \tau, \quad (5.44)$$

$$\langle w, w \rangle \leq 1. \quad (5.45)$$

Обозначим

$$B_t^* = \max \left\{ \beta_t^{s\tau} : s \in T_1, \tau \in T_2, s, \tau \neq t \right\}, \quad t \in T. \quad (5.46)$$

Теорема 5.3. Числа $B_t, t \in T$, удовлетворяют условию корректности в задаче (5.28)–(5.33), если

$$B_t \geq B_t^*, \quad t \in T. \quad (5.47)$$

Доказательство. Пусть зафиксирован некоторый индекс $t \in T$. Точка x^t исключается из обучающей выборки при $y_t = 1$, если ограничение (5.29) для данного индекса выполняется при любых допустимых значениях оставшихся переменных.

Обозначим $y = (y_\tau, \tau \in T)$, Y – множество всех y , удовлетворяющих ограничениям (5.31), (5.33), $D(y)$ – множество всех векторов W , удовлетворяющих ограничениям (5.29), (5.30) при заданном значении вектора y . Рассмотрим вектор $y \in Y$ такой, что $y_t = 1$. Положим

$$\beta_t(y) = \min \left\{ \theta : g^t(W) \geq \delta - \theta, W \in D(y) \right\} = \max \left\{ \delta - g^t(W) : W \in D(y) \right\}.$$

Обозначим $\beta_t^* = \max \{ \beta_t(y) : y \in Y, y_t = 1 \}$. Очевидно, что условием исключения точки x^t из обучающей выборки при $y_t = 1$ является выполнение неравенства $B_t \geq \beta_t^*$. Пусть $s \in T_1$, $\tau \in T_2$, $s, \tau \neq t$. Обозначим $y^{s\tau} = (y_t, t \in T, y_s = 0, y_\tau = 0, y_j = 1, j \neq s, \tau)$. Нетрудно видеть, что для любого

$y \in Y$ такого, что $y_s = 0, y_\tau = 0$, выполняется $D(y) \subseteq D(y^{s\tau})$, т.е.

$\beta_t(y) \leq \beta_t(y^{s\tau})$. Откуда $\beta_t^* = \max \left\{ \beta_t(y^{s\tau}) : s \in T_1, \tau \in T_2, s, \tau \neq t \right\}$. Учитывая, что $\beta_t(y^{s\tau}) = \beta_t^{s\tau}$, т.е. $B_t^* = \beta_t^*$, получаем утверждение теоремы. ■

Рассмотрим более подробно задачу (5.43)–(5.45). Учитывая (5.4), перепишем эту задачу в виде

$$\beta_t^{s\tau} = - \min_{w, w_0} \left\{ \langle w, x^t \rangle + w_0 - \delta \right\}, \quad (5.48)$$

$$\langle w, x^s \rangle + w_0 \geq \delta, \quad (5.49)$$

$$-\langle w, x^\tau \rangle - w_0 \geq \delta, \quad (5.50)$$

$$\langle w, w \rangle \leq 1. \quad (5.51)$$

Если система ограничений (5.49)–(5.51) несовместна, то $\beta_t^{s\tau} = -\infty$. Это имеет место, если $\delta > \|x^s - x^\tau\|$. В силу (5.27) всегда существует пара s, τ , такая что $\delta \leq \|x^s - x^\tau\|$.

Легко видеть, что в оптимальном решении задачи (5.48)–(5.51) ограничение (5.51) обязательно выполняется как равенство, а из ограничений (5.49), (5.50) одно обязательно активно. Возможна также ситуация, когда активны оба ограничения. Рассмотрим случай, когда ограничение (5.50) неактивно в оптимальном решении. Используя правило множителей Лагранжа, получаем для оптимального решения

$$w = \frac{x^s - x^t}{\|x^s - x^t\|}, \quad w_0 = \delta - \langle w, x^s \rangle, \quad \beta_t^{s\tau} = \|x^s - x^t\|.$$

При этом для полученного вектора (w, w_0) должно выполняться ограничение (5.50). Если это ограничение не выполняется, то оптимальное решение должно строиться с учетом того, что ограничение (5.50) активно.

Полученные соотношения позволяют сравнительно просто определять значения B_t^* , $t \in T$.

5.1.5. Сравнение с другими подходами

Наряду с рассмотренными выше моделями будут рассматриваться метод опорных векторов (см., например, [72]) и метод робастного разделения [9, 88, 138] для построения классификатора, разделяющего два линейно неразделимых множества. Пусть, как и ранее, $\Omega_i = \{x^t, t \in T_i\}$, $i = 1, 2$, $T = T_1 \cup T_2$.

В методе опорных векторов решается задача, которую можно представить в следующем виде: найти

$$\min_{u, u_0, \xi} \left\{ \frac{1}{2}(u, u) + C \cdot \sum_{t \in T} \xi^t \right\} \quad (5.52)$$

при ограничениях

$$(u, x^t) + u_0 \geq 1 - \xi^t, t \in T_1, \quad (5.53)$$

$$(-u, x^t) - u_0 \geq 1 - \xi^t, t \in T_2, \quad (5.54)$$

$$\xi^t \geq 0, t \in T, \quad (5.55)$$

где $u \in R^n$, $u_0 \in R$, $\xi^t \in R$, $t \in T$.

При робастном разделении двух множеств [9, 88, 138] построение классификатора выполняется в два этапа. На первом решается задача: найти

$$\min_{u, u_0, \xi} \left\{ \frac{1}{|T_1|} \cdot \sum_{t \in T_1} \xi^t + \frac{1}{|T_2|} \cdot \sum_{t \in T_2} \xi^t \right\} \quad (5.56)$$

при ограничениях (5.53)–(5.55). На втором этапе при фиксированных значениях вектора u уточняется значение параметра u_0 .

Для сравнения различных подходов представим аналог задачи (5.28)–(5.33) для случая двух множеств в следующем виде:

$$Q^* = \min_{w, w_0, y} \left\{ \sum_{t \in T} y_t \right\} \quad (5.57)$$

при ограничениях

$$(w, x^t) + w_0 \geq \delta - B \cdot y_t, \quad t \in T_1, \quad (5.58)$$

$$(-w, x^t) - w_0 \geq \delta - B \cdot y_t, \quad t \in T_2, \quad (5.59)$$

$$(w, w) \leq 1, \quad (5.60)$$

$$\sum_{t \in T_i} y_t \leq |T_i| - 1, \quad i = 1, 2, \quad (5.61)$$

$$0 \leq y_t \leq 1, \quad t \in T, \quad (5.62)$$

$$y_t = 0 \vee 1, \quad t \in T. \quad (5.63)$$

Сделаем замену переменных в задаче (5.57)–(5.63): $w = \delta u$, $w_0 = \delta u_0$, $\xi^t = \frac{B y_t}{\delta}$, $t \in T_1 \cup T_2$. Тогда непрерывная релаксация этой задачи примет вид

$$q^* = \frac{\bar{\delta}}{B} \cdot \min_{u, u_0, \xi} \left\{ \sum_{t \in T} \xi^t \right\} \quad (5.64)$$

при ограничениях

$$(u, x^t) + u_0 \geq 1 - \xi^t, \quad t \in T_1, \quad (5.65)$$

$$(-u, x^t) - u_0 \geq 1 - \xi^t, \quad t \in T_2, \quad (5.66)$$

$$(u, u) \leq \frac{1}{\bar{\delta}^2}, \quad (5.67)$$

$$\xi^t \geq 0, \quad t \in T, \quad (5.68)$$

$$\xi^t \leq \frac{B}{\bar{\delta}}, \quad t \in T, \quad (5.69)$$

$$\sum_{t \in T_i} \xi^t \leq \frac{B}{\bar{\delta}} (|T_i| - 1), \quad i = 1, 2. \quad (5.70)$$

Обозначим $\chi, \gamma_i, i=1,2$, двойственные переменные для ограничений (5.67), (5.70) и рассмотрим функцию Лагранжа

$$L(\chi, \gamma, \xi, u) = \frac{\delta}{B} \sum_{t \in T} \xi^t + \chi \cdot ((u, u) - 1/\delta^2) + \sum_{i=1}^2 \gamma_i \left(\sum_{t \in T_i} \xi^t - \frac{B}{\delta} (|T_i| - 1) \right).$$

Положим

$$\varphi(\chi, \gamma) = \min_{u, u_0, \xi} L(\chi, \gamma, \xi, u) \quad (5.71)$$

при ограничениях (5.65), (5.66), (5.68), (5.69).

Пусть задан штрафной коэффициент C в задаче (5.52)–(5.55). Нетрудно видеть, что, полагая $\gamma = 0$ и выбирая χ из условия $\frac{\delta}{2\chi B} = C$, получаем

$$L(\chi, \gamma, \xi, u) = 2\chi \left\{ \frac{1}{2}(u, u) + C \cdot \sum_{t \in T} \xi^t \right\} - \frac{\chi}{\delta^2},$$

т.е. задача (5.71), (5.65), (5.66), (5.68) эквивалентна задаче (5.52)–(5.55) при указанном выборе значений двойственных переменных.

Полагая $\chi = 0$, получаем

$$L(\chi, \gamma, \xi, u) = \sum_{i=1}^2 \left\{ \left(\frac{\bar{\delta}}{B} + \gamma_i \right) \sum_{t \in T_i} \xi^t \right\} - \sum_{i=1}^2 \gamma_i \left(\frac{B}{\bar{\delta}} (|T_i| - 1) \right),$$

откуда следует, что, подбирая соответствующим образом значения γ , получаем задачу (5.71), (5.65), (5.66), (5.68), эквивалентную задаче, которая решается при робастном разделении двух множеств.

Таким образом, задачи, которые решаются в методе опорных векторов и при робастном разделении двух множеств являются частным случаем задачи (5.71), (5.65), (5.66), (5.68).

5.1.6. Результаты вычислительных экспериментов

5.1.6.1. Использование программных средств оптимизации общего назначения

Для решения рассмотренных задач может использоваться существующее эффективное программное обеспечение задач оптимизации общего назначения, если число точек в обучающей выборке не очень велико.

Для оценки возможностей предлагаемых подходов при построении линейных классификаторов был проведен вычислительный эксперимент. Рассматривались задачи вида (5.9), для которых норма $\eta(W)$ выбиралась в соответствии с (5.6) – задача линейного программирования с дополнительными квадратичными ограничениями, или в соответствии с (5.7) – задача линейного программирования. Каждая задача состояла из обучающей выборки для классов в пространстве R^2 , которые схематично представлены на рис. 5.3. Выборка для каждого класса формировалась случайным образом на основании равномерного распределения точек внутри класса.

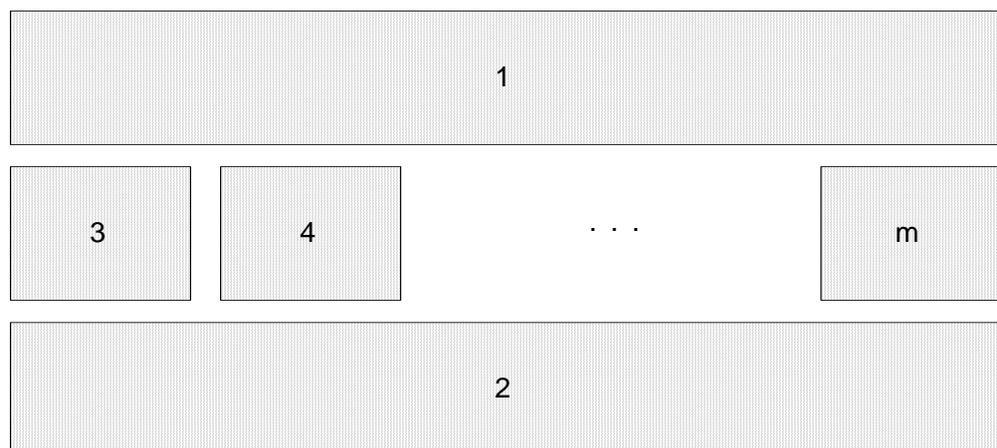


Рис. 5.3 Схематическое представление классов тестовых задач

Для решения задач использовались стандартные пакеты оптимизации, доступные в интернете на сервере NEOS (<http://www-neos.mcs.anl.gov/>) – MINOS, SNOPT, KNITRO, LOQO. Описание тестовых задач и формирование

исходных данных реализовано на языке моделирования AMPL (<http://www.ampl.com/>). Вычисления производились при различном расстоянии между границами классов, однако эти параметры несущественно сказывались на времени счета. Результаты вычислительных экспериментов приведены в табл. 5.1, 5.2. Число точек (в таблицах обозначается k) в выборке для каждого класса изменялось от 10 до 1000. Число классов (в таблицах – m) изменялось от 12 до 100.

Для тех тестовых задач, для которых были найдены оптимальные решения, оптимальные значения целевых функций положительны. Некоторые задачи не были решены из-за недостатка оперативной памяти (в таблицах пометка «memory»), для некоторых было получено допустимое решение, но оптимум не достигнут (пометка «not optimal»), некоторые не были завершены по неизвестной причине (не был получен ответ с сервера NEOS). Время счета приведено в табл. 5.1, 5.2. Из таблиц видно, что время решения задач с квадратичными ограничениями для задач большой размерности намного превышает время решения задач ЛП. В целом время решения вполне приемлемо для задач, возникающих на практике.

Таблица 5.1.

Результаты вычислительных экспериментов для задачи ЛП

№ задачи	m	k	Время решения задачи, сек.			
			MINOS	SNOPT	KNITRO	LOQO
1	12	1000	258.06	362.66	memory	53.73
2	12	500	136.44	189.13	81.86	14.46
3	12	100	5.8	25.01	2.37	1.12
4	12	10	0.19	0.64	0.3	0.19
5	25	10	10.61	25.05	2.51	1.05
6	50	10	unknown	unknown	17.17	2.39
7	100	10	unknown	unknown	memory	87.94

Таблица 5.2.

Результаты вычислительных экспериментов для задачи с квадратичными ограничениями

№ задачи	m	k	Время решения задачи, сек.		
			KNITRO	SNOPT	LOQO
1	12	1000	memory	411.67	not optimal
2	12	500	memory	63.09	not optimal
3	12	100	memory	5.59	not optimal
4	12	10	0.72	0.15	not optimal
5	25	10	7.44	2.66	not optimal
6	50	10	54.87	not optimal	not optimal
7	100	10	memory	not optimal	not optimal

При большом числе точек в обучающей выборке целесообразно использовать методы негладкой оптимизации [138, 184].

5.1.6.2. Использование методов негладкой оптимизации и специальных программных средств

Вычислительные эксперименты проводились для случая двух классов. Повторим формулировку задачи (определения значений параметров W) для линейно разделимых множеств $\Omega_1 = \{x^t : t \in T_1\}$, $\Omega_2 = \{x^t : t \in T_2\}$, $T = T_1 \cup T_2$: найти

$$g^* = \max_W \{g(W) : \eta(W) \leq 1, W \in R^{n+1}\}, \quad (5.72)$$

здесь $W = (w_0, w)$, $w = (w_1, \dots, w_n)$, $\eta(W) = \sqrt{\sum_{j=1}^n (w_j)^2}$,

$$g(W) = \min \{g^t(W) : t \in T\}, \text{ где } g^t(W) = \begin{cases} (w, x^t) + w_0, & \text{если } t \in T_1, \\ -((w, x^t) + w_0), & \text{если } t \in T_2. \end{cases}$$

Эту задачу можно записать в эквивалентной форме

$$\eta^* = \min_V \{\eta(V) : g^t(V) \geq 1, t \in T, V \in R^{n+1}\}. \quad (5.73)$$

Для построения классификатора в случае линейно неразделимых множеств будет использоваться непрерывная релаксация задачи минимизации эмпирического риска (5.28)–(5.33): найти

$$q^* = \min \sum_{t \in T} d^t(W) \quad (5.74)$$

при ограничениях

$$\eta(W) \leq 1, \quad (5.75)$$

$$\sum_{t \in T_i} d^t(W) \leq |T_i| - 1, i = 1, 2, \quad (5.76)$$

$$d^t(W) \leq 1, t \in T, \quad (5.77)$$

где $d^t(W) = \max\left(0, \frac{1}{B}(\bar{\delta} - g^t(W))\right)$, $\bar{\delta} > 0$ – параметр требуемой

надежности разделения точек обучающей выборки, B – некоторый параметр (достаточно большое положительное число).

В методе опорных векторов (SVM) решается задача, которая может быть представлена в следующем виде: найти

$$\eta^* = \min \left\{ \frac{1}{2} \eta(V)^2 + C \sum_{t \in T} \xi^t : g^t(V) \geq 1 - \xi^t, \xi^t \geq 0, t \in T, V \in R^{n+1} \right\}. \quad (5.78)$$

Метод опорных векторов (SVM) используется для построения оптимального классификатора как для линейно разделимых классов, так и для линейно неразделимых классов.

В случае линейно разделимых классов из теорем о негладких штрафах [155, 184] следует, что при достаточно большом коэффициенте C задачи (5.73) и (5.78) имеют одинаковые решения.

Существенной проблемой при использовании задачи (5.78) в случае линейно неразделимых классов является выбор значения коэффициента C . Необходимо отметить, что при использовании задачи (5.74)–(5.77) такая проблема не возникает.

Рассмотрим особенности задач (5.72) и (5.74)–(5.77), полезные при использовании подходов, описанных в предыдущих разделах:

1. Точка $W^0 = 0$ является внутренней точкой допустимого множества.
2. Оптимальные значения этих задач всегда больше или равны нулю.
3. Одномерный поиск точки на границе допустимого множества (при условии, что начальная допустимая точка $W^0 = 0$) реализуется просто: пусть $\eta^k = \eta(W^k)^2$ – квадрат нормы точки W^k , $\eta^k > 1$, тогда

точка $\bar{W} = \frac{W^k}{\sqrt{\eta^k}}$ является искомой точкой на границе допустимого

множества.

4. Функции $g^t(W)$ обладают свойством: $g^t(\alpha W) = \alpha g^t(W)$.

Для рассматриваемых задач были программно реализованы следующие подходы:

для задач (5.72) и (5.74)–(5.77) – метод точных штрафных функций с автоматической регулировкой штрафного коэффициента, метод выпуклых конических продолжений;

для задач (5.73) и (5.78) – метод точных штрафных функций без автоматической регулировки штрафного коэффициента.

Задачи безусловной оптимизации, к которым сводились исходные задачи с ограничениями, решались с помощью r -алгоритма Н.З. Шора [184].

Для проведения вычислительных экспериментов генерировались случайным образом задачи построения линейных классификаторов для двух классов. Параметры задач изменялись в пределах:

размерность n признакового пространства R^n – от 5 до 100;

число точек в обучающей выборке – от 40 до 100 000.

Точки в обучающей выборке для каждого класса генерировались на основании равномерного распределения внутри единичного куба. Эти кубы смещены относительно друг друга по первой координате так, что расстояние между ними равно единице. Для каждой задачи P_0 , сгенерированной таким образом, формировалось семейство задач P_i , $i = 1, \dots, 10$ за счет уменьшения смещения между классами (кубами). Расстояние между классами задачи P_i равно 2^{-i} . Все задачи сформированных семейств являются линейно разделимыми.

По результатам вычислительных экспериментов для линейно разделимых задач можно сделать следующие заключения:

метод точных штрафных функций с автоматической регулировкой штрафного коэффициента и метод выпуклых конических продолжений для задачи (5.11) показали приблизительно одинаковую эффективность, все задачи сформированных семейств успешно решены (точность по целевой

функции $\sim 10^{-6}$), число итераций r -алгоритма изменялось от ~ 100 для размерности $n = 5$ до ~ 1500 для размерности $n = 100$;

при использовании модели SVM (задача (5.78)) существенным является выбор коэффициента C – в вычислительных экспериментах использовалось значение $C = 1000$, при этом задачи P_i , $i \leq 5$ сформированных семейств решались успешно (была найдена разделяющая гиперплоскость), задачи P_i , $i \geq 7$ решены не были (не найдена разделяющая гиперплоскость).

Разработанные программные средства сравнивались с существующим программным обеспечением (LIBSVM – <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>). Время решения задач построения линейного классификатора в зависимости от числа точек в обучающей выборке приведено в табл. 5.3. Размерность пространства признаков – $n = 100$. Использовались стандартные установки для LIBSVM.

Таблица 5.3.

Время решения задач построения линейного классификатора

Число точек	Время решения, сек.	
	LIBSVM	Автоматическая регулировка штрафного коэффициента
5000	9,421	20,8
10000	24,234	24,3
25000	83,468	43
40000	186,484	51,1
50000	266,203	84,8

Таким образом, методы негладкой оптимизации предоставляют более широкие возможности при построении линейных классификаторов по сравнению с традиционными подходами. При этом быстрое действие разработанных программ оказывается сравнимым с существующими программными средствами.

5.1.6.3. Линейно неразделимые множества

В ходе вычислительных экспериментов проводилось сравнение качества решений, получаемых при использовании модели минимизации эмпирического риска (5.28)–(5.33), ее непрерывной релаксации (5.28)–(5.32) и модели SVM (5.78). Критерием качества является ошибка классификации – число точек обучающей выборки, классифицируемых неправильно. Для решения генерируемых задач использовался известный пакет программ CPLEX.

Как и ранее точки в обучающей выборке для каждого из двух классов генерировались на основании равномерного распределения внутри единичного куба. Эти кубы смещены относительно друг друга по первой координате так, что расстояние между ними равно 0,1. Семейство линейно неразделимых задач (множеств) формировалось итеративно, путем перемещения на текущей итерации по одной точке каждого класса в противоположный.

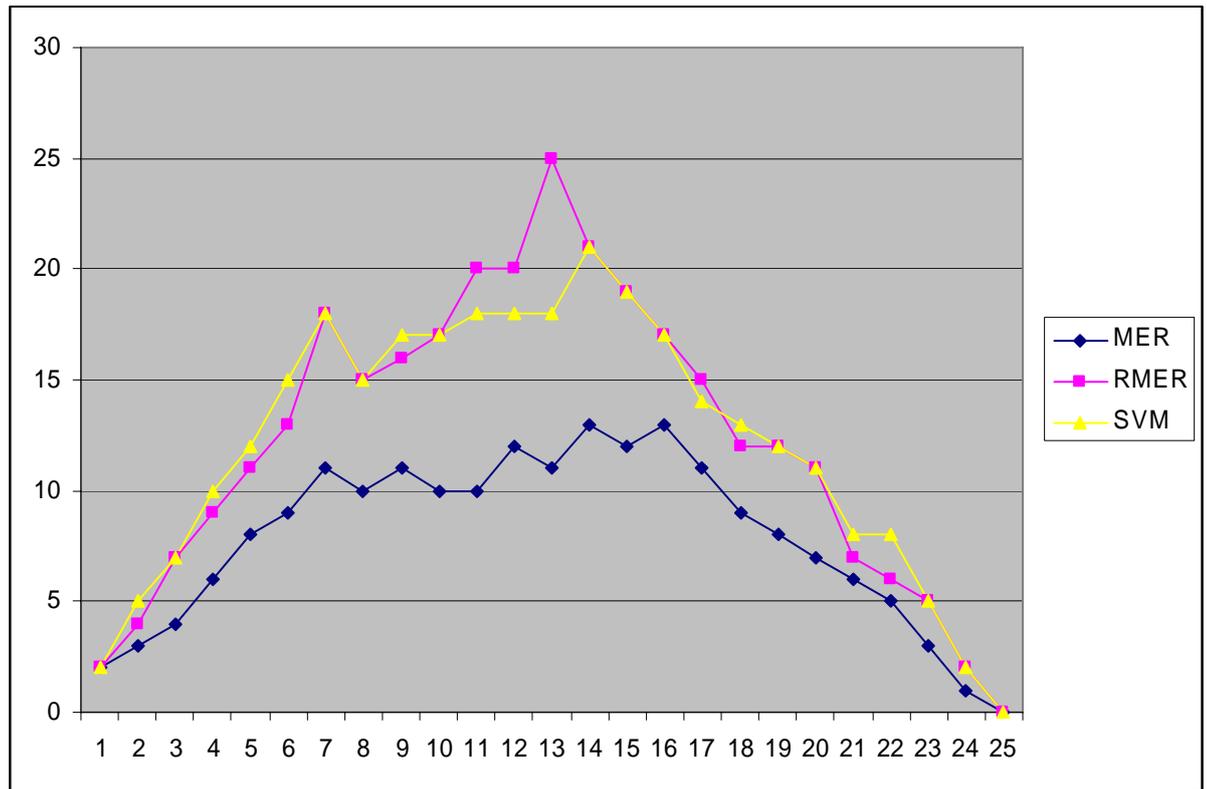


Рис. 5.4. Зависимость ошибки классификации от числа перемещаемых точек, $n = 5$, $|\Omega_i| = 25$, $i = 1, 2$

Модель (5.28)–(5.33) – NP-полная, поэтому для вычислительного эксперимента генерировались задачи малой размерности. На рис. 5.4 приведены результаты для случая, когда $|\Omega_i| = 25$, $i = 1, 2$, (Ω_i – точки обучающей выборке для класса i), $n = 5$ (n – размерность пространства признаков R^n). За 25 итераций все точки одного класса перемещаются в другой и наоборот. По оси абсцисс откладывается число перемещаемых точек одного класса, по оси ординат – ошибка классификации, MER – модель минимизации эмпирического риска (5.28)–(5.33), RMER – релаксированная модель минимизации эмпирического риска (5.28)–(5.32).

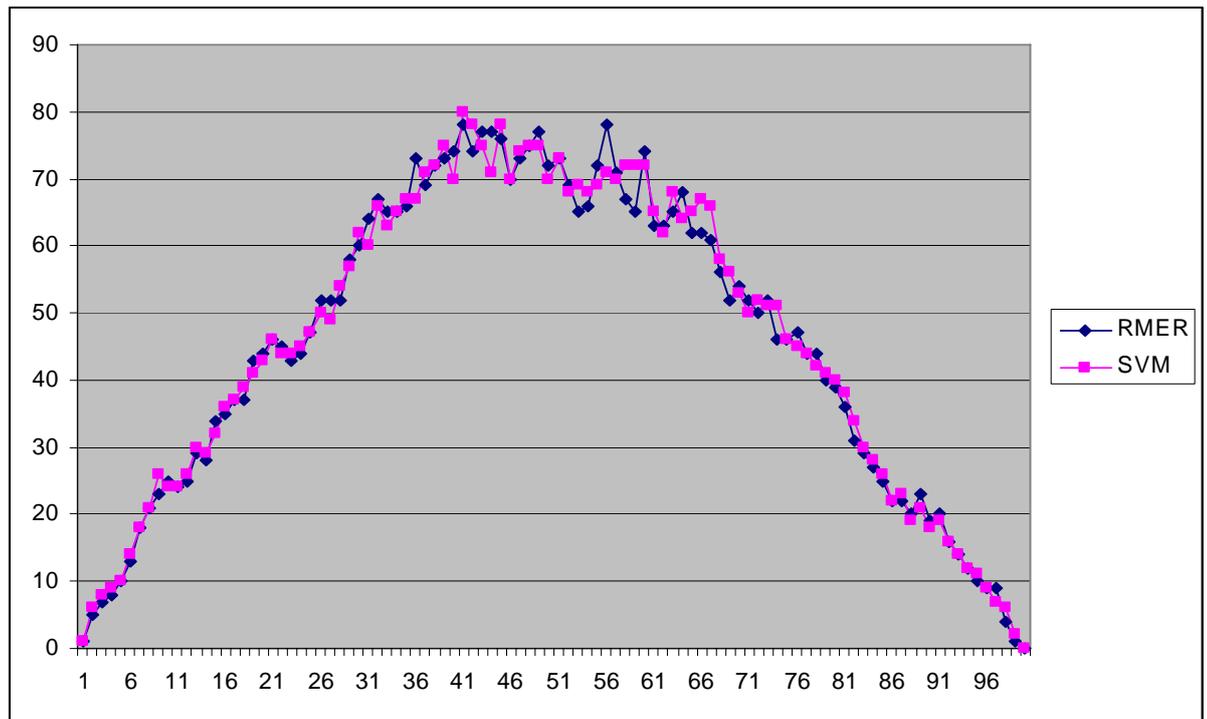


Рис. 5.5. Зависимость ошибки классификации от числа перемещаемых точек, $n = 30$, $|\Omega_i| = 100$, $i = 1, 2$

Трудоёмкость точного решения задач минимизации эмпирического риска (5.28)–(5.33) для семейства, приведенного на рис. 5.4, достигала 90 мин. При решении задач большей размерности появляются сообщения пакета CPLEX о недостаточности вычислительных ресурсов.

В связи с этим сравнение на задачах большей размерности проводилось только для релаксированной модели минимизации эмпирического риска RMER и модели SVM. На рис. 5.5 приведены результаты для случая $|\Omega_i| = 100$, $i = 1, 2$, $n = 30$.

Существенным для анализа возможностей разных моделей является сравнение на задачах, в которых величины $|\Omega_i|$, $i = 1, 2$, существенно различаются. При этом необходимо оценивать величину ошибки классификации отдельно по каждому классу. На рис. 5.6 приведены результаты для случая $|\Omega_1| = 30$, $|\Omega_2| = 200$, $n = 30$. Количество итераций при построении семейства задач равно 30.

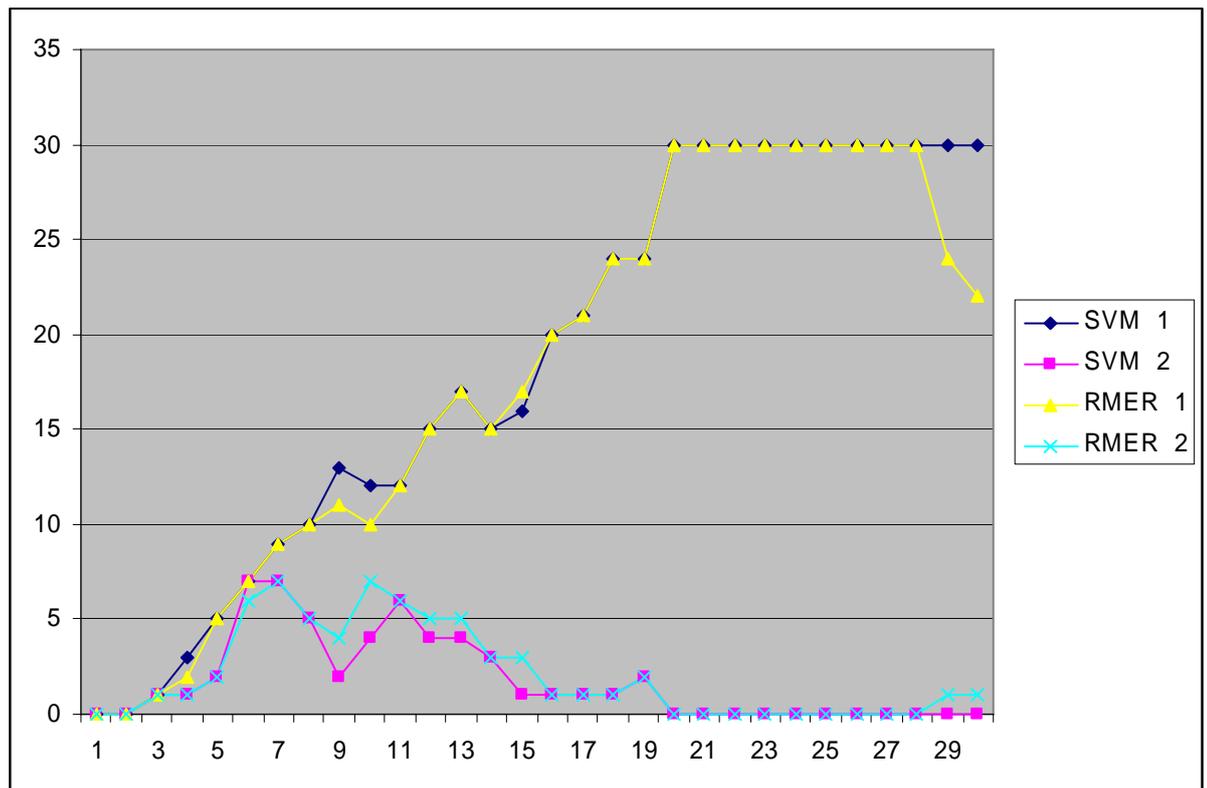


Рис. 5.6. Зависимость от числа перемещаемых точек ошибки классификации по каждому классу, $n = 30$, $|\Omega_1| = 30$, $|\Omega_2| = 200$,

SVM 1 – модель SVM, множество Ω_1 ,

SVM 2 – модель SVM, множество Ω_2 ,

RMER 1 – модель RMER, множество Ω_1 ,

RMER 2 – модель RMER, множество Ω_2

В заключение можно отметить, что ошибка классификации, получаемая при использовании модели минимизации эмпирического риска, существенно меньше ошибки, получаемой при использовании непрерывной релаксации этой модели и модели метода опорных векторов. Это сравнение проводилось на задачах малой размерности ввиду NP-полноты первой модели. Сравнение второй и третьей модели производилось также на задачах большей размерности. Получаемые ошибки классификации приблизительно одинаковы.

Полученные результаты позволяют сделать вывод о целесообразности разработки приближенных алгоритмов решения задачи минимизации

эмпирического риска для повышения качества формируемых классификаторов. Эти алгоритмы должны основываться на идеях направленного перебора (последовательного анализа вариантов, метода ветвей и границ), локального поиска.

5.2. Вложенные линейные классификаторы

Вложенным линейным (выпуклым кусочно-линейным) будем называть классификатор $a(x, W) = \arg \max_i \{f_i(x, W^i), i = 1, \dots, m\}$, для которого

$$f_i(x, W^i) = \max \{(w^s, x) + w_0^s, s \in S_i\}, S_i \cap S_j = \emptyset, i \neq j.$$

Обозначим $S = \bigcup_{i=1}^m S_i$. Ассоциированным назовем линейный классификатор

$$\tilde{a}(x, W) = \arg \max_s \{(w^s, x) + w_0^s, s \in S\}.$$

Обозначим:

$K_i, i = 1, \dots, m$ – разбиение пространства признаков R^n на классы вложенным классификатором $a(x, W)$;

$\tilde{K}_s, s \in S$ – разбиение пространства признаков R^n на классы ассоциированным классификатором $\tilde{a}(x, W)$.

Нетрудно видеть, что имеет место следующее

Утверждение 5.1. $K_i = \bigcup_{s \in S_i} \tilde{K}_s, i = 1, \dots, m$.

Лемма 5.4. Для произвольной совокупности конечных непересекающихся множеств $\Omega_i, i = 1, \dots, m$ существует вложенный линейный классификатор $a(x, W)$, правильно разделяющий точки из этих множеств.

Доказательство. Пусть точки множеств $\Omega_i, i = 1, \dots, m$, перенумерованы, T_i – совокупность индексов точек множества Ω_i , $\Omega_i = \{x^t, t \in T_i\}$. Обозначим $T = \bigcup_{i=1}^m T_i$. Считая каждую точку из множества $\{x^t, t \in T\}$ отдельным классом, построим линейный классификатор $\tilde{a}(x, W) = \arg \max_t \{(w^t, x) + w_0^t : t \in T\}$, правильно разделяющий эти точки. Это возможно в силу леммы 5.1.

Для каждого множества Ω_i , положим $f_i(x, W^i) = \max \{(w^t, x) + w_0^t, t \in T_i\}$. Нетрудно проверить, что в силу утверждения 5.1 классификатор $a(x, W) = \arg \max_i \{f_i(x, W^i) : i = 1, \dots, m\}$ удовлетворяет утверждению леммы. Лемма доказана. ■

Переход от линейных к вложенным линейным классификаторам аналогичен переходу от однослойных к многослойным нейронным сетям.

Так же, как и для многослойных нейронных сетей, построение вложенного классификатора включает две задачи:

- определение структуры кусочно-линейных функций – числа линейных функций для каждого $i, i = 1, \dots, m$,
- настройка значений весовых коэффициентов W .

Пусть структура кусочно-линейных функций задана. Для настройки значений весовых коэффициентов W рассмотрим задачу (5.9). Целевую функцию $g(W)$ (отступ классификатора $a(x, W)$) представим следующим образом

$$g(W) = \min \{f_i(x, W^i) - f_j(x, W^j) : j \in \{1, \dots, m\} \setminus i, x \in \Omega_i, i = 1, \dots, m\}.$$

В качестве нормы $\eta(W)$ классификатора выберем функцию (5.5). Рассматриваемая задача настройки значений весовых коэффициентов W имеет вид: найти

$$\max g(W) \quad (5.79)$$

при ограничении

$$\sum_{s \in S} \sum_{j=1}^n (w_j^s)^2 \leq 1. \quad (5.80)$$

Поскольку, $f_i(x, W^i) - f_j(x, W^j)$ есть разность кусочно-линейных выпуклых функций, то задача (5.79), (5.80) является задачей невыпуклой недифференцируемой оптимизации. Практические задачи при этом имеют большую размерность. Эффективные методы решения таких задач в настоящее время неизвестны.

В связи с выше сказанным особое значение приобретают эвристические схемы решения задач построения вложенных классификаторов.

Предлагаемая схема заключается в последовательном разбиении исходных множеств $\Omega_i, i = 1, \dots, m$ на непересекающиеся подмножества $\Omega_i = \bigcup_{j \in J_i} \Omega_i^j$ и в попытках построения для полученного семейства подмножеств ассоциированного линейного классификатора, правильно разделяющего точки подмножеств.

Эвристическая схема построения вложенного классификатора состоит из конечного числа шагов, на каждом из которых обрабатывается текущее

разбиение $\Omega_i = \bigcup_{j \in J_i} \Omega_i^j, i = 1, \dots, m$, и выглядит следующим образом:

1) На первом шаге $k=1$, текущее разбиение $\Omega_i = \bigcup_{j \in J_i} \Omega_i^j, i = 1, \dots, m$

полагается равным $\Omega_i, i = 1, \dots, m$.

2) На k -ом шаге для текущего разбиения $\Omega_i = \bigcup_{j \in J_i} \Omega_i^j, i = 1, \dots, m$

решается задача (5.11)–(5.13). Если оптимальное значение $g^* > 0$, то процесс завершен. В противном случае приближенно решается задача (5.28)–(5.33).

На основании полученного решения каждое подмножество Ω_j^j разбивается на два подмножества – точки, для которых $y_t = 0$ и точки, для которых $y_t = 1$. Текущее разбиение уточняется, полагается $k = k + 1$ и выполняется переход на 2).

Нетрудно видеть, что процесс конечен, и по завершении будет построен вложенный классификатор, правильно разделяющий точки из $\Omega_i, i = 1, \dots, m$.

5.3. Последовательные линейные классификаторы

Наряду с линейным классификатором (5.1) рассмотрим другой подход, который будем называть последовательным линейным классификатором. Пусть для каждой пары множеств $\Omega_i, \Omega_j, i, j = 1, \dots, m, i \neq j$ определен классификатор $a_{ij}(x, W^{ij})$ вида (5.2), разделяющий эти множества. Последовательный линейный алгоритм классификации (классификатор) заключается в последовательном применении классификаторов $a_{ij}(x, W^{ij})$ для анализа точки x . Если в результате применения $a_{ij}(x, W^{ij})$ окажется, что точка x принадлежит полупространству, содержащему множество Ω_i , то из дальнейшего анализа альтернативное множество Ω_j должно быть исключено. Таким образом, для классификации точки x необходимо $m - 1$ раз применить классификатор вида (5.2) для различных пар множеств $\Omega_i, \Omega_j, i, j = 1, \dots, m, i \neq j$.

Будем говорить, что множества $\Omega_i, i = 1, \dots, m$, последовательно линейно разделимы, если существует последовательный линейный классификатор, правильно разделяющий эти множества.

Нетрудно видеть, что условием последовательной линейной разделимости множеств является непересечение выпуклых оболочек этих множеств.

В пункте 5.1.1 приведен пример, показывающий, что непересечения выпуклых оболочек множеств не достаточно для линейной делимости в случае $m > 2$.

Таким образом возможности последовательных линейных классификаторов шире по сравнению с линейными классификаторами вида (5.1).

5.4. Задачи выделения информативных признаков

При построении алгоритмов классификации, разделяющих заданные совокупности множеств, часто возникают задачи выделения наиболее информативных признаков. Так в настоящее время накоплено огромное количество информации по экспрессии генов человека в связи с теми или другими заболеваниями. Размерность признакового пространства для таких задач может достигать $12000 \div 30000$. Понятно, что как получение такого объема информации при диагностике больных, так и обработка этой информации оказываются слишком трудоемкими. Специалисты в соответствующих прикладных областях предполагают, что информативными являются $10 \div 100$ признаков-генов.

Поскольку, как было отмечено в пункте 5.3, для случая многих классов возможности последовательных линейных классификаторов шире по сравнению с линейными классификаторами вида (5.1), задачи выделения информативных признаков будут рассматриваться для двух классов.

Рассмотрим случай линейно делимых множеств. Пусть множества Ω_1, Ω_2 заданы, задача (5.9) построения классификаторов решена и получено оптимальное значение $g^* > 0$. Понятно, что сокращение числа признаков при описании точек из Ω_1, Ω_2 будет приводить к ухудшению оптимального значения g^* . Зададимся некоторым порогом $0 < \bar{g} \leq g^*$ ухудшения значения g^* и рассмотрим задачу исключения максимального числа признаков при

условии того, что для зазора классификатора $g(W)$ выполняется $g(W) \geq \bar{g}$. Эта задача может быть представлена в виде: найти

$$\min \sum_{i=1}^n z_i, \quad (5.81)$$

$$g(W) \geq \bar{g}, \quad (5.82)$$

$$\sqrt{\sum_{i=1}^n w_i^2} \leq 1, \quad (5.83)$$

$$|w_i| \leq z_i, \quad i = 1, \dots, n, \quad (5.84)$$

$$z_i = 0 \vee 1, \quad i = 1, \dots, n. \quad (5.85)$$

Напомним, что $W = (w_0, w_1, \dots, w_n)$.

Задача (5.81)–(5.85) является *NP*-трудной. Рассмотрим ее непрерывную релаксацию. Можно показать, что она приводится к виду: найти

$$\min \sum_{i=1}^n |w_i|, \quad (5.86)$$

$$g(W) \geq \bar{g}, \quad (5.87)$$

$$\sqrt{\sum_{i=1}^n w_i^2} \leq 1. \quad (5.88)$$

Решение задачи (5.86)–(5.88) будем использовать в процедуре выделения информативных признаков (процедуре построения приближенных решений задачи (5.81)–(5.85)), состоящей из двух этапов, на каждом из которых в цикле последовательно решаются две задачи:

- сначала задача (5.9); в результате получаем оптимальное решение W^* и значение g^* ,

- потом решается задача (5.86)–(5.88), при этом точка W^* используется как начальная допустимая, на основании g^* уточняется \bar{g} , полученное решение \bar{W} используется для исключения части признаков.

На первом этапе исключаются все признаки такие, что $|\bar{w}_i| \leq \varepsilon$, где ε – параметр (значение ε выбиралось $\sim 10^{-5}$), на втором – k признаков с наименьшими значениями $|\bar{w}_i|$ (k – параметр). Переход с первого на второй этап осуществляется при уменьшении числа признаков до \bar{N} (\bar{N} – еще один параметр).

Процедура построения приближенных решений (сокращения неинформативных признаков) завершается, если при решении первой задачи значение g^* становится меньше G (G – параметр допустимого разделения множеств, выбирался $\sim 0.5 * g_0^*$, где g_0^* – оптимальное значение исходной задачи (5.9) при полном составе признаков).

Для случая линейно неразделимых множеств задачу исключения максимального числа признаков будем формулировать на основе задачи (5.39)–(5.42): найти

$$\min \sum_{i=1}^n z_i \quad (5.89)$$

при ограничениях

$$\sum_{t \in T} d^t(W) \leq \bar{q}, \quad (5.90)$$

$$\eta(W) \leq 1, \quad (5.91)$$

$$\sum_{t \in T_i} d^t(W) \leq |T_i| - 1, \quad i = 1, \dots, m, \quad (5.92)$$

$$d^t(W) \leq 1, \quad t \in T, \quad (5.93)$$

$$|w_i| \leq z_i, \quad i = 1, \dots, n, \quad (5.94)$$

$$z_i = 0 \vee 1, \quad i = 1, \dots, n. \quad (5.95)$$

Здесь $\bar{q} \geq q^*$, где q^* – оптимальное значение задачи (5.39)–(5.42).

Соответственно непрерывная релаксация имеет вид: найти

$$\min \sum_{i=1}^n |w_i| \quad (5.96)$$

при ограничениях

$$\sum_{t \in T} d^t(W) \leq \bar{q}, \quad (5.97)$$

$$\eta(W) \leq 1, \quad (5.98)$$

$$\sum_{t \in T_i} d^t(W) \leq |T_i| - 1, \quad i = 1, \dots, m, \quad (5.99)$$

$$d^t(W) \leq 1, \quad t \in T, \quad (5.100)$$

Для построения приближенных решений задачи (5.89) – (5.95) будем использовать аналог описанной ранее двухэтапной процедуры выделения информативных признаков: в каждом ее цикле сначала решается задача (5.39)–(5.42) для текущего состава признаков, а потом задача (5.96)–(5.100).

На практике часто возникает задача выделения минимального состава информативных признаков, содержащего фиксированное подмножество некоторых, существенных по каким-либо соображениям, признаков. Приближенное решение такой задачи может быть достигнуто путем запрета удаления элементов фиксированного подмножества в рассмотренной процедуре после решения задачи (5.86)–(5.88) (или задачи (5.96)–(5.100)) при удалении несущественных признаков.

Аналогичным образом может быть использована задача (5.52)–(5.55) (математическая модель метода опорных векторов). Однако при таком использовании возникает все та же проблема выбора коэффициента C .

Предложенный подход использовался при выделении информативных признаков в задаче диагностики онкологических заболеваний мозга (глиобластома мозга). Один из подходов к такой диагностике, исследованию

которого в настоящее время посвящено большое количество работ, основан на измерении так называемой экспрессии генов. Эта характеристика измеряется у больных для каждого гена из некоторой заданной совокупности и принимает цифровые значения из интервала $[0, r]$, $r \sim 10$. Накоплены большие объемы информации по экспрессии генов у больных и здоровых пациентов. Эта информация может использоваться при диагностике новых пациентов.

Собранная информация предоставлена Институтом генетики НАН Украины и состоит из выборки здоровых пациентов и выборки четырех групп больных пациентов разных стадий заболевания – всего 453 элемента.

Каждый элемент выборки характеризуется значениями экспрессии 25000 генов (размерность признакового пространства – 25000).

На основании этой информации решались следующие задачи:

- построение линейных классификаторов для каждой пары множеств из выборки;
- построение вариантов подмножеств информативных признаков.

Состав фиксированного подмножества признаков определялся специалистами в предметной области.

Поскольку размерность признакового пространства уменьшается по ходу работы предложенной двухэтапной процедуры, то для решения формируемых задач безусловной оптимизации использовался алгоритм обобщенного градиентного спуска, если размерность признакового пространства была больше 300, и r -алгоритм, если размерность пространства была меньше 300.

В результате применения предложенного подхода количество информативных признаков удавалось сокращать до $20 \div 50$, что соответствует предположениям специалистов в прикладной области.

5.5. Выводы

В разделе рассматриваются задачи построения классификаторов, относящих каждую точку пространства признаков к одному из m классов. Для каждого класса считается заданной дискриминантная функция, зависящая от параметров. При классификации точки вычисляются значения всех дискриминантных функций в этой точке, и точка относится к классу, имеющему наибольшее значение дискриминантной функции. Такие классификаторы называются дискриминантными. В случае использования линейных дискриминантных функций классификаторы называются линейными.

В разделе исследуются различные математические модели задач построения оптимальных линейных классификаторов для заданных обучающих множеств (выборок). Приводятся условия линейной разделимости множеств, близкие известным ранее [147,158].

Для линейно разделимых множеств сформулированные модели мало отличаются от известных ранее. Задачи построения классификаторов решаются эффективно.

Для линейно неразделимых множеств предложена новая модель минимизации эмпирического риска. Сформулированная задача является частично целочисленной. Для вычисления оценки оптимального решения и построения приближенных решений предлагается использовать непрерывную релаксацию такой задачи. Показано, что известные математические модели, используемые в методе SVM опорных векторов (см., например, [72]) и методе робастного линейного программирования [9], являются частным случаем Лагранжевой релаксации (при специальном выборе множителей Лагранжа) предложенной модели.

Также для линейно неразделимых множеств рассмотрена задача построения полосы минимальной ширины, вне которой точки обучающей выборки классифицируются правильно. Рассмотренная задача невыпукла, но

при использовании нормы $\max\{|w_j|: j=1, \dots, n\}$ решение задачи сводится к решению $2n$ задач линейного программирования, где n – размерность пространства признаков.

Приведены результаты вычислительных экспериментов, показывающие эффективность использования методов негладкой оптимизации в задачах с большим числом точек в обучающих множествах.

В разделе также рассмотрены классификаторы, в которых в качестве дискриминантных функций используются выпуклые кусочно-линейные функции. С помощью таких классификаторов можно обеспечить правильное разделение точек для любой совокупности конечных непересекающихся множеств. Однако задачи построения таких классификаторов являются невыпуклыми и для их решения неизвестны эффективные алгоритмы.

Наряду с линейными классификаторами в случае многих классов в разделе рассмотрены последовательные линейные классификаторы, основанные на формировании линейных классификаторов для каждой пары множеств из обучающей выборки. Показано, что возможности последовательных линейных классификаторов шире по сравнению с линейными классификаторами.

Важным приложением разрабатываемых подходов являются задачи выделения информативных признаков в медицинской диагностике. В разделе представлены разработанные алгоритмы выделения наиболее информативных признаков в задачах диагностики на основании экспрессии генов человека. Работа выполнялась совместно с Институтом генетики НАН Украины. Исходная размерность признакового пространства – 25000 генов. В результате применения предложенного подхода количество информативных признаков удавалось сокращать до $20 \div 50$, что соответствует предположениям специалистов в прикладной области.

РАЗДЕЛ 6. ЗАДАЧИ ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ СЛОЖНЫХ ТЕХНИЧЕСКИХ ОБЪЕКТОВ

Энергетика является одним из ключевых секторов экономики Украины. Более 40% тепловой и электрической энергии производится на тепловых электростанциях. Энергоблоки большинства электростанций исчерпали свой технический ресурс и работают с низким КПД. В настоящее время в ремонте и реконструкции нуждаются более 100 энергоблоков. Энергетические котлоагрегаты являются сложными техническими объектами, состоящими из большого количества взаимосвязанных компонент. Стоимость одного энергоблока - \$70 - \$500 млн. Большие финансовые и материальные затраты при создании и реконструкции энергетических котлоагрегатов ТЭС обуславливают практическую важность разработки программных средств оптимизации проектных решений для таких технических объектов [151, 92, 172, 164]. В Институте Кибернетики им. В.М. Глушкова НАНУ совместно с Харьковским ЦКБ «ЭНЕРГОПРОГРЕСС» на протяжении многих лет ведутся работы по решению отдельных задач этого класса [103]. В разделе приводятся основные результаты этих работ.

В настоящее время существует множество реализаций программных средств оптимизации общего назначения. Однако их использование для решения сложных реальных задач оптимизации связано с существенной трудоемкостью привязки к предметной области и обычно редко приводит к успеху. Это в значительной степени обусловлено тем, что в таких реализациях не учитываются специфические особенности задач конкретной области приложения. Опыт показывает, что эффективное программное обеспечение сложных задач оптимизации должно разрабатываться на основе учета их структурных, информационных и математических особенностей.

В разделе рассматриваются особенности создания математического и программного обеспечения задач оптимального проектирования

энергетических котлоагрегатов. Эти задачи описываются на содержательном уровне, приводятся соответствующие математические модели. Подробно рассматривается модель оптимального проектирования на номинальный режим и ее особенности, позволяющие редуцировать исходную задачу к задаче меньшей размерности. Описывается модель проектирования, учитывающая альтернативные режимы эксплуатации. Приводятся описание разработанных программных средств и результаты их апробации на реальных задачах.

Работы выполнялись в рамках:

научного проекта „5.9. Разработка программных средств оптимального проектирования при реконструкции и модернизации энергетических котлоагрегатов ТЭС” программы „Проблемы ресурса и безопасности эксплуатации конструкций, сооружений и машин” (РЕСУРС) НАН Украины– 2007 – 2009 гг.,

научно-технического и инновационного проекта организаций НАН Украины ВК 120.12.08 «Разработка прототипа программных средств оптимального проектирования при реконструкции и модернизации энергетических котлоагрегатов ТЭС сверхкритического давления» – сроки выполнения: II кв. 2008 – IV кв. 2008 года,

научно-технического проекта „Разработка прототипа программных средств оптимального проектирования энергетических котлоагрегатов ТП-210А сверхкритического давления” – договор № 1744 от 21.04.2008 г. с Харьковским ЦКБ "Энергопрогресс",

научно-технического и инновационного проекта организаций НАН Украины ВК 120.12.14 ”Разработка программных средств оптимального проектирования энергетических котлоагрегатов при реконструкции и модернизации ТЭС с учетом альтернативных режимов эксплуатации”, сроки выполнения: I кв. 2014 – IV кв. 2014 года.

6. 1. Общая характеристика задач

Энергетические котлоагрегаты являются сложными техническими объектами, состоящими из большого количества взаимосвязанных компонент. При их проектировании стремятся минимизировать стоимостные, технико-экономические (приведенные затраты) или экологические показатели, учитывая множество технических и технологических ограничений.

Расчеты значений показателей, входящих в указанные ограничения и целевые функции, выполняются в настоящее время с большой точностью по отработанным методикам [171, 146]. При этом значения конструктивных характеристик должны быть зафиксированы.

Задача выбора рациональных значений конструктивных характеристик элементов котлоагрегата имеет особую актуальность в связи с дефицитом материалов и возрастающими требованиями к эксплуатационным характеристикам. Ее решение связано с большими, часто принципиальными трудностями, и осуществляется конструктором на основании собственного опыта и нормативных рекомендаций, быстро устаревающих в связи с появлением новых технических решений.

В силу существенной взаимосвязанности элементов котлоагрегата и необходимости удовлетворить всем упомянутым ограничениям, в ходе проектирования приходится прорабатывать большое количество различных вариантов конструкций (до 500). Конструкторы вынуждены привязываться к существующим прототипам, что не всегда целесообразно.

Аналогичное положение характерно для процессов проектирования других технических объектов большой сложности. Для решения указанных проблем разрабатываются математические модели сложных технических объектов, специальные методы оптимизации, средства автоматизированного проектирования, включающие в себя подсистемы математического моделирования и оптимизации [172, 151, 91].

Автоматизация реальных процессов проектирования сложных технических объектов теплоэнергетики и решения комплексных задач оптимизации, которые при этом возникают, требуют разработки детальных математических моделей и специальных методов решения таких задач.

Компонентами энергетического котлоагрегата являются: топка, поверхности нагрева радиационных, ширмовых и конвективных перегревателей, экономайзеры, воздухоподогреватели и др. Связи между компонентами определяются тепловой схемой котлоагрегата. Проектируемая конструкция должна работать в нескольких альтернативных режимах, которые определяются используемым топливом и величиной нагрузки, и обеспечивать заданные параметры. При этом для всех режимов работы котлоагрегата должны выполняться технологические ограничения (ограничения по температуре стенок труб, коэффициентам запаса прочности, аэродинамическому сопротивлению, гидравлическому сопротивлению, скоростям теплоносителей, жесткости конструкций, по выбросам загрязняющих веществ и др.). Некоторые ограничения являются трудно формализуемыми и учитываются косвенным образом – ограничения по жесткости конструкций учитываются путем задания достаточно больших границ снизу на диаметры и толщины стенок труб, ограничения по выбросам загрязняющих веществ в окружающую среду обеспечиваются путем задания ограничения сверху на температуру газа на выходе из топки и доли рециркуляции газов.

Качество проектируемой конструкции оценивается одним или несколькими целевыми критериями. В качестве таких критериев рассматриваются стоимость, материалоемкость и приведенные затраты. Естественным критерием является коэффициент полезного действия котлоагрегата. Этот показатель непосредственно связан с температурой уходящих газов – повышение КПД приводит к снижению температуры. Однако при низких температурах резко повышается коррозия металлов. В

связи с этим в математических моделях используются достаточно жесткие ограничения по температуре уходящих газов.

В разработанных моделях состав компонент, тепловая схема, структурные характеристики всех поверхностей нагрева (шахматная или коридорная компоновка, прямоточное или противоточное включение по пару, типы оребрения и др.) предполагаются заданными. При необходимости конструктор может осуществить перебор по различным вариантам состава компонент. Также считаются заданными для каждого режима: способ сжигания и характеристики топлива; температура воздуха на входе воздушного тракта; характеристики теплоносителей на входах и выходах пароводяных трактов.

При формировании конкретной задачи оптимизации задается список варьируемых конструктивных параметров котлоагрегата (например, диаметры и толщины стенок труб, поперечные и продольные шаги, заходности, число петель и др.) и критерий качества конструкции (например, стоимость, материалоемкость, приведенные затраты).

В результате решения задачи оптимального проектирования котлоагрегата определяются значения варьируемых конструктивных параметров, и для каждого режима выполняется расчет характеристик материальных потоков (для каждого элемента тепловой схемы определяется расход теплоносителя, давление, температура на входах и выходах), значения показателей, определяющих технологические ограничения.

Все переменные в модели являются непрерывными величинами. В результате решения оптимизационной задачи могут быть получены дробные значения для величин, которые содержательно должны принимать дискретные значения (например, число петель, заходность, сортамент труб). Для таких переменных полученные значения округляются до ближайших дискретных значений.

Сложные функциональные зависимости в математических моделях определяются с некоторыми упрощениями с целью уменьшения трудоемкости вычислений. Допустимость этих упрощений определяется анализом и проверкой получаемых решений с использованием сертифицированных программ, удовлетворяющих утвержденным методикам [171].

Наиболее важным элементом информационного представления задачи является функциональная (тепловая) схема котлоагрегата. Тепловая схема описывает порядок омывания теплоносителями компонент котлоагрегата и определяет уравнения теплового и материального баланса математической модели. Эти уравнения являются линейными. В элементах тепловой схемы, соответствующих поверхностям нагрева, происходят процессы теплообмена, которые описываются нелинейными уравнениями. Отдельным элементом тепловой схемы является топка, в которой происходит сгорание топлива. Процессы тепловыделения и теплообмена в топке также описываются нелинейными уравнениями. В совокупности эти линейные и нелинейные уравнения формируют систему уравнений, определяющую процессы теплообмена в котлоагрегате в целом. Для решения такой системы уравнений в теплотехнике используется методика, называемая **тепловым расчетом** [171]. Тепловой расчет обеспечивает точность, необходимую для выполнения проектных работ, однако является слишком трудоемким для использования в оптимизационных алгоритмах и не обеспечивает расчет дополнительных необходимых характеристик (частных производных тех или других величин по входным параметрам).

В настоящем разделе рассматриваются две оптимизационные модели проектирования котлоагрегатов – модель проектирования на номинальный режим эксплуатации и модель проектирования, учитывающая альтернативные режимы эксплуатации.

В модели проектирования на номинальный режим эксплуатации расход топлива и значения управляющих характеристик (рециркуляция газов и величины впрысков) считаются заданными. В качестве целевых критериев рассматриваются стоимость, материалоемкость и приведенные затраты. Расчеты для режимов работы на частичных нагрузках, на альтернативном топливе выполняют после решения оптимизационной задачи для проверки полученных решений. Аналогичный подход при решении задач оптимального проектирования используется в [172].

В модели проектирования, учитывающей альтернативные режимы эксплуатации, кроме значений конструктивных характеристик котлоагрегата необходимо также для каждого режима определить расход топлива и оптимальные значения управляющих характеристик (рециркуляция газов и величины впрысков). В качестве целевых критериев используются стоимость, материалоемкость или функциональные характеристики основного режима – приведенные затраты, КПД, расход топлива.

В целом задачи оптимального проектирования (выбора рациональных значений конструктивных характеристик) энергетического котлоагрегата формулируются как задачи математического программирования. Число переменных и ограничений в таких задачах после возможных упрощений достигает нескольких тысяч. Система ограничений включает линейные и нелинейные уравнения и неравенства.

Важным моментом при формулировке математических моделей оптимального проектирования является выбор независимых переменных задач оптимизации. В исходной задаче требуется определить значения конструктивных характеристик котлоагрегата.

Одним из подходов является выбор этих величин в качестве независимых переменных. Как следствие получаем значения тепловосприятий поверхностей нагрева в тех режимах, которые учитываются в математической модели. При этом на каждой итерации оптимизационных

алгоритмов приходится решать сложные системы нелинейных уравнений большой размерности.

В случае, когда в математической модели учитывается только номинальный режим работы котлоагрегата, возможен альтернативный подход – выбор тепловосприятливых поверхностей нагрева в качестве независимых переменных. Одна из конструктивных характеристик поверхностей нагрева в этом случае получается в результате расчета. При таком подходе система уравнений задачи распадается на совокупность слабосвязанных блоков, что позволяет строить более эффективные алгоритмы.

Аналогичные особенности характерны для математических моделей других сложных технических объектов. Для описания структуры таких моделей вводится (на содержательном уровне) понятие «вычислительный модуль». В соответствии с объектно-ориентированным подходом строится иерархия классов (библиотека классов RBL – Recursive Block Library), базовым классом которой является класс «вычислительный модуль». Библиотека RBL включает в себя класс «система уравнений» и класс агрегирования сети вычислительных моделей. Указанные классы позволяют рекурсивно строить новые объекты из уже существующих. Библиотека RBL содержит также классы формирования и решения оптимизационных задач. При реализации конкретных приложений прикладные классы должны наследоваться от базовых классов библиотеки RBL.

В разделе рассматриваются примеры структуры некоторых задач оптимального проектирования энергетических котлоагрегатов, приводится описание разработанного прототипа системы оптимального проектирования, результаты вычислительных экспериментов, проведенных совместно с Харьковским ЦКБ „ЭНЕРГПРОГРЕС” для анализа возможного эффекта от использования предложенного подхода в процессах проектирования.

6.2. Модели оптимального проектирования на номинальный режим

При расчете котлоагрегата необходимо решать следующие группы уравнений:

материального баланса;

теплового баланса;

баланса лучистого тепла;

конвективного теплообмена;

поглощения лучистого тепла из топки.

Уравнения материального баланса и баланса лучистого тепла являются линейными. Уравнения теплового баланса являются линейными по удельным энтальпиям, но коэффициенты этих уравнений зависят от решения уравнений материального баланса. Остальные уравнения – нелинейные.

Параметрами уравнений теплового баланса, баланса лучистого тепла являются величины тепловосприятий поверхностей нагрева котла. Решения этих уравнений определяют температуры теплоносителей на входах и выходах каждой поверхности нагрева, а также лучистые потоки на входах каждой поверхности.

Каждой поверхности нагрева соответствует одно уравнение конвективного теплообмена и одно уравнение поглощения лучистого тепла из топки (последнее только для ширмовых и первой по ходу газов конвективной поверхности). Эти уравнения определяют величины тепловосприятий поверхностей нагрева как функции от входных (выходных) температур теплоносителей и конструктивных характеристик каждой поверхности.

Для поверхностей нагрева определяются температуры стенок труб и коэффициенты запаса прочности в выделенных точках, гидравлическое и аэродинамическое сопротивления. Эти величины являются функциями от

конструктивных характеристик и температур теплоносителей, в рассматриваемых задачах по ним формируются ограничения.

Особенности задач оптимального проектирования (выбора рациональных значений конструктивных характеристик) энергетического котлоагрегата:

уравнения материального баланса линейны и легко могут быть решены;

если задаться некоторыми лучистым и конвективным тепловосприятиями каждой поверхности нагрева, то уравнения теплового баланса, баланса лучистого тепла также легко могут быть решены (при фиксированных материальных потоках);

для каждой конвективной поверхности нагрева уравнения теплообмена могут быть решены (при фиксированных тепловосприятиях и характеристиках теплоносителей на входе и выходе) за счет варьирования конструктивных характеристик;

при решении уравнений теплообмена для поверхностей нагрева с фиксированными конструктивными характеристиками приходится пересчитывать всю тепловую схему котлоагрегата.

Поскольку при выборе рациональных значений конструктивных характеристик в математической модели явно учитывается только номинальный режим работы котлоагрегата, тепловосприятия поверхностей нагрева могут быть включены в состав независимых переменных. Одна из конструктивных характеристик поверхностей нагрева в этом случае определяется в результате расчета. При таком подходе система уравнений задачи распадается на совокупность слабосвязанных блоков, что позволяет строить более эффективные алгоритмы.

Режимы работы на частичных нагрузках, на альтернативном топливе используются для проверки полученных решений.

6.2.1. Структурное описание задач оптимального проектирования

Для описания структуры задачи оптимального проектирования введем понятие вычислительного модуля.

Каждый модуль описывается совокупностью входных $x \in X^n$, выходных переменных $y \in Y^m$ и реализует функции вычисления $y(x)$. Модуль также реализует функцию вычисления градиентов выходов y по входам x . В данном подразделе будем предполагать, что функция $y(x)$ определена на всем пространстве X^n . Каждый модуль описывается своими совокупностями входных и выходных переменных.

Понятиями вычислительных модулей могут описываться слабосвязанные блоки, на которые распадается система уравнений задачи оптимального проектирования. При этом одни модули будут описывать подсистемы линейных уравнений, другие – небольшие подсистемы нелинейных уравнений, связанные с поверхностями нагрева или группами поверхностей нагрева. Входами таких модулей будут параметры подсистем уравнений, выходами – их решения.

Рассмотрим модуль, описывающий некоторую подсистему нелинейных уравнений

$$g(x, y) = 0, \quad (6.1)$$

где x – параметры системы уравнений (входы модуля), y – переменные (выходы модуля), относительно которых решается система уравнений.

Такие модули будем называть модулями систем уравнений.

Вектор-функцию $g(x, y)$ будем задавать вычислительным модулем, входами которого является вектор (x, y) , а выходами – вектор $g(x, y)$.

Используя введенные понятия, будем описывать структуру задач оптимального проектирования совокупностью (сетью) вычислительных модулей, в которой выходы одних модулей являются входами других. Два

модуля будем связывать ориентированной дугой, если некоторые выходы одного модуля являются входами другого. Будем рассматривать ациклические сети модулей.

Объединение всех переменных (входных и выходных) модулей, входящих в сеть, будем называть переменными сети модулей. Если переменная является выходом какого-либо модуля, будем называть ее выходом сети. Будем рассматривать сети, в которых каждая выходная переменная является выходом только одного модуля.

Переменную сети модулей, не являющуюся выходной, будем называть входом (входной переменной) сети модулей.

Заметим, что любая переменная сети может быть входом многих модулей.

Нетрудно видеть, что ациклическая сеть модулей однозначно определяет значения выходных переменных для любого набора значений входных, т.е. также является вычислительным модулем. Такие модули будем называть модулями агрегирования.

Вопросы решения некоторых структурированных задач оптимизации рассматривались в разделе 2.

Рассмотрим некоторые примеры.

6.2.2. Задача оптимального проектирования поверхностей нагрева конвективной шахты.

Рассматриваются поверхности нагрева, расположенные в конвективной шахте парового котла – экономайзеры и воздухоподогреватели. Порядок расположения поверхностей нагрева в газоходе и их включение по обогреваемому теплоносителю считаются заданными. Параметры греющего теплоносителя на входе конвективной шахты, параметры воздуха на входе воздушного тракта, параметры воды на входе пароводяного тракта считаются

заданными. Поверхности нагрева должны снять заданное суммарное количество тепла. Для простоты будем рассматривать случай, когда в одном сечении газотока находится одна поверхность нагрева.

Необходимо определить распределение тепловосприятий между поверхностями нагрева и значения их конструктивных характеристик.

В данной задаче для поверхностей нагрева учитывается только конвективный теплообмен. Тепловосприятия поверхностей нагрева выбираются в качестве независимых переменных.

Для рассматриваемой задачи должны выполняться уравнения материального баланса, определяющие расходы теплоносителей для каждой поверхности нагрева. Эти уравнения будем описывать вычислительным модулем "материальный баланс" и обозначать UnitMassBalance.

При заданных расходах могут быть определены энтальпии теплоносителей из уравнений теплового баланса. Эти уравнения будем описывать вычислительным модулем "тепловой баланс" и обозначать UnitHeatBalance.

С помощью понятия вычислительный модуль будем описывать объект «поверхность нагрева». Входами модуля «поверхность нагрева» будут конструктивные характеристики поверхности нагрева, входные значения параметров теплоносителей, балансное тепловосприятие $Q_{бал}$ поверхности нагрева. Выходами будут выходные значения параметров теплоносителей, расчетное тепловосприятие Q_p поверхности нагрева, температуры металла стенок труб в выделенных точках, коэффициент запаса прочности в этих точках, и др. Расчет значений выходных переменных производится в соответствии с принятыми методиками [171].

Для модуля «поверхность нагрева» будем использовать обозначение UnitPaket.

Уравнение теплообмена для поверхности нагрева состоит в

$$Q_{\text{бал}} = Q_p \cdot \quad (6.2)$$

Модуль системы уравнений для (6.2) будем обозначать UnitPaket_SE. Уравнение (6.2) решается относительно конструктивной характеристики "число петель" поверхности нагрева.

Расчет аэродинамических, гидравлических сопротивлений трактов реализуется вычислительным модулем UnitTrakt.

На схемах вычислительные модули будут представляться овалами, для модулей систем уравнений внутри овала будут указываться модули, реализующие функциональные зависимости системы уравнений.

С учетом принятых обозначений структура рассматриваемой задачи имеет вид, представленный на рис. 6.1.

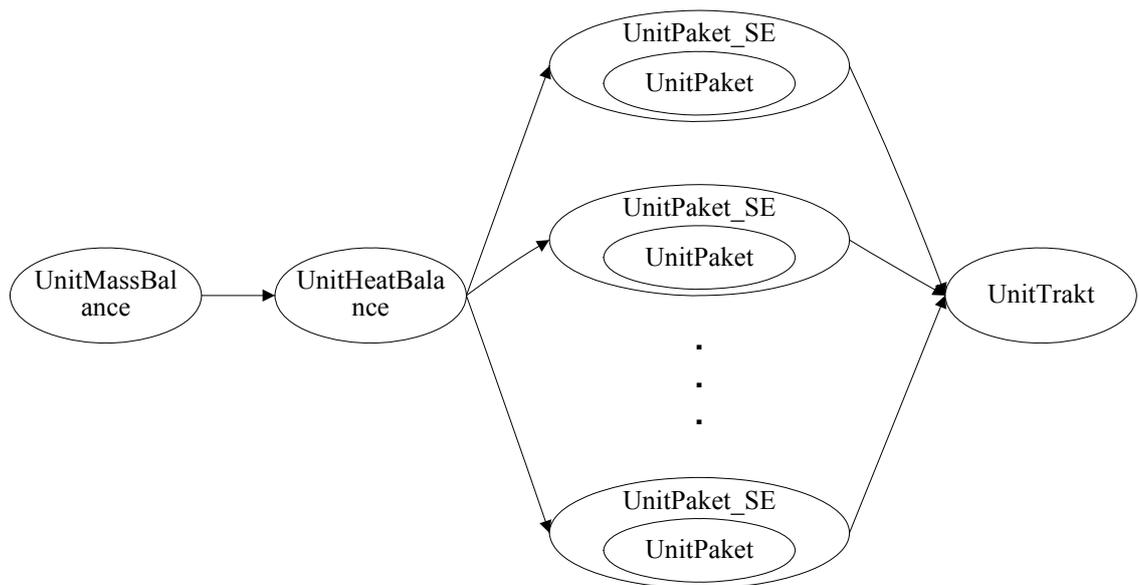


Рис. 6.1. Структура расчета функциональных зависимостей задачи оптимального проектирования поверхностей нагрева конвективной шахты

6.2.3. Задача оптимального проектирования многоступенчатого пароперегревателя

Рассматриваются поверхности нагрева, расположенные в горизонтальном газоходе – ширмовые и конвективные перегреватели. Порядок расположения поверхностей нагрева в газоходе и их включение по обогреваемому теплоносителю считаются заданными. Параметры греющего теплоносителя и поток лучистого тепла на входе горизонтального газохода, параметры пара на входе пароперегревателя считаются заданными. Поверхности нагрева должны снять заданное суммарное количество тепла. Рассматривается случай, когда в одном сечении газохода находится одна поверхность нагрева.

Необходимо определить распределение тепловосприятий между поверхностями нагрева и значения их конструктивных характеристик.

В отличие от предыдущей задачи, для поверхностей нагрева необходимо учитывать и конвективный, и лучистый теплообмен. Как и ранее тепловосприятия (тепло принятое паром) поверхностей нагрева выбираются в качестве независимых переменных.

В связи с наличием дополнительных уравнений, связанных с лучистым теплообменом, система нелинейных уравнений рассматриваемой задачи не распадается на независимые подсистемы, как это было в предыдущей задаче. Тем не менее, структура задачи может быть представлена в виде, отображенном на рис. 6.2.

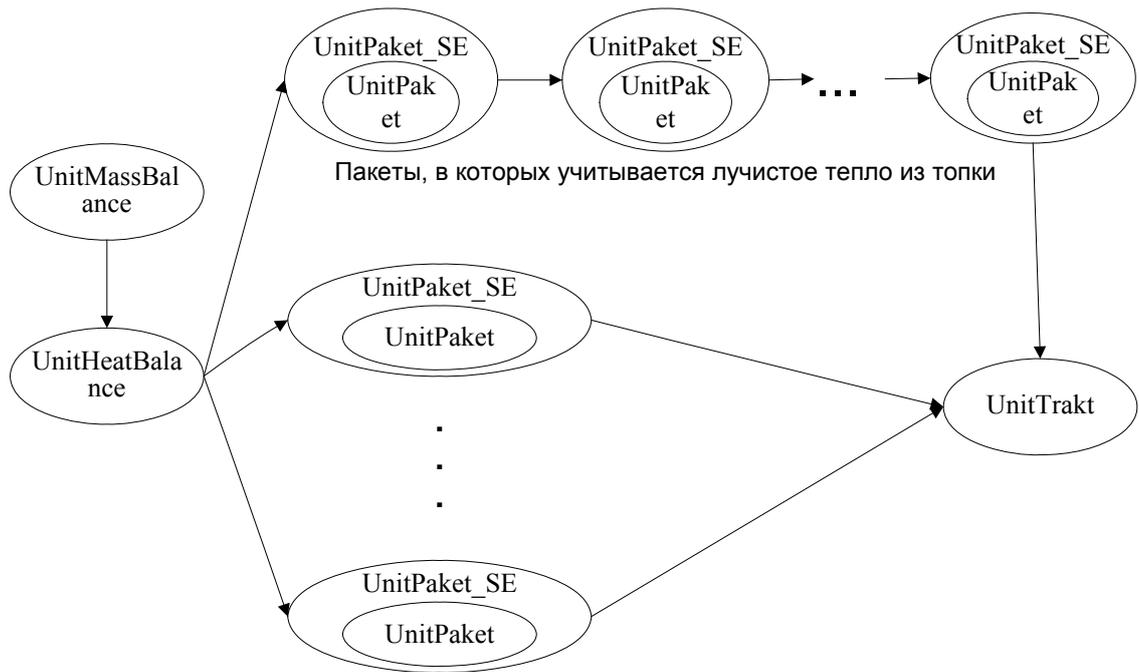


Рис. 6.2. Структура задачи оптимального проектирования многоступенчатого пароперегревателя

6.2.4. Задача оптимального проектирования энергетического котлоагрегата в целом

В задачах, которые рассматривались выше, предполагалось, что конструктивные характеристики для всех поверхностей нагрева варьируются при поиске оптимального решения. Это позволяло в качестве независимых переменных выбирать тепловосприятия, в результате чего системы уравнений задач распадались на слабосвязанные подсистемы.

При проектировании котлоагрегата в целом такое предположение неверно. Конструкция некоторых поверхностей нагрева должна быть зафиксирована. В частности, это относится к газоповоротной камере. Учет этих обстоятельств приводит к существенному усложнению структуры задачи.

Кроме поверхностей нагрева, которые рассматривались ранее, модель котлоагрегата должна включать объекты, связанные с топкой. Модель также должна учитывать переменные управления (впрыски питательной воды, рециркуляцию газов), используемые для поддержания режимов функционирования поверхностей нагрева в заданных границах.

Порядок включения всех элементов котла по греющему и обогреваемому теплоносителям определяется тепловой схемой парового котла и считается фиксированным.

От переменных управления зависит материальный баланс котла. Эти переменные включаются в вычислительный модуль *UnitMassBalance*, выходом которого являются расходы теплоносителей по всем элементам котла.

Расчет тепловосприятий экранов топки производится по принятым методикам, при этом считаются заданными расход и температура t_{ex}^o воздуха на входе топки и оценочная температура $t_{вых}^o$ газа на выходе из топки. После расчета тепловосприятий экранов определяется расчетная температура $t_{вых}^p$ газа на выходе из топки. В ходе расчета топки варьируется оценочная температура газа на выходе до тех пор, пока она не совпадет расчетной температурой на выходе, т.е. решается уравнение

$$t_{вых}^o = t_{вых}^p \quad (6.3)$$

Вычислительный модуль, в котором производится расчет тепловосприятий экранов топки, обозначим *UnitLight*

Вычислительный модуль, в котором выполняется расчет топки (решается уравнение (6.3)), обозначим *UnitТопка*.

Выполняемые расчеты состоят из нескольких этапов.

На первом этапе выполняется расчет топки при заданной температуре воздуха на входе топки, задаются значения балансовых тепловосприятий для всех поверхностей нагрева. Для поверхностей нагрева с фиксированными

конструктивными характеристиками эти значения будут оценочными и на последующих этапах должны уточняться для выполнения уравнений теплообмена. Для поверхностей нагрева с варьируемыми конструктивными характеристиками эти величины являются независимыми переменными, и на последующих этапах для выполнения уравнений теплообмена варьируются соответствующие конструктивные характеристики.

При заданных значениях балансовых тепловосприятий определяются температуры теплоносителей для всех поверхностей нагрева, а для поверхностей нагрева с фиксированными характеристиками определяются также расчетные тепловосприятия. Для пакетов (поверхностей нагрева), в которых учитывается лучистое тепло из топки, необходимо также выполнять расчет пакетов с варьируемыми характеристиками. Структура вычислений первого этапа представлена на рис. 6.3. В дальнейших структурных схемах совокупность этих вычислительных модулей будет представляться агрегированным модулем *UnitHeatAgregate*.

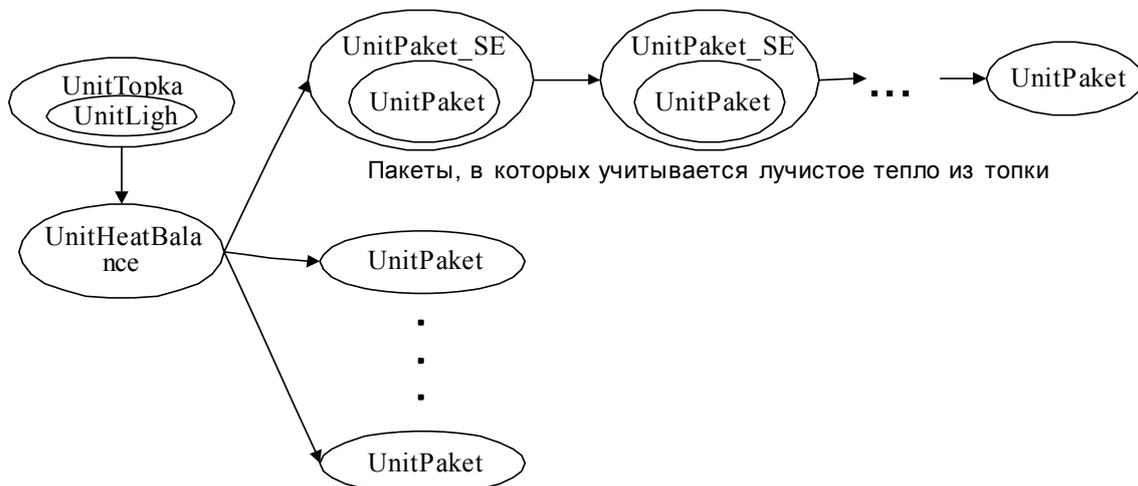


Рис. 6.3. Структура агрегированного модуля *UnitHeatAgregate*

На втором этапе за счет варьирования оценочных тепловосприятий для поверхностей нагрева с фиксированными конструктивными

характеристиками решается система уравнений теплообмена для этих поверхностей

$$Q_{бал}^i = Q_p^i, i \in I_{фикс},$$

где $I_{фикс}$ – множество поверхностей нагрева с фиксированными конструктивными характеристиками.

Вычислительный модуль, реализующий решение такой системы уравнений, будем обозначать *UnitHeatCalc*. На каждой итерации при решении системы уравнений для определения расчетных тепловосприятий используется агрегированный модуль *UnitHeatAgregate*.

На третьем этапе решается система уравнений теплообмена для поверхностей нагрева с варьируемыми конструктивными характеристиками, также рассчитываются аэродинамические (гидравлические) сопротивления трактов. Структура вычислений *третьего* этапа представлена на рис. 6.4. Совокупность этих вычислительных модулей будем называть агрегированным модулем *UnitBoiler*.

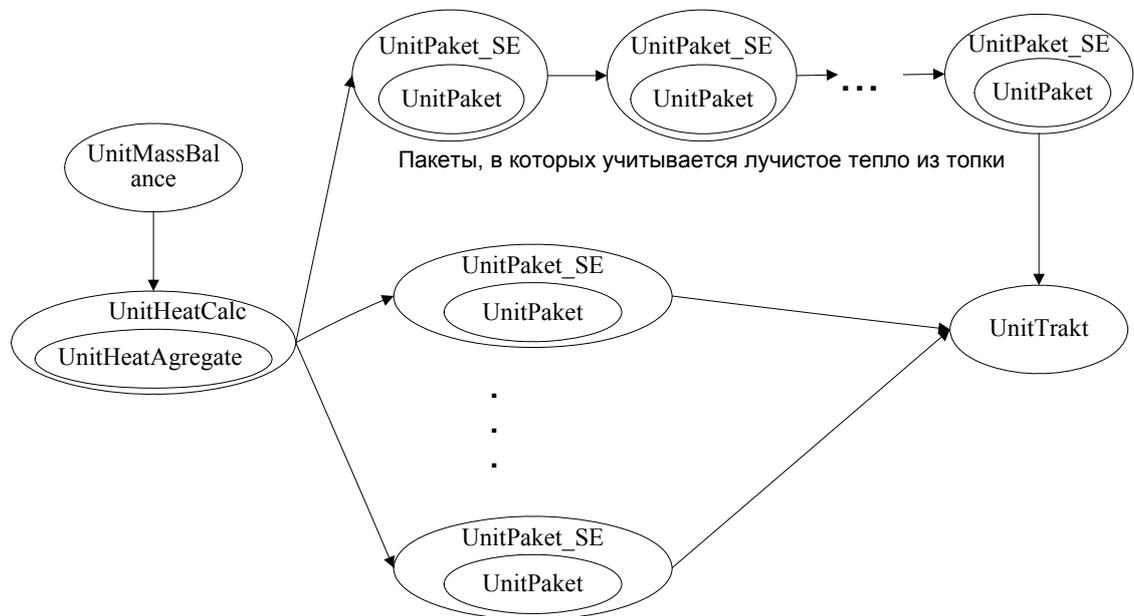


Рис. 6.4. Структура расчета функциональных зависимостей задачи оптимального проектирования энергетического котлоагрегата

В приведенных структурных схемах отражены основные вычислительные модули, используемые при расчете функциональных зависимостей котлоагрегата. Число модулей *UnitPaket* (поверхность нагрева) в агрегированном модуле *UnitHeatAgregate* в простейшем случае равно числу поверхностей нагрева с фиксированными конструктивными характеристиками. Соответственно в агрегированном модуле *UnitBoiler* число вычислительных модулей *UnitPaket_SE* равно числу поверхностей нагрева с варьируемыми конструктивными характеристиками.

В случае, когда в одном сечении газохода может находиться не более одной поверхности нагрева, каждый модуль *UnitPaket_SE* соответствует системе уравнений с двумя переменными для чисто конвективных пакетов, и системе уравнений с тремя переменными для ширмовых пакетов.

В случае, когда в одном сечении газохода может находиться две поверхности нагрева, число переменных в системах уравнений, соответствующих модулям *UnitPaket_SE*, увеличивается на единицу.

Особенностями рассмотренных структурных схем является следующее:

- базовыми объектами являются относительно простые (с точки зрения организации вычислений) вычислительные модули: *UnitPaket* (расчеты, связанные с поверхностями нагрева), *UnitMassBalance*, *UnitHeatBalance* (системы линейных уравнений материального и теплового балансов);
- из этих простых модулей порождаются новые более сложные модули с помощью формирования систем уравнений и операций агрегирования;
- отдельные блоки систем уравнений состоят из небольшого числа уравнений ($\sim 1-4$) и могут быть эффективно решены при фиксированных значениях независимых переменных;
- операции агрегирования и формирования систем уравнений могут применяться рекуррентно.

Указанные особенности позволяют выполнить редукцию исходной задачи, которая описывается сложной сетью вычислительных модулей, к эквивалентной задаче существенно меньшей размерности, формально описываемой одним результирующим функциональным модулем с заданным списком входных и выходных переменных. Возможные подходы к такой редукции рассматривались в подразделе 2.1.

Уменьшение размерности задачи производится за счет усложнения вычислений на каждой итерации оптимизационных алгоритмов. Ограничения, определяющие области определения используемых функций, сравнительно просты: в каждой поверхности нагрева температура греющего теплоносителя должна быть выше температуры обогреваемого. Для начальной точки, задаваемой пользователем, эти ограничения, как правило, выполняются.

Редуцированная задача имеет размерность: число переменных – 50 ~ 200, число ограничений-неравенств имеет такой же порядок. Эта задача может быть представлена в следующем виде: найти

$$f^0(y, v, x, u) \rightarrow \min_{x, u}, \quad (6.4)$$

$$F(y, v, x, u) \leq b, \quad (6.5)$$

$$y = H(x, u), \quad (6.6)$$

$$v = G(x, u), \quad (6.7)$$

$$x \in X, u \in U, \quad (6.8)$$

где

x – независимые переменные (конструктивные характеристики),
 u – независимые переменные (тепловосприятости поверхностей, управляющие параметры используемого режима эксплуатации – впрыски, отборы теплоносителей, расход топлива),

y – расчетные конструктивные характеристики (число петель поверхностей нагрева, заходность для ширм, ...),

v – расчетные показатели теплоносителей,

F, G, H – вектор-функции,

f^0 – целевая функция,

множества X, U определяют ограничения сверху и снизу на значения независимых переменных.

Задача (6.4)–(6.7) невыпуклая негладкая. Для ее решения используются точные штрафные функции по ограничениям (6.5) и специальная модификация r -алгоритма. В общем случае можно говорить только о поиске локального минимума задачи.

В качестве целевой функции (6.4) может использоваться суммарная стоимость поверхностей нагрева, материалоемкость, или приведенные затраты.

Схемы вычислений, подобные показанным на рис. 6.1–6.4, позволяют детально учитывать специфику решаемых задач для сложных объектов различного типа, эффективно реализовывать различные варианты поиска решений. Однако программная реализация этих схем связана со значительными трудностями, а известные системы оптимизации и моделирования [26, 31] не всегда обеспечивают их преодоление.

6.3. Модель учета альтернативных режимов эксплуатации

При построении модели оптимального проектирования, учитывающей альтернативные режимы эксплуатации, важным является, как и ранее, выбор независимых переменных.

Естественно в качестве независимых переменных выбирать конструктивные характеристики котлоагрегата. При этом на каждой итерации оптимизационных алгоритмов для всех учитываемых режимов

должен выполняться тепловой расчет котлоагрегата, после чего могут быть выполнены расчеты всех необходимых показателей. Формально тепловой расчет сводится к решению системы нелинейных уравнений. С использованием введенных понятий структуру системы уравнений для каждого режима также можно представить сетью функциональных блоков. Однако в данном случае эта сеть уже не будет ациклической, и для решения системы уравнений должны разрабатываться специальные эффективные методы.

Альтернативным является включение в состав независимых переменных тепловосприятий поверхностей нагрева для каждого альтернативного режима эксплуатации. Это позволяет использовать возможности структурного описания расчетов для каждого режима, описанные в предыдущем подразделе. При этом для каждого режима определяется свое значение расчетной конструктивной характеристики поверхностей нагрева, и эти значения оказываются разными для разных режимов. Такой подход требует введения в модель дополнительных ограничений, обеспечивающих равенство расчетных конструктивных характеристик для разных режимов, и математическая модель оптимального проектирования принимает вид

$$f^0(y, v, x, u) \rightarrow \min_{x, u}, \quad (6.9)$$

$$F^k(y^k, v^k, x, u^k) \leq b^k, \quad k = 1, \dots, K, \quad (6.10)$$

$$y^k = H^k(x, u^k), \quad k = 1, \dots, K, \quad (6.11)$$

$$v^k = G^k(x, u^k), \quad k = 1, \dots, K, \quad (6.12)$$

$$y^k = y^s, \quad k, s = 1, \dots, K, \quad (6.13)$$

$$x \in X, \quad u^k \in U^k, \quad k = 1, \dots, K, \quad (6.14)$$

где

$$y = (y^1, \dots, y^K), \quad v = (v^1, \dots, v^K), \quad u = (u^1, \dots, u^K),$$

x – независимые переменные (конструктивные характеристики), общие для всех режимов

u^k – независимые переменные (тепловосприятости поверхностей, управляющие параметры используемого режима эксплуатации – впрыски, отборы теплоносителей, расход топлива) режима k ,

y^k – расчетные конструктивные характеристики (число петель поверхностей нагрева, заходность для ширм, ...) режима k ,

v^k – расчетные показатели теплоносителей режима k ,

F^k, G^k, H^k – вектор-функции режима k ,

f^0 – целевая функция,

множества X, U^k определяют ограничения сверху и снизу на значения независимых переменных.

В качестве целевой функции используется одна из целевых функций номинального режима, рассматриваемых ранее. Множества U^k должны содержать достаточно жесткие ограничения по температуре уходящих газов, что однозначно определяет ограничения на КПД котлоагрегата для каждого режима.

Для решения задачи (6.9)–(6.14) используются точные штрафные функции по ограничениям (6.10), (6.13) и модификация r -алгоритма.

6.4. Описание прототипа системы оптимального проектирования котлоагрегатов

Проектирование энергетических котлоагрегатов – сложный трудоемкий процесс, связанный с обработкой больших объемов информации, с выполнением различного вида расчетов, подготовкой проектно-конструкторской документации.

В настоящее время для выполнения наиболее трудоемких нормативных расчетов используются отдельные программы, как правило, хорошо

апробированные и удовлетворяющие существующим потребностям. Однако в силу их разрозненности возникают проблемы повторного ввода при подготовке исходных данных для этих программ и использования результатов их расчетов. Многие виды расчетов не автоматизированы и выполняются вручную.

Для автоматизации рутинных трудоемких операций процесса проектирования необходима организация единой базы проектов, в которой должна накапливаться в стандартизованном виде и обрабатываться вся информация, необходимая для проектирования.

Для выработки рациональных технических решений необходимо применять современные средства математического моделирования и оптимизации, позволяющие учитывать различные технические, технологические и экологические ограничения в процессе проектирования.

Решение проблем такого рода требует создания системы комплексной автоматизации процессов проектирования энергетических котлоагрегатов. Описание прототипа такой системы приводится ниже.

6.4.1. Состав разрабатываемого программного обеспечения

Состав разрабатываемого прототипа программного обеспечения (для котла ТП-80 докритического давления) приведен на рис. 6.5.

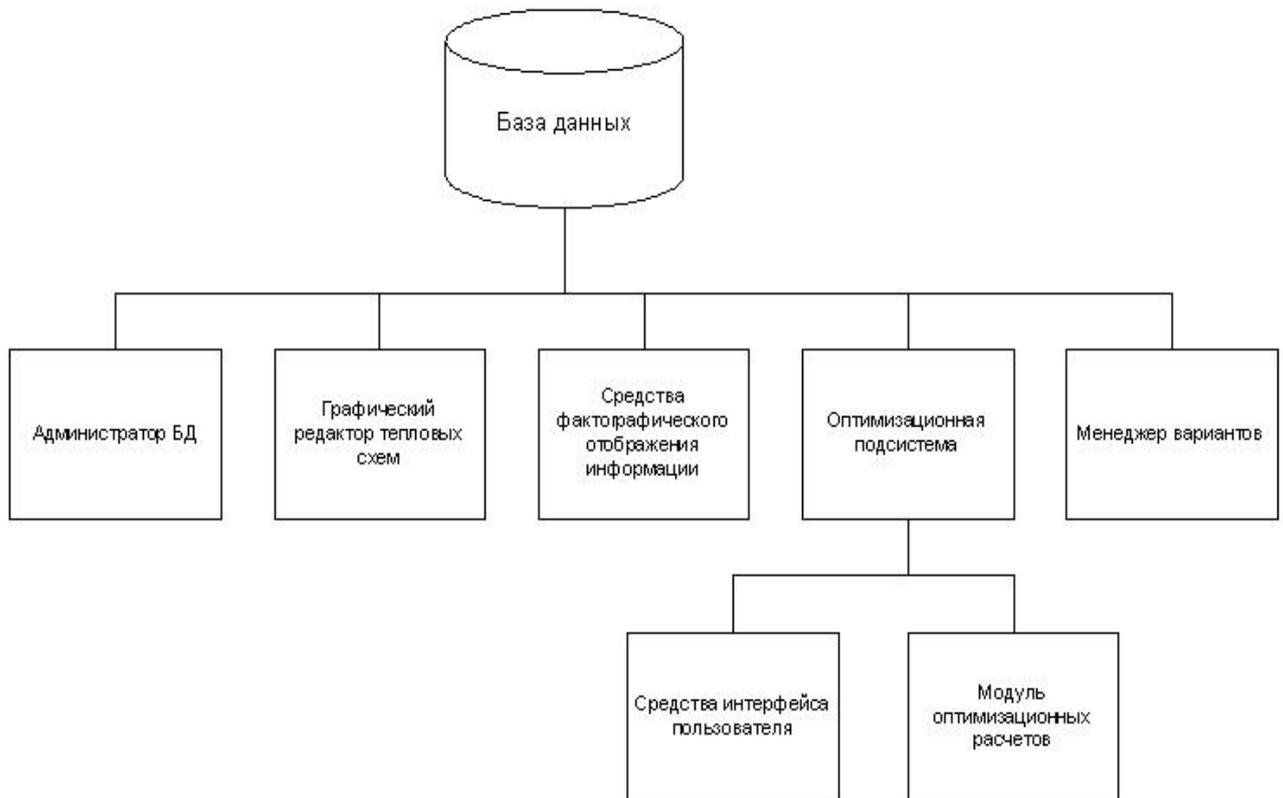


Рис. 6.5. Структура прототипа оптимального проектирования котлоагрегатов

Назначение компонент и среда разработки:

база данных (БД) должна содержать всю необходимую для проектирования котлоагрегатов информацию, поддерживать многовариантный процесс проектирования, должна обеспечивать гибкость системы при добавлении новых типов элементов конструкций, расширения состава характеристик каждого типа, среда разработки – СУБД ORACLE;

администратор БД должен поддерживать ведение справочников системы, обеспечивать гибкость системы при добавлении новых типов элементов конструкций, расширения состава характеристик каждого типа, среда разработки – DELPHI;

графический редактор предназначен для формирования тепловых схем проектируемых котлоагрегатов в графическом режиме, среда разработки – DELPHI;

средства фактографического отображения информации предназначены для отображения и редактирования в табличном виде всех данных, содержащихся в БД, среда разработки – DELPHI;

оптимизационная подсистема предназначена для решения задач оптимального проектирования котлоагрегатов;

средства интерфейса пользователя оптимизационной подсистемы предназначены:

- для формирования математической модели оптимизационной задачи (определения состава варьируемых переменных и ограничений, которые учитываются при решении оптимизационной задачи), отображения результатов расчетов для анализа, среда разработки – DELPHI;
- для графического отображения хода оптимизационного процесса, среда разработки – Microsoft VISUAL C++;

модуль оптимизационных расчетов предназначен для решения задачи оптимального проектирования на основе сформированной математической модели, среда разработки – Microsoft VISUAL C++;

менеджер вариантов предназначен для отображения и манипулирования деревом вариантов, поддержки многовариантного процесса оптимального проектирования, среда разработки – DELPHI.

6.4.2. База данных

Структура базы данных ориентирована на такую технологию работы конструктора, при которой в ходе проектирования создается множество различных вариантов создаваемой конструкции. При этом различаются несколько этапов разработки вариантов:

- выбор структуры конструкции;
- выбор значений параметрических характеристик;

- выбор режимов, в которых должна работать конструкция;
- выполнение расчетов для каждого режима.

На каждом этапе конструктор может создавать различные варианты. Создаваемые варианты связаны определенными отношениями иерархии, отражающими поэтапность процесса проектирования:

общие данные по котлу =>> структурный вариант =>> параметрический вариант =>> режимный вариант =>> вариант расчета

С каждым предшествующим вариантом может быть связано несколько последующих. Вариант расчета может быть получен, если определены все предшествующие варианты иерархии. Для каждого этапа (варианта) определен свой состав используемой информации, данные предшествующего варианта являются общими для всех связанных с ним последующих вариантов.

При работе с базой данных должны выполняться определенные правила. Доступ к базе данных имеют только зарегистрированные пользователи. Каждый пользователь всегда работает с конкретным набором вариантов (ветвью дерева вариантов). Для регистрации пользователей и хранения информации о выбранной ветви дерева вариантов используется специальная управляющая таблица. Все программные компоненты системы работают с этой таблицей.

Одновременно с базой данных могут работать несколько пользователей. Для организации совместной работы предусмотрены возможности разделения доступа к данным – для каждого варианта определен владелец варианта и список пользователей, которые могут просматривать информацию варианта. Каждый пользователь может изменять данные только тех вариантов, владельцем которых он является, остальные варианты ему недоступны. При необходимости владелец может передать

вариант другому пользователю, после чего вариант становится недоступным старому владельцу.

База данных включает всю информацию, необходимую для выполнения основных расчетов, решения оптимизационных задач, формирования эскизов, подготовки отчетов.

Одной из существенных проблем при проектировании базы данных является разнородность элементов, из которых состоит котлоагрегат:

элемент тепловой схемы может быть пакетом, впрыском, отбором, позицией газохода и др.;

поверхность нагрева – ширмой, конвективной поверхностью, воздухоподогревателем, оребренной поверхностью и т.п.

Элементы каждого типа описываются своим составом характеристик.

Дополнительным является требование обеспечения гибкости базы данных – система должна включать эффективные средства добавления новых типов элементов, расширения состава характеристик каждого типа.

Для преодоления указанных проблем в базу данных была включена система справочников, определяющая для каждого типа элементов свой состав характеристик. Значение каждой характеристики хранится в отдельной записи, а не как поле в записи соответствующего элемента. Справочники могут изменяться как при разработке базы данных, так и на этапе эксплуатации. Однако эти изменения может производить только администратор базы данных по согласованию с разработчиками прикладных программ.

Принятая организация базы данных обеспечивает высокую гибкость при работе с информацией, но существенно усложняет реализацию базы данных – при определении типа нового элемента тепловой схемы, или новой поверхности нагрева (или изменении типа) автоматически должен формироваться (изменяться) список всех характеристик этого типа.

Аналогично, при изменении классификатора должны изменяться списки характеристик для всех элементов.

База данных содержит более 200 таблиц и поддерживает достаточно сложные правила целостности между таблицами.

6.4.3. Администратор базы данных

Программная компонента **администратор БД** предназначена для поддержки ведения справочников системы и выполнения ряда дополнительных функций – регистрации новых пользователей, формирования фильтров для отображения данных.

Все компоненты котлоагрегата (элементы тепловой схемы, поверхности нагрева, участки газохода, тракты тепловой схемы, ...), топливо, котлоагрегат в целом (общие характеристики котлоагрегата) описываются списками характеристик, которые могут изменяться как при разработке базы данных, так и при развитии системы. В связи с наличием иерархии вариантов, отражающей поэтапность процесса проектирования, характеристики разбиты на группы, относящиеся к тому или другому уровню вариантов (структурный, параметрический, режимный, расчетный или уровень математического моделирования).

Состав характеристик некоторых объектов (элементы тепловой схемы, поверхности нагрева, топливо, ...) зависит от типа объекта. Это в свою очередь требует введения справочников по типам объектов.

6.4.4. Менеджер вариантов

Компонента предназначена для отображения и манипулирования деревом вариантов, поддержки многовариантного процесса оптимального проектирования. На рис. 6.6 представлено главное окно менеджера вариантов.

В левой панели окна отображается дерево вариантов в стандартной для WINDOWS форме. С вариантом каждого уровня связан некоторый поясняющий комментарий, который формируется пользователем. Необходимость такого комментария обусловлена большим числом вариантов, которые создаются в процессе проектирования. Непосредственно в дереве вариантов отображается начальная часть комментария, полный текст комментария отображается в правой части окна менеджера вариантов при выборе в дереве соответствующего варианта. Также в правой части отображается дополнительная информация, полностью идентифицирующая вариант.

Для вариантов каждого уровня могут быть вызваны функции добавления, удаления и копирования текущего варианта. При удалении и копировании удаляются и копируются как текущий вариант, так и все дочерние варианты.

Важным элементом системы проектирования является **Управляющая таблица**. В этой таблице каждому зарегистрированному пользователю выделяется строка, в которую заносится информация о ветви дерева вариантов, с которой работает пользователь, и о ветви, с которой производится сравнение текущей ветви вариантов. При отображении данных проектируемой конструкции эта информация автоматически выбирается из управляющей таблицы. Для вариантов расчета в правой части окна отображаются дополнительные кнопки, которые позволяют заносить в управляющую таблицу данные о текущей ветви вариантов и о ветви для сравнения. Просмотреть содержание управляющей таблицы можно с помощью пункта меню «**Управляющая таблица**».

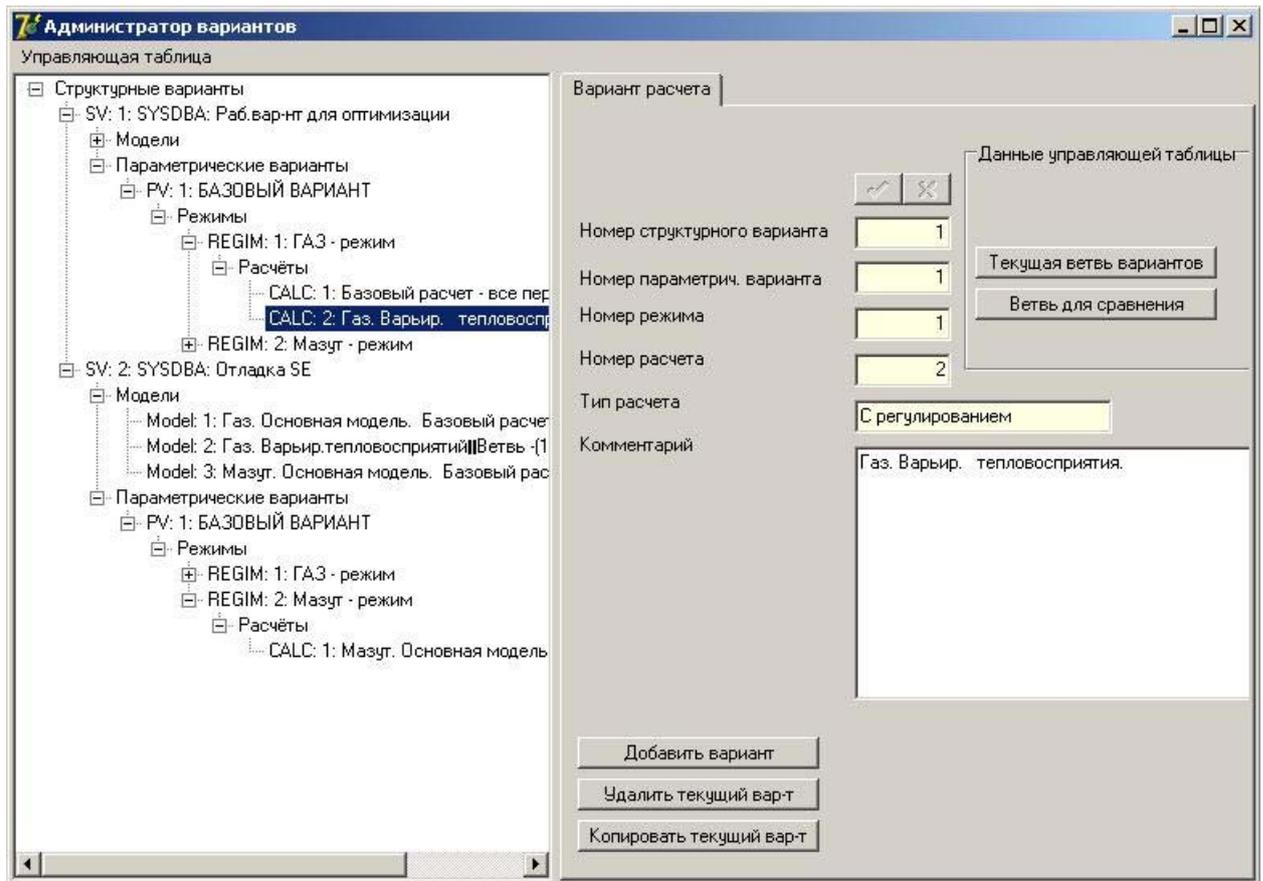


Рис. 6.6. Главное окно менеджера вариантов

Для структурных вариантов определены владелец варианта, т.е. пользователь, который может изменять данные текущего и всех дочерних вариантов, а также список зарегистрированных пользователей с правом просмотра текущего и дочерних вариантов.

Владелец варианта может передать свои права другому зарегистрированному пользователю. При этом он теряет свои права. Т.е. только один пользователь может изменять данные варианта, и только он несет ответственность за эти данные.

6.4.5. Графический редактор

Предназначен для формирования тепловых схем проектируемых котлоагрегатов в графическом режиме. Пользователь формирует (рисует) тепловую схему, используя графические символы стандартных элементов и устанавливая связи между элементами. При этом автоматически заполняются

соответствующие таблицы базы данных. Предусмотрено несколько форм отображения и редактирования данных, различные функции, повышающие производительность пользователя, средства автоматической трассировки схем для их представления в различном формате. Имеется возможность отображать в тепловых схемах различную информацию из базы данных (как исходную, так и результаты расчетов), а так же делать фиксированные надписи. Общий вид интерфейса графического редактора приведен на рис 6.7. Каждый тракт тепловой схемы отображается в отдельном окне. Для работы с окнами предусмотрены стандартные возможности, принятые в WINDOWS: изменение размеров окон, прокрутка изображения в окне, изменение масштаба изображения. Вызов функциональных процедур осуществляется с помощью кнопок панели инструментов.

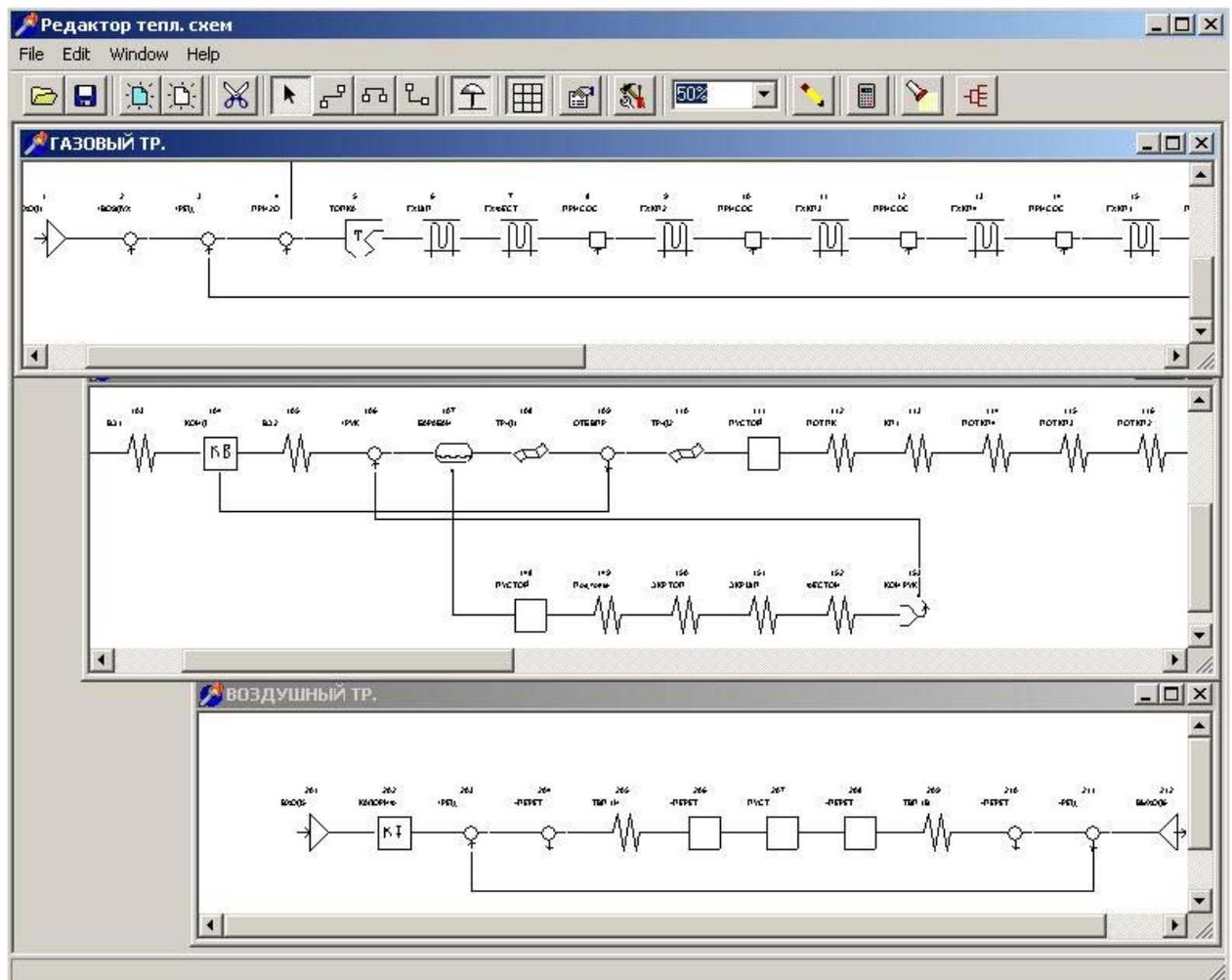


Рис. 6.7. Общий вид интерфейса графического редактора

6.4.6. Средства фактографического отображения данных

Данный программный модуль предназначен для детализированного отображения данных о котлоагрегате, необходимых для основных расчетов, выполняемых в ходе проектирования. Эти данные используются как для выполнения оптимизационных расчетов, так и для выполнения расчетов сертифицированными программами.

При отображении данных, содержащихся в БД, всегда используется информация из управляющей таблицы, относящаяся к текущему пользователю. Т.е. отображаются данные, относящиеся к ветви, зафиксированной в учетной записи текущего пользователя. Такая организация работы пользователя принята для минимизации действий по идентификации варианта, данные о котором необходимо отобразить.

Состав отображаемых данных зависит от типа поверхности нагрева и определяется справочником характеристик для типов поверхностей нагрева. При изменении справочника характеристик автоматически изменяется состав отображаемых данных. Это обеспечивает необходимую гибкость при развитии системы проектирования.

6.5. Описание оптимизационной подсистемы

В разделе 6.2 на содержательном уровне были введены понятия вычислительных модулей, с помощью которых описывались структуры задач оптимального проектирования. При создании оптимизационной подсистемы разработаны программные средства, позволяющие эффективно реализовать комплексные расчеты всех необходимых показателей котлоагрегатов с учетом описанной структуры задач. Разработка выполнялась в среде языка C++ и основана на объектно-ориентированной парадигме программирования.

6.5.1. Рекурсивная библиотека классов моделирования сложных объектов

В соответствии с объектно-ориентированным подходом построена иерархия классов, используемых при описании задач оптимизации сложных технических объектов. Совокупность основных классов этой иерархии будем называть библиотекой RBL. В конкретных приложениях прикладные классы формируются путем наследования от основных классов библиотеки RBL.

1. В качестве базового класса будем использовать класс CUnit, реализующий понятие вычислительного модуля. Класс CUnit описывается совокупностью входных переменных $x \in R^n$ и реализует вычисление значений некоторой вектор-функции $y(x): R^n \rightarrow R^m$. Если вектор-функция $y(x)$ определена на ограниченном множестве, то должны использоваться процедуры продолжения функций на все пространство, которые рассматривались в разделе 2. В дальнейшем предполагается, что все используемые функции определены при любых значениях их аргументов.

Часть компонент вектор-функции $y(x)$ используются для формирования ограничений вычислительного модуля, которые должны учитываться при решении оптимизационных задач. Например, для модуля «Поверхность нагрева» ограничения формируются для температур стенки труб в заданных точках, для коэффициентов запаса прочности, скоростей теплоносителей. Будем считать, что эти ограничения имеют вид

$$y_i(x) \leq 0, i \in I, \quad (6.15)$$

где $I \in \{1, \dots, m\}$. В число этих ограничений включается также ограничение допустимости текущей точки.

Кроме вычисления значений выходов $y(x)$ к числу основных функций класса CUnit принадлежат функции вычисления градиентов выходов y по входам x . Класс также содержит ряд вспомогательных функций контроля и управления вычислениями.

Класс CUnit является абстрактным (в терминологии C++) классом и определяет интерфейс основных функций. Конкретная реализация этих функций производится в дочерних классах.

2. Класс агрегирования CUnitAgregate наследуется от класса CUnit и позволяет представить как одно целое совокупность (сеть) связанных между собой (выходы одних являются входами других) блоков – объектов типа CUnit, или порожденных от класса CUnit. Блоки i и j связаны между собой дугой (i, j) , если некоторые выходы блока i являются входами блока j . Сеть блоков должна быть ациклической.

Выходами модуля CUnitAgregate будет объединение всех выходов блоков, входами – объединение входов блоков, не являющиеся выходами других блоков.

Расчет выходов модуля CUnitAgregate осуществляется путем последовательного (определяемого упорядочением сети) расчета отдельных блоков. Аналогично выполняется расчет градиентов (почти градиентов [180]) выходов модуля CUnitAgregate.

Объекты, полученные в результате агрегирования, опять могут использоваться для последующего агрегирования.

3. Описание и решение систем нелинейных уравнений реализуется классом CUnitSysEq, который также наследуется от класса CUnit. Класс CUnitSysEq содержит в себе дополнительно объект pBlock типа CUnit, входами которого является вектор z , выходами – вектор $v(z)$. Вектор z представляется в виде $z = (x, y)$, где x – параметры системы уравнений, y – переменные, относительно которых решается система уравнений. Часть компонент вектора $v(z)$ используется как вектор-функция $g(x, y)$, описывающая систему нелинейных уравнений (6.1). Класс CUnitSysEq реализует решение системы уравнений (6.1), его входами является вектор x ,

выходами – векторы $y(x)$ и $v(x, y(x))$, где $y(x)$ – решение системы уравнений (6.1).

Членами класса CUnitSysEq являются также матрицы A_x , A_y и вектор b , определяющие область локализации $\Omega = \{(x, y): A_x x + A_y y \leq b\}$ решений системы уравнений (6.1). Если при заданном x в области Ω решений не существует, выполняется поиск регуляризованного решения и вычисления функции $d(x)$, определяемой соотношением (2.12). Функция $d(x)$ также является выходом класса CUnitSysEq.

Выходы класса CUnitSysEq определены при любых значениях входов.

4. Класс CUnitMax также наследуется от класса CUnit. Содержит дополнительно один блок (объект типа CUnit), выходы которого используются для формирования задачи математического программирования. Класс реализует методы решения этой задачи. В настоящее время содержит некоторые модификации r -алгоритма [176]. Для решения задач с ограничениями используются негладкие штрафные функции.

5. Библиотека RBL также содержит классы для решения систем линейных уравнений, вспомогательные классы и функции.

Средствами библиотеки эффективно описываются

блочные системы нелинейных уравнений;

ациклические сети блоков уравнений – каждый блок описывается объектом типа CUnitSysEq, сеть блоков - объектом типа CUnitAgregate;

другие структурированные объекты и задачи.

В библиотеке предусмотрены механизмы контроля актуальности текущих значений переменных, что позволяет избежать повторных вычислений.

6.5.2. Описание методик теплотехнических расчетов

Описание расчет конвективных и ширмовых поверхностей нагрева приводится в Приложении А.

6.5.2.1. Уравнения теплового и материального баланса

Основой теплового расчета является тепловая схема котлоагрегата, которая описывает состав компонент и порядок омывания их теплоносителями [171, 146].

Уравнения материального, теплового баланса и баланса лучистого тепла записываются для каждого элемента тепловой схемы и связывают входы и выходы этого элемента. Входами/выходами элементов являются потоки теплоносителей, которые характеризуются давлением, расходом, энтальпией и температурой. Для элементов горизонтального газохода входами/выходами также являются потоки лучистого тепла из топки.

Если элементы k и l тепловой схемы связаны дугой, исходящей из k в l , то выход элемента k является входом элемента l .

Уравнения материального баланса являются линейными и включают соотношения баланса воды и водяного пара, баланса воздуха и баланса дымовых газов.

При формулировке уравнений теплового баланса принято упрощающее предположение, которое касается учета дополнительных и ограждающих пакетов – предполагается, что в каждой позиции газохода такие пакеты получают фиксированную долю от тепла, которое снимают основные пакеты. Значение этой доли задается для каждой позиции газохода, распределение тепла между разными дополнительными пакетами считается фиксированным и определяется в соответствии с пропорциями площадей этих пакетов. Принятое предположение позволяет существенно снизить размерность формируемой оптимизационной задачи.

Уравнения теплового баланса для пароводяного тракта записываются относительно величин (полных энтальпий) $I = D * i$, где D – расход пара по дуге тепловой схемы тракта, i – удельная энтальпия пара (воды) на этой дуге. При использовании таких переменных уравнения теплового баланса для пароводяного тракта становятся линейными относительно полных энтальпий и балансовых тепловосприятий $Q^{бал}$ (тепло, которое воспринимает пар).

Для элементов (топка, радиационные поверхности нагрева, ширма, перегреватели) тепловой схемы, в которых имеет место лучистый теплообмен, записываются дополнительно уравнения баланса лучистого тепла. Эти уравнения являются линейными относительно оценочных тепловосприятий лучистого тепла из топki $Q^{л.бал}$.

Уравнения материального, теплового баланса однозначно определяют характеристики теплоносителей на входах и выходах всех элементов тепловой схемы при фиксированных значениях следующих величин:

расхода сжигаемого топлива (считается заданным при решении оптимизационной задачи),

характеристик теплоносителей на входах пароводяных трактов,

температуры воздуха на входе воздушного тракта,

балансовых тепловосприятий $Q^{бал}$ и тепловосприятий лучистого тепла из топki $Q^{л.бал}$ для всех поверхностей нагрева,

величин отбора и впрыска пара (конденсата), отбора газа на рециркуляцию.

Для упрощения оптимизационной задачи в уравнения материального, теплового баланса добавляются дополнительные линейные ограничения-равенства:

фиксированные значения теплоносителей на выходах пароводяных трактов,

фиксированная температура на выходе из барабана,
 фиксированная температура воздуха на входе топки (эта величина является независимой переменной оптимизационной задачи).

В связи с введением дополнительных линейных ограничений-равенств часть балансовых тепловосприятий $Q^{бал}$ должна определяться из условий баланса тепла при решении рассматриваемой линейной системы уравнений.

Заметим, что уравнения материального и теплового баланса описываются разреженными матрицами, близкими к треугольным, и для их решения применимы эффективные специальные алгоритмы.

Тепловая схема котлоагрегата описывается классом *SHeatChart*. Этот класс поддерживает функции ввода исходных данных о тепловой схеме, формирования совокупности всех элементов, формирования системы линейных уравнений материального и теплового баланса, решения этой системы уравнений.

Описание уравнений материального и теплового баланса приведено в Приложении Б.

6.5.2.2. Расчет свойств жидкого и газообразного топлива, термодинамических свойств воздуха и уходящих газов

Расчет топлива производится в соответствии с нормами [171]:

Программы реализованы совместно с Харьковским ЦКБ «ЭНЕРГОПРОГРЕСС» в соответствии с нормами теплового расчета [171]. Выполняются расчеты для всех видов топлива, используемых на тепловых электростанциях, и их смесей. Рассматриваемые свойства топлива используются при расчете топки и при определении термодинамических свойств воздуха и уходящих газов.

Расчет свойств топлива реализуется в классах *SToplivo*, *SToplivoCoal*, *SToplivoGaz*, *SToplivoMazut*, *SMixtureOfTopl*.

Расчет термодинамических свойств воздуха и уходящих газов реализуется в классе *CGazAirAprox*. В отличие от норм [171] расчет термодинамических свойств уходящих газов выполняется приближенно – предполагается, что химический состав уходящих газов постоянный для всех участков газотока и совпадает с составом на выходе из топки. Погрешности определения энтальпии по температуре газа (и наоборот, температуры по энтальпии) при этих упрощениях не превышает 3%.

6.5.2.3. Расчет термодинамических свойств воды и водяного пара

Программы расчета реализуют расчеты термодинамических свойств воды и водяного пара в докритической области. В базу данных введены таблицы из норм теплового расчета [171]:

Таблица 5 "Коэффициент динамической вязкости воды и водяного пара";

Таблица 6 "Коэффициент теплопроводности воды и водяного пара";

Таблица 7 "Критерий Pr физических свойств воды и водяного пара";

Таблица 23 "Удельные объемы и энтальпии сухого насыщенного пара и воды на кривой насыщения";

Таблица 24 "Удельные объемы и энтальпии с воды";

Таблица 25 "Удельные объемы и энтальпии перегретого пара при докритическом давлении".

При выполнении расчетов задается интервал давлений, для которого выполняются расчеты, из базы данных в оперативную память выбирается информация из указанных таблиц для заданного интервала, расчет выполняется путем линейной интерполяции табличных значений.

Расчеты термодинамических свойств воды и водяного пара реализуются в классе *CWaterSteam*.

6.5.2.4. Тепловой расчет в топке

Расчет тепловосприятий экранов топки производится по принятым методикам, при этом считаются заданными расход и температура t_{ex}^o воздуха на входе топки и оценочная температура $t_{вых}^o$ газа на выходе из топки. После расчета тепловосприятий экранов определяется расчетная температура $t_{вых}^p$ газа на выходе из топки. В ходе расчета топки варьируется оценочная температура газа на выходе до тех пор, пока она не совпадет расчетной температурой на выходе, т.е. решается уравнение

$$t_{вых}^o = t_{вых}^p. \quad (6.16)$$

Классом, выполняющим расчет температуры газов $t_{вых}^p$ на выходе топки, является класс *CUnitLight_T*.

Входными параметрами этого класса являются:

- температура(энтальпия) воздуха (газа) на входе в топку;
- объем воздуха (газа) на входе в топку;
- оценочная температура газа на выходе топки;
- расход топлива;
- конструктивные характеристики, определяющие объем топки, поверхность стен топки.

Выходные параметры

- расчетная температура газа на выходе топки;
- лучистый поток через выходное окно;
- тепловосприятие поверхностей нагрева в топке.

6.5.3. Методы решения задач оптимального проектирования

Для решения используется метод негладких штрафных функций и модификация r -алгоритма [184, 176] для решения задачи, в ограничения

которой входят только ограничения областей определения используемых функций.

Модификация r -алгоритма заключается в регулировке шага таким образом, чтобы в ходе оптимизации не нарушались области определения функций.

6.5.4. Интерфейс пользователя

Программные средства формирования математической модели оптимизационной задачи и отображения результатов расчетов для анализа реализованы в среде системы программирования DELPHI. Содержат следующие экранные формы:

- общие ограничения модели;
- ограничения участков газохода;
- ограничения трактов (по аэродинамическому или гидравлическому сопротивлению);
- ограничения элементов тепловой схемы;
- ограничения поверхностей нагрева;
- тепловосприятости поверхностей нагрева;
- ограничения регулярных участков топки;
- входы-выходы,
- выполнение оптимизационных расчетов.

На рис. 6.8 приведена экранная форма «Тепловосприятости поверхностей нагрева». В верхнем окне формы отображается список участков газохода и поверхности нагрева (пакеты), размещенные в каждом участке. Для каждого пакета определяется характеристика «Тип модели». Возможные значения:

«Q-модель» – тепловосприятие является независимой переменной, одна из конструктивных характеристик пакета определяется в результате расчета;

«F-модель» – все конструктивные характеристики пакета являются независимыми переменными, тепловосприятие определяется в результате расчета;

«Фиктивная модель» – расчеты пакета не производятся, в тепловом расчете учитывается его фиксированное тепловосприятие (используется при упрощенных расчетах).

Поз.ГХ	№	Имя пакета	Тип пакета	Тип модели
5	119	ПОТ ТОП	Радиационная	Q - модель
5	150	ЭКР ТОП	Радиационная	Q - модель
5	149	Под топки	Радиационная	Q - модель
6	122	ШП	Ширмовая	Q - модель
7	152	ФЕСТОН	Конвективная гладкотруби	Фиктивная модель
9	132	КП2	Конвективная гладкотруби	Q - модель
11	138	КП3	Конвективная гладкотруби	Q - модель
13	144	КП4	Конвективная гладкотруби	Q - модель
15	113	КП1	Конвективная гладкотруби	Q - модель

№	Имя пакета	Характеристика	Значение	Нач.значение	MIN	MAX	Штраф	Невязка	Учет/варьир.
103	ВЭ1	Теплосъем, ккал/кг топ.	617,78	708,1	300	1300	100	0	Баланс
105	ВЭ2	Теплосъем, ккал/кг топ.	652,3	652,3	300	800	0	0	Нет
112	ПОТ ПК	Теплосъем, ккал/кг топ.	9,7426	11,33956	0,1	17,00934	0	0	Нет
113	КП1	Теплосъем, ккал/кг топ.	600	319,4	100	600	1000	0	Баланс
119	ПОТ ТОП	Теплосъем, ккал/кг топ.	101,72	0	50	155,05815	0	0	Нет
122	ШП	Теплосъем, ккал/кг топ.	776,49	845,6	100	1200	0	0	Да
132	КП2	Теплосъем, ккал/кг топ.	500,01	1004,1	500	1538,4725	0	0	Да
138	КП3	Теплосъем, ккал/кг топ.	627,74	534,7	200	700	0	0	Да
144	КП4	Теплосъем, ккал/кг топ.	499,99	282,2	100	500	0	0	Да
149	Под топки	Теплосъем, ккал/кг топ.	62,364	0	0	200	0	0	Нет
150	ЭКР ТОП	Теплосъем, ккал/кг топ.	3348,4	3303,652	0	4955,478	0	0	Нет

Рис. 6.8. Тепловосприятия поверхностей нагрева

В нижнем окне формы для тепловосприятия каждой поверхности нагрева отображаются следующие характеристики:

Значение – значение, полученное в результате расчета;

Нач.значение – начальное значение, задаваемое пользователем;

MIN – минимальное значение, которое может принимать переменная или расчетная характеристика математической модели (задается пользователем);

МАХ – максимальное значение, которое может принимать переменная или расчетная характеристика математической модели (задается пользователем);

Штраф – значение коэффициента ограничения (для расчетных характеристик) в негладкой штрафной функции (задается пользователем);

Невязка – невязка ограничения, полученная в результате расчетов;

Учет/варьирование – задается пользователем, возможные значения:

- «Да» – ограничение учитывается для расчетных характеристик, переменная варьируется для переменных модели;
- «Нет» – ограничение не учитывается для расчетных характеристик, переменная не варьируется для переменных модели.
- «Баланс» – тепловосприятие поверхности определяется из условий теплового баланса тепловой схемы котла (это необходимо, например, когда зафиксированы температуры теплоносителя на входе и выходе пароводяного тракта).

В нижнем окне формы «Тепловосприятия поверхностей нагрева» отображается список поверхностей, для которых поле «Тип модели» не равно значению «Фиктивная модель».

На рис. 6.9 приведена экранная форма «Ограничения поверхностей нагрева».

Ограничения поверхностей нагрева				
Поверхности нагрева				
Поз.ГХ	№	Имя пакета	Тип пакета	Тип модели
5	119	ПОТ ТОП	Радиационная	Q - модель
5	150	ЭКР ТОП	Радиационная	Q - модель
5	149	Под толки	Радиационная	Q - модель
6	122	ШП	Ширмовая	Q - модель
7	152	ФЕСТОН	Конвективная гладкотруби	Фиктивная модель
9	132	КП2	Конвективная гладкотруби	Q - модель
11	138	КП3	Конвективная гладкотруби	Q - модель
13	144	КП4	Конвективная гладкотруби	Q - модель
15	113	КП1	Конвективная гладкотруби	Q - модель

Ограничения выделенной поверхности нагрева							
Характеристика выделенного пакета	Значение	Нач.значение	MIN	MAX	Штраф	Невязка	Учет/варьирование
Толщина стенки трубы, мм	5	5	2,5	7,5	0	0	Нет
Поперечный шаг, мм	80	80	80	120	0	0	Нет
Заходность	3	3	2	4	0	0	Нет
Смещение от начала уч.газохода, м	1,4689	0	0	999999	0	0	Нет
Число петель	2,3383	2	2	8	50	0	Нет
Доля газохода	1	1	0,5	1,5	0	0	Нет
Темп.(наружная) стенки в т.1, град	533,62	999	0	585	700	0	Нет
Наружный диаметр трубы, мм	32	32	16	48	0	0	Нет
Темп.(наружная) стенки в т.2, град	569,87	999	0	585	700	0	Нет
Козф. запаса прочности в т.1	1,6575	999	1	7	100	0	Нет
Козф. запаса прочности в т.2	1,1039	999	0,9243	7	500	0	Нет
Линейная скорость газов м/с	13,84	999	0	21	0	0	Нет
Линейная скорость пара (воздуха) м/с	16,135	999	0	24	0	0	Нет

Рис. 6.9. Ограничения поверхностей нагрева

В экранной форме в верхнем окне отображается список поверхностей. В нижнем окне отображаются характеристики выделенной поверхности нагрева (для тех поверхностей, для которых значение поля «Тип модели» не равно значению «Фиктивная модель»).

Остальные экранные формы организованы аналогичным образом.

Замечание. Для решения рассматриваемой задачи используется метод негладких штрафных функций. Если в результате расчетов невязка некоторых ограничений оказывается ненулевой, необходимо увеличивать соответствующие значения поля «Штраф» – при достаточно больших значениях штрафных коэффициентов решение исходной задачи совпадает с решением задачи безусловной оптимизации с негладкой штрафной целевой функцией.

В связи с тем, что процесс решения оптимизационной задачи может быть достаточно длительным, в систему включены средства графического отображения хода оптимизационного процесса – на каждой итерации отображаются текущие значения целевой функции и максимальной невязки ограничений. Пример экранной формы средств графического отображения приведен на рис. 6.10.

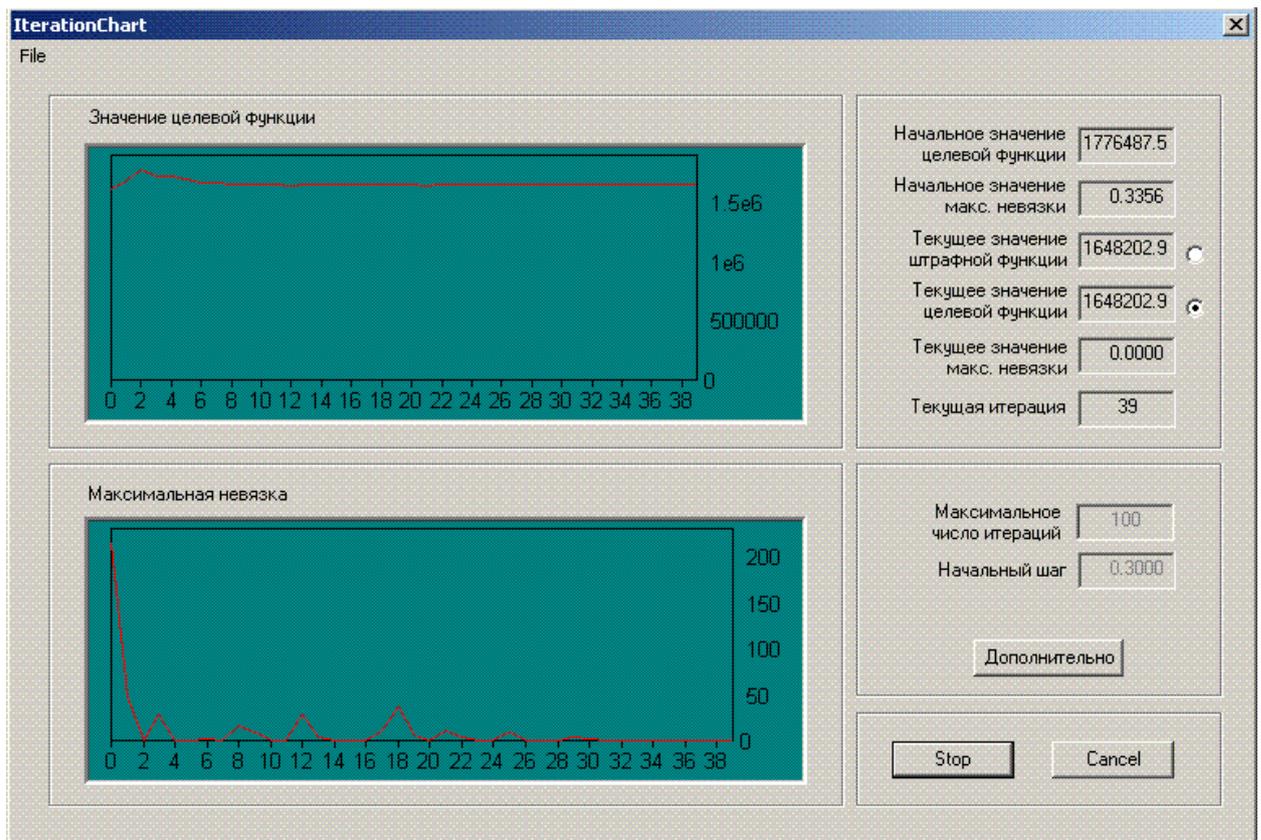


Рис. 6.10. Экранная форма средств графического отображения

Пользователь при необходимости может прервать вычислительный процесс. Среда разработки – Microsoft VISUAL C++;

6.6. Результаты вычислительных экспериментов

В настоящее время разработан прототип оптимизационной подсистемы, реализующий основные функции оптимизации проектируемых

конструкций для котла ТП-80. В качестве базового рассматривался проект, разработанный в Харьковском ЦКБ «ЭНЕРГОПРОГРЕСС».

Для оценки эффекта от применения предлагаемого подхода решались задачи оптимизации стоимости котлоагрегата при всех ограничениях, перечисленных выше. Для аэродинамических и гидравлических сопротивлений трактов предельные величины определялись по базовому варианту. Расчеты производились для номинального режима, топливо – газ.

Решаемые задачи отличались по составу переменных оптимизации. Рассматривались следующие задачи:

базовый вариант – все переменные зафиксированы;

вариант 1 – варьируются тепловосприятости поверхностей нагрева, остальные переменные зафиксированы;

вариант 2 – варьируются:

- тепловосприятости поверхностей нагрева,
- угол потолка первого участков газохода,
- остальные переменные зафиксированы;

вариант 3 – варьируются

- тепловосприятости поверхностей нагрева,
- угол потолка первого участков газохода,
- для ширмы – поперечный шаг, толщина стенки, наружный диаметр трубы,
- для конвективных перегревателей – заходность, поперечный шаг, толщина стенки, наружный диаметр трубы.
- остальные переменные зафиксированы.

Результаты вычислительных экспериментов приведены в табл. 6.1.

Таблица 6.1

Результаты вычислительных экспериментов

Задача	Стоимость, \$	Сокращение стоимости, %	Материалоемкость, т
Базовый вариант	1745600	0	335.28
Вариант 1	1490100	14.6	293.99
Вариант 2	1452400	16.8	287.16
Вариант 3	1085100	37.8	214.4

Необходимо отметить, что конструкции, полученные в результате решения оптимизационных задач, должны уточняться специалистами в прикладной области, поскольку используемые математические модели являются в значительной степени приближенными и не учитывают явно всех особенностей реальных проблем. Так основной эффект для варианта 3 достигнут за счет уменьшения диаметра и толщины стенки труб. Это приводит к уменьшению жесткости конструкций, что требует специальных дополнительных работ по обеспечению требуемой жесткости. Другая проблема связана с отложениями солей на стенках труб в ходе эксплуатации котлов. При малых диаметрах труб проходное сечение быстро сужается, что сокращает сроки эксплуатации или также требует дополнительных работ.

Косвенно учесть эти особенности можно, вводя достаточно высокие ограничения снизу на диаметры труб в решаемых оптимизационных задачах.

6.7. Выводы

В разделе для задачи оптимального проектирования энергетических котлоагрегатов с учетом номинального режима эксплуатации разработана математическая модель, которая обеспечивает вычисление всех необходимых характеристик котлоагрегата в соответствии с принятыми нормами расчетов

и обладает структурой, позволяющей выполнить редукцию исходной задачи большой размерности к задаче существенно меньшей размерности. Существенным в такой модели является включение в состав независимых переменных, по которым производится оптимизация, тепловосприятлий поверхностей нагрева.

В результате редукции формируется негладкая невыпуклая задача. Для ее решения используется модификация негладкого метода оптимизации – r -алгоритма Н.З. Шора.

Предложенная схема решения задачи оптимального проектирования оказалась очень эффективной, однако в процессе решения не учитывались ограничения альтернативных режимов эксплуатации. Это приводит к необходимости дальнейших существенных доработок получаемых конструкций.

Для преодоления этих проблем была разработана обобщенная модель, в которой для каждого режима эксплуатации использовалась эффективная модель с учетом одного режима, а связь между разными режимами обеспечивалась с помощью точных штрафных функций.

Для подготовки исходных данных и анализа результатов расчетов был разработан прототип системы автоматизированного проектирования, который передан в опытную эксплуатацию в Харьковское ЦКБ «ЭНЕРГОПРОГРЕСС». Акты передачи в опытную эксплуатацию содержатся в отчетных документах по темам:

ВК.120.12.08 „Розробка прототипу програмних засобів оптимального проектування при реконструкції та модернізації енергетичних котлоагрегатів ТЕС надкритичного тиску”, 2008 г., № госрегистрации 0108U002200;

ВК.120.12.14 ”Розробка програмних засобів оптимального проектування енергетичних котлоагрегатів при реконструкції та модернізації ТЕС з урахуванням альтернативних режимів експлуатації”, 2014 г., № госрегистрации 0114U001862.

По результатам работы можно сделать следующие выводы о том, что разработанные программные средства позволяют существенно

- сократить время выработки технических решений на этапах тендерного и эскизного проектирования;
- снизить стоимость, материалоемкость, эксплуатационные затраты проектируемых конструкций
- повысить качество принимаемых решений по надежности и загрязнению окружающей среды.

Для эффективного использования разработанных программных средств должны быть автоматизированы все этапы процессов проектирования энергетических котлоагрегатов, а программные средства оптимизации проектируемых конструкций должны органично вписываться в общую автоматизированную систему.

ЗАКЛЮЧЕНИЕ

Работы, результаты которых изложены в диссертации, были инициированы необходимостью обобщить и обосновать подходы, приведенные в разделе 6 и позволившие реализовать эффективную схему решения сложной практической задачи оптимального проектирования энергетических котлоагрегатов.

Разработанная математическая модель задачи состояла из совокупности взаимосвязанных расчетных модулей. Выходы одних модулей являлись входами других. Такая структура задачи позволяла выражать одни величины через другие и выполнить редукцию к задаче существенно меньшей размерности по сравнению с исходной.

Существенной проблемой была ограниченность областей определения функций расчетных модулей. Естественным решением было продолжение функций с допустимой области на все пространство переменных. В результате формировалась негладкая невыпуклая задача, которая решалась r -алгоритмом Н.З. Шора.

Для обоснования предложенных подходов требовалось разработать такие процедуры продолжения функций, которые сохраняли бы выпуклость построенной задачи, если исходная задача с ограничениями была выпуклой. Этим вопросам посвящен раздел 2. Разработаны новые эффективные методы построения задач безусловной оптимизации эквивалентных исходным задачам с ограничениями. Формируемая выпуклая задача безусловной оптимизации зависит от одного числового параметра, значение которого определяется по ходу работы оптимизационного алгоритма. Реализованные программные средства продемонстрировали существенные преимущества при решении плохо обусловленных специальных задач по сравнению с существующими оптимизационными пакетами.

Разработанные методы продолжения функций обладали полезным свойством – для определения значения параметра, от которого зависит формируемая задача безусловной оптимизации, не требовалось решать трудоемкие вспомогательные задачи. Это кардинально отличалось от ситуации, характерной для точных штрафных функций. Для оценки значений штрафных коэффициентов не были известны алгоритмы, не требующие решений вспомогательных задач. Штрафные коэффициенты либо определялись пользователем при решении практических задач, что часто приводило к использованию завышенных значений коэффициентов, либо для их оценки решались вспомогательные задачи.

Раздел 3 посвящен описанию новых подходов к использованию точных штрафных функций, не требующих решения сложных вспомогательных задач. Схема решения исходной задачи с ограничениями заключается в следующем. Формируется штрафная функция с некоторыми начальными штрафными коэффициентами и запускается алгоритм безусловной оптимизации штрафной функции. Штрафные коэффициенты уточняются по ходу работы оптимизационного алгоритма. На каждой итерации алгоритма определяется текущая точка x_k , некоторым образом вспомогательная допустимая точка y_k исходной задачи (может использоваться на всех итерациях одна и та же начальная допустимая точка) и выполняются оценки штрафных коэффициентов. Предлагается три способа оценки.

Первый способ оценки основан на вычислении производных по направлению p , $p = \frac{y_k - x_k}{\|y_k - x_k\|}$, целевой функции и функции ограничений исходной задачи в точке \bar{y}_k , лежащей на пересечении отрезка $[y_k, x_k]$ с границей допустимого множества исходной задачи. Приводятся соотношения для вычисленных величин, при нарушении которых штрафные

коэффициенты должны увеличиваться. Формулируются условия конечности штрафных коэффициентов.

Второй способ оценки основан на анализе поведения функций исходной задачи в δ -окрестности точки x_k на отрезке $[y_k, x_k]$. Этот подход использовался для построения алгоритмов решения негладких невыпуклых задач. Основной результат – сходимость к стационарной допустимой точке.

Третий способ оценки основан на анализе значений функций исходной задачи в точках x_k, y_k .

Величина штрафных коэффициентов, которая устанавливается по ходу работы оптимизационного алгоритма, существенно зависит от правила P , определяющего допустимую точку y_k для текущей точки x_k (генерируемой оптимизационным алгоритмом). Выбор слишком больших значений штрафных коэффициентов приводит к ошибкам округления, неустойчивости работы алгоритма.

Для третьего способа оценки штрафного коэффициента формулируется задача выбора оптимального правила P . Эта задача по трудоемкости эквивалентна исходной, но для практического применения может использоваться приближенное решение. В разделе приводится оценка величины штрафных коэффициентов, которая может устанавливаться при использовании специального правила P для задач линейного программирования. Эта оценка совпадает с известными классическими результатами.

Предложенные подходы наиболее эффективны при использовании алгоритмов негладкой оптимизации, r -алгоритма Н.З. Шора для решения задач оптимизации с ограничениями.

Особое значение предложенные подходы имеют в методах декомпозиции блочных задач оптимизации со связывающими переменными. В таких методах возникают существенные проблемы в ситуациях, когда

подзадачи не имеют решений при некоторых значениях связывающих переменных. Эти проблемы могут быть решены при использовании точных штрафных функций. Однако вспомогательные упрощенные задачи, необходимые для оценки значений штрафных коэффициентов традиционными методами, оказываются слишком трудоемкими. Предложенные подходы позволяют естественным образом преодолеть эти проблемы

Раздел 4 посвящен вопросам применения схем декомпозиции по переменным для решения структурированных нелинейных задач выпуклой оптимизации. Рассматриваются следующие проблемы:

точные решения подзадач для отдельных блоков получены быть не могут, необходимо использовать приближенные решения;

подзадачи могут не иметь допустимых решений при некоторых значениях связывающих переменных; выявление таких ситуаций может быть очень непростой проблемой; существующие алгоритмы могут не сходиться к оптимальному решению при возникновении таких ситуаций; эти проблемы требуют разработки специальных приемов регуляризации исходной задачи;

для точных штрафных функций, которые могут использоваться при регуляризации исходной задачи, невозможно определить значения штрафных коэффициентов существующими методами – трудоемкость вспомогательных задач сравнима с трудоемкостью исходной задачи.

Также рассматриваются вопросы построения эффективных алгоритмов декомпозиции и реализации соответствующих программных средств.

В разделе сформулированы условия, при выполнении которых может быть вычислен ε -субградиент оптимального значения подзадачи при заданных значениях связывающих переменных. Для таких вычислений должна быть задана допустимая точка подзадачи (приближенное решение) и выполняться определенные соотношения для ε -субградиентов функций подзадач в заданной точке. Описаны процедуры вычисления ε -субградиента

оптимального значения подзадачи на основе кусочно-линейной аппроксимации, полученной в процессе решения этой подзадачи.

Рассматриваются различные приемы регуляризации исходных задач, обеспечивающие существование допустимых решений подзадач при любых значениях связывающих переменных. Все эти приемы основаны на использовании точных штрафных функций.

Подраздел 4.3 посвящен вопросам определения значений штрафных коэффициентов при использовании точных штрафных функций в схемах декомпозиции. Детализируются подходы, изложенные в разделе 3, применительно к схемам декомпозиции. Для задач с ограниченными областями определения функций в подразделе 4.4 предлагается использовать методы выпуклого продолжения, описанные в разделе 2. Функции подзадач должны быть липшицевыми.

Подраздел 4.6 посвящен естественному обобщению блочных задач со связывающими переменными – задачам линейного программирования с вложенной структурой. Эти задачи возникают в таких прикладных областях, как моделирование в экономике, многоэтапное стохастическое программирование и др. В подразделе описываются схема декомпозиции для задач такого вида, особенности программной реализации и результаты вычислительных экспериментов для задач планирования инвестиций в электроэнергетике.

В разделе 5 рассматриваются задачи построения классификаторов, относящих каждую точку пространства признаков к одному из m классов, $m \geq 2$, условия линейной разделимости множеств.

Для линейно неразделимых множеств предложена новая модель минимизации эмпирического риска. Сформулированная задача является частично целочисленной. Для вычисления оценки оптимального решения и построения приближенных решений предлагается использовать непрерывную релаксацию такой задачи. Проводится сравнение с известными

математическими моделями. В случае, когда необходимо получать решения близкие к оптимальным, предложенная модель может использоваться для построения алгоритмов направленного поиска (методов ветвей и границ).

Также для линейно неразделимых множеств рассмотрена задача построения полосы минимальной ширины, вне которой точки обучающей выборки классифицируются правильно. Рассмотренная задача невыпукла. В общем случае для этой задачи неизвестны эффективные методы поиска глобального минимума. Но задача оказывается полиномиально разрешимой при использовании специальной нормы в пространстве признаков. Полученный результат может использоваться для формирования приближенных решений в общем случае.

Наряду с линейными классификаторами в случае многих классов в разделе рассмотрены последовательные линейные классификаторы, основанные на формировании линейных классификаторов для каждой пары множеств из обучающей выборки. Показано, что возможности последовательных линейных классификаторов шире по сравнению с линейными классификаторами. Таким образом, в случае многих классов целесообразно использовать последовательные линейные классификаторы.

Важным прикладными задачами являются задачи выделения информативных признаков. Разработаны алгоритмы программные средства решения таких задач, которые использовались при выделении наиболее информативных признаков в задачах диагностики на основании экспрессии генов человека (совместно с Институтом генетики НАН Украины). Исходная размерность признакового пространства – 25000 генов. В результате применения предложенного подхода количество информативных признаков удавалось сокращать до 20÷50.

Приведены результаты вычислительных экспериментов, показывающие эффективность использования методов негладкой

оптимизации в задачах построения классификаторов с большим числом точек в обучающих множествах.

В разделе 6 рассматривались прикладные задачи оптимального проектирования в электроэнергетике, актуальность которых определяется потребностью в реконструкции и модернизации энергоблоков теплоэлектростанций, выработавших свой ресурс. Энергоблоки являются сложными техническими объектами, состоящими из большого числа связанных между собой отдельных компонент. Для оптимизации конструктивных характеристик таких объектов были разработаны математические модели, позволяющие выполнять редукцию исходных задач большой размерности к задачам существенно меньшей размерности. В результате редукции формируется негладкая невыпуклая задача. Для ее решения используется модификация негладкого метода оптимизации – r -алгоритма Н.З. Шора.

Разработанные программные средства переданы в опытную эксплуатацию в Харьковское ЦКБ «ЭНЕРГОПРОГРЕСС» и позволяют существенно

сократить время выработки технических решений на этапах тендерного и эскизного проектирования;

снизить стоимость, материалоемкость, эксплуатационные затраты проектируемых конструкций

повысить качество принимаемых решений.

Для эффективного использования программные средства оптимизации проектируемых конструкций должны органично вписываться в общую систему автоматизированного проектирования, принятую в проектной организации.

Приложение А. Расчет конвективных и ширмовых поверхностей нагрева

Определение расчетного тепловосприятия Q_m^p ширмовых и конвективных пакетов (m – индекс пакета) производится в соответствии с нормами теплового расчета [171]. Для описания структуры оптимизационной задачи важно отметить, что Q_m^p есть функция от характеристик теплоносителей на входе пакета m : D – расход, t' – температура, p' – давление пара, V – расход ($V = V_g + V_r$), θ' – температура дымового газа, J'_l – поток лучистого тепла из топки; характеристик теплоносителей на выходе: t'' , p'' , θ'' ; от конструктивных характеристик: d – внешний диаметр труб, δ – толщина стенок труб, s_1 – поперечный шаг, s_2 – продольный шаг, z – заходность, n – число петель, A – ширины, B – средней высоты (глубины) сечения газохода

$$Q_m^p = Q_m^p(D, t', p', V, \theta', J'_l, t'', p'', \theta'', d, \delta, s_1, s_2, z, n, A, B).$$

Расчетное лучистое тепло из топки $Q_m^{l.p}$, воспринимаемое пакетом m , также вычисляется в соответствии с нормами теплового расчета [171], и является функцией потока лучистого тепла из топки на входе J'_l и конструктивных характеристик

$$Q_m^{l.p} = Q_m^{l.p}(J'_l, d, s_1, s_2, z, n, A, B).$$

Уравнения конвективного теплообмена и поглощения лучистого тепла из топки являются нелинейными и сводятся к условию того, что расчетное тепловосприятие равно балансовому для каждого пакета m

$$Q_m^{бал} - (Q_m^p(D, t', p', V, \theta', J'_l, t'', p'', \theta'', d, \delta, s_1, s_2, z, n, A, B) + Q_m^{l.p}(J'_l, d, \delta, s_1, s_2, z, n, A, B)) = 0, \quad m \in M, \quad (A.1)$$

$$Q_m^{бал} = Q_m^{l.p}(J'_l, d, s_1, s_2, z, n, A, B), \quad m \in M_l. \quad (A.2)$$

где M – множество всех основных пакетов, $M_{л}$ – множество основных пакетов, получающих лучистое тепло из топки.

Расчет температуры стенок труб пакетов выполняется:

для прямотока в двух точках: первая – в конце первого ряда труб по ходу газа, вторая – в конце первого ряда последней полупетли;

для противотока в одной точке – в конце первого ряда последней полупетли.

Расчет коэффициентов запаса прочности выполняется в тех же точках в соответствии с нормами теплового расчета [171].

Расчеты аэродинамического сопротивления для газового и воздушного трактов, гидравлического сопротивления пароводяных трактов, скоростей теплоносителей в поверхностях нагрева осуществляются в соответствии с нормами теплового расчета [171].

Расчет металлоемкости, стоимости, приведенных затрат производится в соответствии с [92].

Поверхности нагрева, входящие в состав котлоагрегата ТП-80, описываются следующими классами, порожденными от базового класса CPacketGH:

CShirma – гладкотрубный ширмовый перегреватель;

CConvectPacket – гладкотрубные конвективные перегреватели, экономайзеры;

CTVP_Packet – трубчатые воздухоподогреватели;

CWallPacket – настенная поверхность (в газоповоротной камере).

Базовый класс CPacketGH определяет общий интерфейс. Реализация функций в каждом дочернем классе выполнена в соответствии нормами теплового расчета для соответствующего типа поверхностей нагрева.

Расчет температуры стенок труб поверхностей нагрева производится с некоторыми упрощениями по сравнению с нормами теплового расчета [171].

Обозначим i', i'' энтальпии пара на входе и выходе из поверхности нагрева, i – уточненная средняя энтальпия пара в расчетном сечении, $\Delta i = i'' - i'$.

Определим величины

$$i = i' + \zeta * \Delta i * \eta_{uu};$$

$$i_{\max} = i + (\eta_r \eta_k / \rho_e - 1) * \zeta * \Delta i * \eta_{uu};$$

где

ζ – уточняющий коэффициент для расчета температуры стенки,

η_{uu} – коэффициент неравномерности тепловосприятия по ширине котла,

η_r – коэффициент неравномерности тепловосприятия развернутой трубы, отнесенный к среднему тепловосприятию поверхности нагрева,

η_k – коэффициент конструктивной нетождественности,

ρ_e – коэффициент гидравлической развертки.

По величинам i, i_{\max} определяются уточненная средняя температура t и температура t_{\max} пара в расчетном сечении развернутой трубы.

Обозначим q_{\max} – максимальное удельное тепловосприятие в расчетном сечении развернутой трубы,

$$q_{\max} = \eta_{uu} \eta_r (\theta_p - t) \left(\beta \mu \left(\frac{2\delta}{\lambda_m (1 + \beta)} \right) + 1 / \alpha_2 \right) + 1 / \alpha_1 + 0.25 \varepsilon,$$

где

$$\alpha_1 = k_{mp} \alpha_k + \alpha_l,$$

k_{mp} – коэффициент неравномерности тепловосприятия по окружности трубы;

θ_p – температура газов в рассчитываемом ряду:

$$\theta_p = \begin{cases} \theta^1 + \frac{H_1}{H} (\theta^1 - \theta^0), & \text{при прямотоке,} \\ \theta^0, & \text{при противотоке,} \end{cases}$$

здесь θ^0, θ^1 – температуры газов на входе и на выходе поверхности нагрева,
 H – величина поверхности нагрева, H_1 – величина поверхности одного ряда;

α_k – коэффициент теплоотдачи конвекцией;

α_l – коэффициент теплоотдачи излучением, α_k и α_l вычисляются при
 температуре θ_p и температуре загрязненной стенки $t_3 = \theta_p + \Delta$;

β – отношение наружного диаметра трубы к внутреннему;

μ – коэффициент растечки тепла, $\mu = 1$ (константа);

δ – толщина стенки трубы;

λ_m – коэффициент теплопроводности металла трубы при температуре
 $t_{\max} + 50$;

α_2 – коэффициент теплоотдачи от стенки к обогреваемой среде при
 температуре t_{\max}

ε – коэффициент загрязнения.

Средняя по толщине температура t_{cm} металла стенки развернутой
 трубы и температура $t_{cm.n.}$ наружной поверхности стенки развернутой трубы
 определяются следующими соотношениями:

$$t_{cm} = t_{\max} + \beta \mu q_{\max} (\delta / (\lambda_m (1 + \beta)) + 1 / \alpha_2);$$

$$t_{cm.n.} = t_{\max} + \beta \mu q_{\max} (2\delta / (\lambda_m (1 + \beta)) + 1 / \alpha_2);$$

В ходе **расчета на прочность** проверяется выполнение следующего
 условия:

$$(\delta - c)(2\sigma_{don}(t_{cm.}) + \rho) / (\rho d) > 1,$$

где

δ – толщина стенок труб ступени, м,

c – суммарная прибавка к расчетной толщине стенки, м, $c=0.001$,

$\sigma_{don}(t_{cm.})$ – номинальное допустимое напряжение, МПа, σ_{don} – функция
 от марки стали и температуры стенки,

ρ – давление среды в расчетном сечении, МПа,

d – наружный диаметр труб ступени, м.

Значение левой части неравенства называется коэффициентом запаса прочности.

Приложение Б. Описание уравнений материального и теплового баланса

Состав типов элементов и способы задания тепловой схемы приняты такими же, как и в программе ТРАКТ, широко используемой в проектных организациях отрасли. Каждый тип элементов в оптимизационной подсистеме реализуется отдельным классом языка C++. Соответствие между типами элементов и классами языка C++ приведено в табл. Б.1. Поскольку в котле ТП-80 задействованы не все возможные типы элементов, некоторым типам в табл. Б.1 не поставлены в соответствие какие-либо классы элементов.

Для описания уравнений теплового и материального баланса введем обозначения:

V'_e, V''_e, V'_g, V''_g – объемы воздуха и газов на входе и выходе из элемента тепловой схемы, $\text{нм}^3/\text{кг}$ ($\text{нм}^3/\text{нм}^3$),

α – избыток воздуха на выходе из топки,

$\Delta\alpha_m$ – присос в топке,

$\Delta\alpha_{nl}$ – присос в пылесистеме,

V_e^0 – теоретический объем сухого воздуха,

V_g^0 – теоретический объем газов при избытке воздуха α ,

j^0 – удельная энтальпия (1 м^3) холодного воздуха,

$J^0 = j^0 V^0$,

φ – коэффициент сохранения тепла,

J', J'' – удельные энтальпии газов (воздуха) на входе и выходе из элемента тепловой схемы, $\text{ккал}/\text{кг}$ ($\text{ккал}/\text{нм}^3$),

Q_H^P – теплота сгорания 1 кг топлива,

$i_{мл}$ – теплота, поступающая в топку в результате подогрева топлива (в расчете на 1 кг),

Q – тепловосприятие элемента тепловой схемы (поверхности нагрева),

γ – доля тепла дополнительных поверхностей (от тепла основных поверхностей),

Q_l – лучистое тепло, воспринимаемое в позиции газохода,

J'_l – лучистый поток на входе в позицию газохода,

J''_l – лучистый поток на выходе из позиции газохода (топки),

d', d'' – расходы пара (воды) на входе и выходе из элемента тепловой схемы, кг/ч,

b_p – расход топлива, кг/ч,

$d^+, i^+, V_g^+, V_{\Gamma}^+, J^+$ ($d^-, i^-, V_g^-, V_{\Gamma}^-, J^-$) – расходы и энтальпии теплоносителя при впрыске (отборе),

u – доля отбора,

i'', i' – удельные энтальпии пара (воды) на входе и выходе из элемента тепловой схемы, ккал/кг.

Описание уравнений материального и теплового баланса приведено в табл. Б.2. В скобках для пароводяных трактов приведены уравнения относительно переменных $I = di / b_p$.

Таблица Б.1.

Соответствие типов элементов (база данных - мат. модель)

Типы элементов (база данных)	Код типа	Типы элементов мат. модели (классы)
Топка	2	CElemTopka
Позиция газохода	3	CElemPozGH
Золоуловитель	6	
Пакет	12	CElemQChange
		CElemPaketAdd
Калорифер	19	CElemHeater
Конденсатор впрысков	17	CElemCondensator
Дроссель	18	
Трубопровод	14	CElemQChange
Сепаратор	15	
Барабан	16	CElemBaraban
Теплообменник	13	
Вход	21	CElem_Input
Впрыск собственного конденсата	26	CElemInject
Отбор теплоносителя	22	CElemReject
Присадка теплоносителя	28	CElemInject
Присадка-отбор тепла (фиксированное тепловосприятие)	23	CElemQChange
Выход	29	CElem_Output
Пустой	24	CElemQChange
Конец рукава	25	CElemEndBranch
Присос	8	CElemAddAir
Газопаровой теплообменник	11	

Таблица Б.2.

Уравнения материального и теплового баланса

Типы (классы) элементов мат. модели	Уравнения материального баланса	Уравнения теплового баланса	Примечания
1	2	3	4
СелемТопка	$V'_g = (\alpha - \Delta\alpha_m - \Delta\alpha_{nl})V^0$ (что определяет объем воздуха на входе воздушного тракта), $V''_g = 0,$ $V''_r = V'_r + V_r^0,$	$J'' - J' = [Q_H^p + i_{ml} + Q_{e.vh} + Q_f - Q_k]^*$ $*(100 - q_3 - q_4 - q_6)/100 + (\Delta\alpha_m + \Delta\alpha_{nl})J^0$ $-(\sum_i Q_i + Q_l)/\varphi,$ $J''_l = Q_l,$ $J'_l = 0$	Здесь $\sum_i Q_i$ – тепловосприятие экранов топки, Q_l – лучистый поток через выходное окно топки Q_f – тепло “форсуночного” пара Q_k – тепло на разложение карбонатов (при сжигании сланцев) $Q_{e.vh}$ – тепло, внесенное воздухом при подогреве его вне котла отборным паром (см. прим.)

Продолж. табл. Б.2.

1	2	3	4
CelemPozGH	$V_{\epsilon}'' = V_{\epsilon}',$ $V_{\Gamma}'' = V_{\Gamma}'$	<p>1. $\varphi(J' - J'') = (1 + \gamma) \sum_i (Q_i - Q_l);$</p> $\varphi(J_l' - J_l'') = (1 + \gamma) \sum_i Q_l^i$ <p>2. $\varphi(J' - J'') = (1 + \gamma) \sum_i (Q_i - Q_l^i),$</p> $\varphi J_l' = (1 + \gamma) \sum_i Q_l^i,$ <p>что эквивалентно</p> $\varphi(J' - J'') = (1 + \gamma) \sum_i Q_i - \varphi J_l';$ <p>3. $\varphi(J' - J'') = (1 + \gamma) \sum_i Q_i$</p>	<p>Суммирование Q_i, Q_l^i производится по основным пакетам позиции газохода.</p> <p>Возможны три варианта:</p> $J_l', J_l'' < > 0;$ $J_l' \geq 0, J_l'' = 0;$ $J_l', J_l'' = 0.$
CelemQChange (газ)	$V_{\epsilon}'' = V_{\epsilon}',$ $V_{\Gamma}'' = V_{\Gamma}'$	$\varphi(J' - J'') = Q;$ $J_l' - J_l'' = Q_l$	Элемент с фиксированным тепловосприятием для газового тракта
CelemQChange (воздух)	$V_{\epsilon}'' = V_{\epsilon}',$ $V_{\Gamma}'' = V_{\Gamma}'$	$J'' - J' = Q$	

Продолж. табл. Б.2.

1	2	3	4
CElemAddAir	$V_g'' = V_g' + \Delta\alpha(1 + 0,016)V^0$ $V_{\Gamma}'' = V_{\Gamma}'$	$J'' = J' + \Delta\alpha(1 + 0,016)J^0;$ $J_{\Gamma}' - J_{\Gamma}'' = 0$	
CElem_Input (воздух)	$V_g'' = V_g^{6x},$ $V_{\Gamma}'' = 0$	$J'' = j^{6x}V_g^{6x}$	
CElemHeater	$V_g'' = V_g',$ $V_{\Gamma}'' = V_{\Gamma}'$	$J'' - J' = Q$	$J_{\Gamma}', J_{\Gamma}'' = 0.$ Не допускается включение между элементами, имеющими лучистый поток
CElem_Output (газ, воздух)	$V_g' = V_g^{6bx},$ $V_{\Gamma}' = V_{\Gamma}^{6bx}$	$J' = j^{6bx}(V_g^{6bx} + V_{\Gamma}^{6bx})$	Для основного выхода возд.тракта – замена на CElemEndBranch
CElemInject (воздух, газ)	$V_g'' = V_g' + V_g^+,$ $V_{\Gamma}'' = V_{\Gamma}' + V_{\Gamma}^+$	$J'' = J' + J^+$ $J_{\Gamma}' - J_{\Gamma}'' = 0$	

Продолж. табл. Б.2.

1	2	3	4
CElemReject (воздух, газ)	$V_g'' = V_g' - V_g^- ,$ $V_g^- = uV_g' ,$ $V_{\Gamma}'' = V_{\Gamma}' - V_{\Gamma}^- ,$ $V_{\Gamma}^- = uV_{\Gamma}'$	$J'' = J' - J^-$ $J^- = uJ'$ $J_n' - J_n'' = 0$	
CElemEndBranch (воздух, газ)	$V_g'' = V_g' ,$ $V_{\Gamma}'' = V_{\Gamma}'$	$J'' - J' = 0$	
CElemBaraban	$d'' = d' - d^- ,$ $d^- = ud'$	$d'i' = d''i^n + d^-i^s ,$ $(I'' = I' - I^- , I'' = d''i^n , I^- = d^-i^s)$	i^n – энтальпия насыщенного пара, i^s – энтальпия воды на линии насыщения
CElemHeater	$d'' = d'$	$I'' - I' = Q$	

Продолж. табл. Б.2.

1	2	3	4
CElemCondensator	$d_+'' = d_+' ,$ $d_-'' = d_-'$	$I_+'' - I_+' = b_p Q ;$ $I_-'' - I_-' = -b_p Q$	$d_+'' , d_+' , I_+'' , I_+' \sim$ – характеристики обогреваемого теплоносителя, $d_-'' , d_-' , I_-'' , I_-' , d_+'' , d_+' , I_+'' , I_+'$ – греющего
CElemQChange	$d'' = d'$	$d'(i'' - i') = b_p Q ,$ $(I'' - I' = b_p Q)$	
CElemPaketAdd	$d'' = d'$	$d'(i'' - i') = b_p \delta (\gamma \sum_k Q_k) ;$ $d'(i'' - i') = b_p \delta (\gamma \sum_k Q_k + Q_l^{энк}) ;$ $(I'' - I' = b_p \delta \gamma \sum_k Q_k) ;$ $((I'' - I') / \delta \gamma = b_p \sum_k Q_k) .$	Если элемент из газоповоротной камеры – случай 2, иначе – 1 (см. прим.); $\gamma \sum_k Q_k$ – суммарная доля доп. поверхностей от суммарного тепла основных пакетов; δ – доля текущей поверхности
CElem_Input	$d'' = d_{ex}$	$i'' = i_{ex} \quad (I'' = d_{ex} i_{ex})$	d_{ex} , i_{ex} – как правило, задаются в исходных данных

Продолж. табл. Б.2.

1	2	3	4
CElemInject	$d'' = d' + d^+$	$d''i'' = d'i' + d^+i^+ \quad (I'' = I' + I^+)$	
CElemReject	$d'' = d' - d^- ,$ $d^- = ud'$	$d''i'' = d'i' - d^-i^- , \quad (I'' = I' - I^-),$ $i^- = i' , \quad (I^- = uI'), \quad d^-i^- = ud'i'$	
CElemEndBranch	$d'' = d'$	$I'' = I'$	
CElem_Output	$d' = d_{\text{блх}}$	$i' = i_{\text{блх}}$ $(I' = d_{\text{блх}}i_{\text{блх}})$	$d_{\text{блх}}, i_{\text{блх}}$ – как правило, задаются в ИСХОДНЫХ ДАННЫХ

Дополнительные особенности формирования системы уравнений материального и теплового баланса:

- избыток воздуха на выходе из топки косвенно определяет объем воздуха на входе воздушного тракта;
- уравнения для барабана косвенно определяют суммарное тепловосприятие предшествующих пакетов, т.е. либо тепловосприятие одного из пакетов должно варьироваться при решении этой системы, либо вход первичного парового тракта должен быть не зафиксированным;
- если зафиксирован выход (температура) первичного тракта, то тепловосприятие одного из пакетов перегревателя должно варьироваться;
- для теплового баланса ГПК вводится дополнительная переменная $Q_a^{знк}$ (в допустимом решении равна нулю) и уравнения принимают вид

$$\varphi(J' - J'') = (1 + \gamma) \sum_i (Q_i - Q_l^i) - Q_a^{знк} ;$$

$$\varphi J'_l = (1 + \gamma) \sum_i Q_l^i + Q_l^{знк} + Q_a^{знк} .$$

- для дополнительных поверхностей уравнение $d'(i'' - i') = I'' - I' = b_p \delta (\gamma \sum_k Q_k + Q_l^{знк})$ приводится к виду

$$Q_z = Q_l^{знк} / \gamma ,$$

$$(I'' - I') / \delta \gamma = b_p (\sum_k Q_k + Q_z) .$$

ЛИТЕРАТУРА

1. *Astorino A., Gaudioso M.* Polyhedral separability through successive LP // Journal of Optimization Theory and Applications. – 2002. – Vol. 112. – N 2. – P. 265–293.
2. *Astorino A., Fuduli A.* Support vector machine polyhedral separability in semisupervised learning // Journal of Optimization Theory and Applications. – 2015. – T. 164. – №. 3. – C. 1039-1050.
3. *Bagirov A.M., Ugon J., Webb D., Karasözen B.* Classification through incremental max–min separability // Pattern Analysis and Applications. – 2011. – Vol. 14, Issue 2. – P 165–174.
4. *Bagirov A.M.* Max–min separability // Optimization Methods and Software. – 2005. – Vol.20. Issue 2-3. – P. 277–296.
5. *Bagirov A. M. et al.* Piecewise Linear Classifiers Based on Nonsmooth Optimization Approaches // Optimization in Science and Engineering. – Springer New York, 2014. – C. 1-32.
6. *Benders J.F.* Partitioning procedures for solving mixed variables programming problems // Numerische Mathematik. – 1962. –N 4. – P. 238–252.
7. *Ben-Tal A., Ghaoui L.E., Nemirovski A.* Robust Optimization. – Princeton University Press, 2009. – 542 p.
8. *Ben-Tal A., Nemirovski A.* Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications – SIAM-MPS Series in Optimizatон. SIAM Publications, Philadelphia, 2001. – 558 p.
9. *Bennett K.P., Mangasarian O.L.* Robust Linear Programming Discrimination of Two Linearly Inseparable Sets // Optimization Methods and Software. – 1996. – № 5. – P. 23–34.

10. *Bennett K. P., Mangasarian O. L.* Serial and parallel multicategory discrimination // *SIAM Journal on Optimization*. – 1994. – Vol. 4. – N 4. – P. 722–734.
11. *Birge J.R., Donohue Ch.J., Holmes D.F., Svintsitski O.G.* A parallel implementation of the nested decomposition algorithm for multistage stochastic linear programs // *Mathematical Programming*. – 1996. – Vol. 75. – P. 327–352.
12. *Bottou L., Cortes C., Denker J., Drucker H., Guyon I., Jacke L. l, LeCun Y., Muller U., Sackinger E., Simard P., Vapnik V.* “Comparison of classifier methods: A case study in handwriting digit recognition,” in *Proc. Int. Conf. Pattern Recognition*. 1994, pp. 77–87.
13. *Bredensteiner E.J., Bennett K.P.* Multicategory Classification by Support Vector Machines // *Computational Optimization and Applications*. – 1999. – Vol. 12. – Issue 1-3. – P. 53–79
14. *Brooke A., Kendrick D., Meeraus A.* GAMS – A User’s Guide. - Redwood City: The Scientific Press , 1988. – 278 p.
15. *Byrd R. H., Nocedal J., Waltz R.A.* “[KNITRO: An integrated package for nonlinear optimization](#)”, In G. di Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, Springer. – 2006. – P. 35–59.
16. *Byrd R.H., Nocedal J., Waltz R.* Steering exact penalty methods // *Optim. Methods Softw.* – 2008. – Vol. 23. – N 2. – P. 197–213.
17. *Byrd R.H., Lopez-Calva G., Nocedal J.* A line search exact penalty method using steering rules // *Math. Program., Series A and B*. – 2012. – V. 133. – P. 39–73.
18. *Dantzig G.B., Wolfe P.* The decomposition algorithm for linear programming // *Econometrica*. – 1961. – Vol. 29. – № 4. – P. 767–778.
19. *Geoffrion A.M.* Primal resource-directive approaches for optimizing nonlinear decomposable systems // *Operations Research*. – 1970. –Vol. 18. – P. 375–403.

20. *Chang C.C., Lin C.J.* LIBSVM: A Library for Support Vector Machines // Journal ACM Transactions on Intelligent Systems and Technology. – 2011. – Vol. 2. – Issue 3. – Article No. 27.
21. *Gondzio J., Vial J.-P.* Warm Start and ε -Subgradients in a Cutting Plane Scheme for Block-Angular Linear Programs // Computational Optimization and Applications. – 1999. – Vol. 14. – P. 17–36.
22. *Crammer K., Singer Y.* On the learnability and design of output codes for multiclass problems // Comput. Learning Theory. – 2000. – P. 35–46.
23. *Demyanov V.F.* Mathematical diagnostics via nonsmooth analysis // Optimization Methods and Software. – 2005. – Vol. 20. – N 2–3. – P. 197–218.
24. *Dogantekin E., Dogantekin A., Avci D.* Automatic Hepatitis Diagnosis System based on Linear Discriminant Analysis and Adaptive Network Based Fuzzy Inference System // Expert Systems with Applications. – 2009. – Vol. 36. – Issue 3. – P. 11282–11286.
25. *Drud A.* CONOPT documentation // ARKI consulting and development A/S. Bagsvaerd, Denmark. – 2004.
26. *Drug A.S.*, Interactions between nonlinear programming and modeling systems // Math. Progr. – 1997. – Vol. 79. – P. 99–123.
27. *Fabian C.* Bundle-Type Methods for Inexact Data // Central European Journal of Operations Research 8 (special issue, T. Csendes and T. Rapcsak, eds.). – 2000. – P. 35–55.
28. *Fabian C., Szoke Z.* Solving two-stage stochastic programming problems with level decomposition // Computational Management Science. – 2007. – Vol. 4. – Issue 4. – P. 313–353
29. *Fletcher R., Leyffer S.* Nonlinear programming without a penalty function // Math. Program. – 2002. – Ser. A 91. – P. 239–269.

30. *Flippo O.E., Rinnoy Kan A.H.G.* Decomposition in general mathematical programming // *Math. Prog.* – 1993. – Vol. 60. – P. 361–382.
31. *Fourer R., Gay D.M., Kernighan B. W.* *AMPL – a Modeling Language for Mathematical Programming*, Second Edition. – Brooks/Cole – Thomson Learning, 2003 – 517 p.
32. *Gassmann H.I.* MSLiP: A computer code for the multistage stochastic linear programming problem // *Math. Prog.* – 1990. – Vol. 47. – P. 407–423.
33. *Geoffrion, A.M.* Generalized Benders decomposition // *Journal of Optimization Theory and Applications.* – 1972. – Vol. 10. – P. 237–260.
34. *Gill P., Wong E.* *User's Guide for SNOPT Version 7.4: Software for Large-Scale Nonlinear Programming.* – Stanford. – 2015. <http://www.sbsi-sol-optimize.com/manuals/SNOPT%20Manual.pdf>.
35. *Grothey A., Leyffer S., Mckinnon K.I.M.* A note on feasibility in Benders Decomposition // *Numerical Analysis Report NA/188*, Department of Mathematics, University of Dundee. – 2000.
36. *Hare W. and Sagastizábal C.* Computing Proximal Points of Nonconvex Functions // *Math. Prog.* – 2009. – Vol. 116(1–2). – P. 221–258.
37. *Hiriart-Urruty J.-B., Lemaréchal C.* *Convex Analysis and Minimization Algorithms. I and II.* – Berlin: Springer Verlag, 1991. – 417 and 347 p.
38. *Hsu C.-W., Lin C.-J.* A comparison of methods for multiclass support vector machines // *IEEE Transactions on Neural Networks.* – 2002. – Vol. 13. – Issue: 2. – P. 415–425.
39. *Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT.* – 2015. – <https://projects.coin-or.org/Ipopt/browser/stable/3.12/Ipopt/doc/documentation.pdf?format=raw>.
40. *Kall P., Wallace S.W.* *Stochastic Programming.* – New York: John Wiley & Sons Ltd. 1994. – 227 p.

41. *Kelley J.* The cutting plane method for solving convex programs // SIAM J. – 1960. – Vol. 8. – N 4. – P. 703–712.
42. *Kiwiel K.C.* Methods of Descent for Nondifferentiable Optimization. – Lecture Notes in Mathematics. – Vol. 1133. – 1985. – 361 p.
43. *Kiwiel K. C.* A proximal bundle method with approximate subgradient linearizations // SIAM Journal on Optimization. – 2006. – Vol. 16. – No. 4. – P. 1007–1023.
44. *Knerr S., Personnaz L., Dreyfus G.* Single-layer learning revisited: A stepwise procedure for building and training a neural network // Neurocomputing: Algorithms, Architectures and Applications, J. Fogelman, Ed. New York: Springer-Verlag. – 1990. – Vol. 68. – Series NATO ASI. – P. 41–50.
45. *Krebel U.* Pairwise classification and support vector machines, in Advances in Kernel Methods – Support Vector Learning, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999. – P. 255–268.
46. *Lemarechal C., Mifflin K.* Nonsmooth Optimization. – Oxford: Pergamon Press, 1978, – 180 p.
47. *Lemarechal C., Nemirovskii A., Nesterov Yu.* New Variants of Bundle Methods // Math. Prog. – 1995. – Vol. 69. – P. 111–147.
48. *Lemaréchal C., Sagastizábal C.* Variable metric bundle methods: from conceptual to implementable forms // Math. Prog. – 1997. – Vol. 76. – P. 393–410.
49. *Louveaux F.V. and Smeers Y.* Optimal investments for electricity generation: A stochastic model and a test-problem. // In R.J-B. Wets and Y. Ermoliev, editors, Numerical Techniques for Stochastic Optimization, Springer-Verlag, 1988, – P. 445–453.

50. *Mayer J.* Stochastic Linear Programming Algorithms: a Comparison Based on a Model Management System. – Amsterdam: Overseas Publishers Association, 1998. – 153 p.
51. *Mifflin R., Sagastizábal C.* A VU-algorithm for convex minimization // Math. Program. – 2005. – Vol. 104. – P. 583–608.
52. *Moreau J.* Proximité et dualité dans un espace hilbertien // Bull. Soc. Math. France. – 1965. – Vol. 93. – P. 273–299.
53. *Murtagh B. A., Saunders M. A.* MINOS 5.0 User's Guide. – STANFORD UNIV CA SYSTEMS OPTIMIZATION LAB, 1983. – №. SOL-83-20.
54. *Oliveira E., Sagastizábal C., Scheimberg S.* Inexact Bundle Methods for Two-Stage Stochastic Programming // SIAM J. Optim. – 2011. – 21(2). – P. 517–544.
55. *Oliveira W., Sagastizábal C.* Level bundle methods for oracles with on-demand accuracy // Optimization Methods and Software. – 2014. – Vol. 29(6). – P. 1180–1209.
56. *Platt J. C., Cristianini N., Shawe-Taylor J.* Large margin DAG's for multiclass classification // Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press. – 2000, – Vol. 12. – P. 547–553.
57. *Rockafellar R.T.* Monotone operators and the proximal point algorithm. // SIAM J. on Control and Optimization. – 1976. – Vol. 14. – P. 877–898.
58. *Ruszczynski A.* A regularized decomposition method for minimizing a sum of polyhedral functions // Math. Prog. – 1986. – Vol. 35. – P. 309–333.
59. *Shapiro A., Dentcheva D., Ruszczynski A.* Lectures on Stochastic Programming: Modeling and Theory. MPS/SIAM //Ser. Optim. – 2009.
60. *Zakeri, G., A.B. Philpott, and D.M. Ryan* Inexact Cuts in Benders Decomposition // SIAM J. on Optimization. – 2000.– Vol. 10(3).– P. 643–657.

61. *Zangwill W.* Non-linear programming via penalty function // *Manag. Sci.* – 1967. – Vol. 13. – No 5. – P. 344–358.
62. *Zhuravlev Yu.I.* An Algebraic Approach to Recognition or Classification Problems // *Pattern Recognition and Image Analysis.* – 1998. – Vol. 8. – № 1.– P. 59–100.
63. *Van Slyke R., Wets R.J.-B.* L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming // *SIAM Journal on Applied Mathematics.* – 1969. – Vol. 17. – P. 638–663.
64. *Vapnik V.N.* The nature of statistical learning theory. – Springer-Verlag, New York, 2000.
65. *R. Wunderling.* Paralleler und Objektorientierter Simplex-Algorithmus, ZIB technical report TR 96-09, Berlin 1996 // (<http://www.zib.de/Optimization/Software/Soplex/soplex.php>)
66. *Айда-заде К.Р.* Исследование нелинейных оптимизационных задач сетевой структуры. // *Автоматика и телемеханика.* 1990. – № 2. – С. 3–14.
67. *Ананьев К.И., Демьянова В.В., Демьянов В.Ф., Кокорина А.В., Свистун С.Я., Стегалин И.С.* Оптимизационные методы в задачах диагностики // *Вестн. С.-Петерб. ун-та, Сер. 10; Прикладная математика, информатика, процессы управления.* – 2011. – Вып. 3. – С. 3–12.
68. *Андрющенко А.И., Змачинский А.В., Понятов В.А.* Оптимизация тепловых циклов и процессов ТЭС. – М.: Высшая школа, 1974. – 280 с.
69. *Березовский О.А., Стецюк П.И.* Об одном способе нахождения двойственных квадратичных оценок Шора // *Кибернетика и системный анализ.* – 2008. – № 2. – С. 89–99.
70. *Бертсекас Д.* Условная оптимизация и множители Лагранжа. – М.: Радио и связь, 1987. – 399 с.
71. *Ванник В. Н., Червоненкис А. Я.* Теория распознавания образов: статистические проблемы обучения. – Наука. Гл. ред. физ.-мат. лит., 1974.

72. *Воронцов К.В.* Машинное обучение. // [http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_\(курс_лекций%2C_К.В.Воронцов\)](http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_(курс_лекций%2C_К.В.Воронцов)).
73. *Груздева Т. В.* Задача сферической бинарной отделимости // Известия Иркутского государственного университета. Серия «Математика». – 2012. – Т. 5. – №. 3. – С. 18-31.
74. *Гупал А.М., Сергиенко И.В.* Оптимальные процедуры распознавания. – Киев: Наук.думка, 2008. – 232 с.
75. *Данилин Ю.М.* Линеаризация и штрафные функции // Кибернетика и системный анализ. – 2002. – № 5. – С. 65–79.
76. *Демьянов В. Ф., Васильев Л. В.* Недифференцируемая оптимизация. – М.: Наука, 1981. – 384 с.
77. *Демьянов В.Ф.* Условия экстремума и вариационное исчисление. – М.: Высш. шк., 2005. – 335 с.
78. *Детуанов V.F., Di Pillo G., Facchinei F.* Exact penalization via Dini and Hadamard conditional derivatives // Optim. Methods and Software. – 1998. –V. 9. – P. 19–36.
79. *Демьянов В. Ф., Факкиней Ф.* Задачи двухуровневой оптимизации и штрафные функции // Известия высших учебных заведений. Математика. – 2003. – № 12. – С. 49–61.
80. *Дубовський С.В., Ленчевський Є.А.* Розроблення нової комплексної системи автоматичного управління електричним навантаженням ОЕС України на основі електротермічних споживачів-регуляторів // Проблеми загальної енергетики. – 2012. – Вип. 4 (31). – С. 12–20.
81. *Дубовський С.В., Левчук А.П., Каденський М.Я.* Підвищення маневрених можливостей енергетичної системи шляхом впровадження теплових насосів-регуляторів у складі ТЕЦ // Проблеми загальної енергетики. – 2013. – Вип. 4 (35). – С. 16–23.

82. *Евтушенко Ю.Г.* Методы решения экстремальных задач и их применение в системах оптимизации. – М.: Наука, 1982. – 432 с.
83. *Евтушенко Ю.Г., Жадан В.Г.* Точные вспомогательные функции в задачах оптимизации // ЖВМ и МФ. – 1990. – т.30. – № 1. – С. 43–57.
84. *Еремин И.И.* Метод "штрафов" в выпуклом программировании // Докл. АН СССР. – 1967. – Т. 173. – № 4. – С. 748–751.
85. *Ермольев Ю.М.* Методы стохастического программирования. – М.: Наука, 1976. – 240 с.
86. *Журавлёв Ю. И.* Об алгебраическом подходе к решению задач распознавания или классификации // Проблемы кибернетики. – 1978. – Т. 33. – С. 5–68.
87. *Журавлёв Ю. И., Рязанов В. В., Сенько О. В.* «Распознавание». Математические методы. Программная система. Практические применения. – М.: Фазис, 2006
88. *Журбенко Н.Г., Саимбетов Д.Х.* К численному решению одного класса задач робастного разделения двух множеств // Методы исследования экстремальных задач. – Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины. – 1994. – С. 52–55.
89. *Журбенко Н.Г.* Об одном ε -субградиентном алгоритме минимизации // Теория оптимальных решений. – 2002. – № 1. – С. 111–118.
90. *Журбенко Н.Г., Ненахов Э.И.* Об одном алгоритме ε -субградиентного типа минимизации выпуклой функции // Теорія оптимальних рішень. – 2004 – № 3. – С. 27–33.
91. *Журбенко Н. Г., Лаптин Ю. П.* Об одном методе расчета тепловых схем // Теорія оптимальних рішень. –2006. – № 5. – С. 120–125.

92. *Змачинский А.В., Медведев В.А., Левченко Г.И.* Техничко-економические расчеты и оптимизация поверхностей нагрева из оребренных труб парогенераторов электростанций. – Саратов: Изд-во СГУ, 1983. – 96 с.
93. *Ириарт-Уррути Ж.-Б.* Оптимизация и выпуклый анализ. – К.: Издательская компания «КИТ», 2004. – 370 с.
94. *Киселева Е.М., Шор Н.З.* Непрерывные задачи оптимального разбиения множеств: теория, алгоритмы, приложения. – К.: Наукова думка, 2005. – 564 с.
95. *Киселева Е.М., Коряшкина Л.С.* Непрерывные задачи оптимального разбиения множеств и r -алгоритмы. – К.: Наукова думка, 2015. – 398 с.
96. *Клер А.М., Скрипкин С.К., Деканова Н.П.* Автоматизация построения статических и динамических моделей теплоэнергетических установок // Изв. РАН. Энергетика. – 1996. – № 3. – 78–84.
97. *Клюшин Д. А., Петунин Ю. И.* Доказательная медицина. Применение статистических методов // М.: Диалектика. – 2008.
98. *Кузьменко В.Н., Бойко В.В.* О применении комбинированного метода выпуклого программирования // Теорія оптимальних рішень. – 2003. – № 2. – С. 19–24.
99. *Кулик М.М.* Техніко-економічні аспекти використання споживачів-регуляторів у системах автоматичного регулювання частотою і потужністю // Проблеми загальної енергетики. – 2015. – Вип.1 (40). – С. 20–28
100. *Лаптин Ю.П., Медведев В.А., Волковицкая П.И.* Система математических моделей расчета и оптимизации конструктивных решений энергетических паровых котлов // Теория оптимальных решений. – К.: Ин-т кибернетики им. В.М. Глушкова НАН Украины. – 1994. – С. 17–22

101. *Лаптин Ю.П.* Объектно-ориентированные средства оптимизационного моделирования сложных объектов // Теория оптимальных решений. – К.: Ин-т кибернетики им. В.М. Глушкова НАН Украины. – 2001. – С. 31–37
102. *Лаптин Ю. П., Журбенко Н. Г.* Разработка программных средств оптимизации сложных технических объектов // Теория оптимальных решений. – К.: Ин-т кибернетики им. В.М. Глушкова НАН Украины. – 2002. – С. 3–12.
103. *Лаптин Ю. П., Журбенко Н. Г., Левин М. М., Волковицкая П.И.* Использование средств оптимизации в системе автоматизированного проектирования энергетических котлоагрегатов КРОКУС // Энергетика и электрификация. – 2003. – № 7. – С. 41–51.
104. *Лаптин Ю.П.* ε -субградиенты в методах декомпозиции по переменным для некоторых задач оптимизации // Теория оптимальных решений. – К.: Ин-т кибернетики им. В.М. Глушкова НАН Украины. – 2003. – № 2. – С. 75 – 82.
105. *Лаптин Ю.П.* Декомпозиция по переменным для некоторых задач оптимизации // Кибернетика и системный анализ. – 2004. – № 1. – С. 98–104.
106. *Лаптин Ю. П., Журбенко Н. Г., Кузьменко В. Н.* Решение блочных нелинейных задач оптимизации со связывающими переменными // Теория оптимальных решений. – К.: Ин-т кибернетики им. В.М. Глушкова НАН Украины. – 2004. – № 3. – С. 142–149.
107. *Лаптин Ю. П.* Решение одного класса задач оптимизации с нелинейными ограничениями // Теория оптимальных решений. – К.: Ин-т кибернетики им. В.М. Глушкова НАН Украины. – 2005 – № 4. – С. 134–139.
108. *Лаптин Ю. П. Журбенко Н. Г.* Некоторые вопросы решения блочных нелинейных задач оптимизации со связывающими переменными // Кибернетика и системный анализ. – 2006. – № 2. – С. 47–55.

109. *Лаптин Ю. П.* Некоторые задачи построения описанных эллипсоидов // Теорія оптимальних рішень. – К.: Ін-т кібернетики ім. В.М. Глушкова НАН України. – 2006. – № 5. – С. 67–75.
110. *Лаптин Ю.П., Лиховид А.П.* Один подход к решению оптимизационных задач с вложенной структурой. // Теорія оптимальних рішень. – К.: Ін-т кібернетики ім. В.М. Глушкова НАН України. – 2007. – № 6. – С. 90–99.
111. *Лаптин Ю.П., Крошко Д.Л.* Некоторые нелинейные оптимизационные задачи сетевой структуры // Теорія оптимальних рішень. – К.: Ін-т кібернетики ім. В.М. Глушкова НАН України. – 2009. – № 8. – С. 127–135.
112. *Лаптин Ю.П.* Один подход к решению нелинейных задач оптимизации с ограничениями // Кибернетика и системный анализ. – 2009. – № 3. – С. 182–187.
113. *Laptin Yu., Vinogradov A.* Exact discriminant function design using some optimization techniques // Classification, Forecasting, Data Mining International. Book Series “INFORMATION SCIENCE & COMPUTING”, Number 8, Sofia, Bulgaria. – 2009. – P. 14–19.
114. *Laptin Yu, Zhuravlev Yu., Vinogradov A.* Minimization of empirical risk in linear classifier problem // New Trends in Classification and Data Mining, ITHEA, Sofia, Bulgaria, 2010. – P. 9–15.
115. *Laptin Yu. P., Likhovid A. P., and Vinogradov A. P.* Approaches to Construction of Linear Classifiers in the Case of Many Classes // Pattern Recognition and Image Analysis. – 2010. – Vol. 20. – No. 2. – P. 137–145.
116. *Laptin Yu. P., Zhuravlev Yu., Vinogradov A. P.* Minimization of empirical risk in linear classifier problem // New Trends in Classification and Data Mining, ITHEA, Sofia, Bulgaria. – 2010. – P. 9–15.
117. *Лаптин Ю.П., Лиховид А.П., Стрюкова Н.Н.* Решение некоторых задач планирования в условиях неопределенности // Теорія оптимальних рішень.

- К.: Ін-т кібернетики ім. В.М. Глушкова НАН України. – 2010. – № 9. – С. 62–71.
118. *Лаптин Ю.П., Лиховид А.П.* Использование выпуклых продолжений функций для решения нелинейных задач оптимизации // Управляющие системы и машины. – 2010. – № 6. – С. 25–31.
119. *Лаптин Ю.П.* Некоторые вопросы использования негладких штрафных функций // Теорія оптимальних рішень. – К.: Ін-т кібернетики ім. В.М. Глушкова НАН України. – 2011. – № 10. – С. 127–135.
120. *Лаптин Ю.П., Журбенко Н.Г., Левин М.М., Волковицкая П.И.* О разработке программного обеспечения задач оптимального проектирования теплоэнергетических установок // Кибернетика и системный анализ. – 2011. – № 1. – С. 116–127.
121. *Лаптин Ю.П., Журавлев Ю.И., Виноградов А.П.* Минимизация эмпирического риска и задачи построения линейных классификаторов // Кибернетика и системный анализ. – 2011. – № 4. – С. 155–164.
122. *Лаптин Ю.П., Журавлев Ю.И., Виноградов А.П.* Построение нелинейных классификаторов в случае многих классов // *Applicable Information Models*. ITNEA, Sofia, 2011. – P. 7–13.
123. *Лаптин Ю.П., Бардадым Т.А.* Некоторые подходы к регуляризации нелинейных задач оптимизации // Проблемы управления и информатики. – 2011. – № 3. – С. 57–68.
124. *Лаптин Ю.П.* Некоторые вопросы определения коэффициентов негладких штрафных функций // Теорія оптимальних рішень. – К.: Ін-т кібернетики ім. В.М.Глушкова НАН України. – 2012. – С. 73–79.
125. *Лаптин Ю.П.* Использование штрафных функций для решения некоторых задач оптимизации // Теорія оптимальних рішень. – К.: Ін-т кібернетики ім. В.М. Глушкова НАН України. – 2013. – С. 95–101.

126. *Laptin Yu., Zhuravlev Yu., Vinogradov A., Zhurbenko N., Likhovid A.* Nonsmooth optimization methods In the problems of constructing a linear classifier // International Journal “Information Models and Analyses”. – 2012. – Vol. 1. – № 2. – P. 103–111.
127. *Laptin Yu., Zhuravlev Yu., Vinogradov A., Likhovid A.* A comparison of some approaches to the recognition problems in case of two classes // Int. Journal "Information models and analyses". – 2013. – Vol. 2. – No 2. – P. 103–111. – ISSN 1314-6416.
128. *Laptin Yu., Zhuravlev Yu., Vinogradov A* A Comparison of Some Approaches to Classification Problems, and Possibilities to Construct Optimal Solutions Efficiently // Pattern Recognition and Image Analysis. – 2014. – Vol. 24. – No. 2. – P. 189–195.
129. *Лаптин Ю.П.* Точные штрафные функции в схемах декомпозиции по переменным // Теорія оптимальних рішень. – К.: Ін-т кібернетики ім. В.М. Глушкова НАН України. – 2014. – С. 39–48.
130. *Лаптин Ю.П.* Решение невыпуклых задач оптимизации с использованием точных штрафных функций // Компьютерная математика. – К.: Ін-т кібернетики ім. В.М. Глушкова НАН України. – 2014. – № 1. – С. 119–130.
131. *Лаптин Ю.П.* Вопросы построения точных штрафных функций // Вестн. С.-Петербур. ун-та. Сер. 10: Прикладная математика. – 2013. – Вып. 4. – С. 21–31.
132. *Определение молекулярных подклассов глиобластом на основе анализа экспрессии генов.* В.В. Дмитренко, А.В. Ершов, П.И. Стецюк, Ю.П. Лаптин и др. // ISSN 0564-3783. Цитология и генетика. – 2014. – Т. 48. – № 6. – С.45–55.
133. *Левин А.Ю.* Об одном алгоритме минимизации выпуклых функций // Доклады Академии Наук СССР. – 1965. – Т. 160. – С. 1244–1247.

134. Мазуров Вл.Д. Математические методы распознавания образов.– Екатеринбург: Уральский государственный университет им. А.М. Горького, 2010. – 101 с.
135. *Математическое моделирование и оптимизация в задачах оперативного управления тепловыми электростанциями* / А.М. Клер, Н.П. Деканова, С.К. Скрипкин и др. – Новосибирск: Наука, 1997. – 120 с.
136. *Математическое моделирование и технико-экономические исследования энерготехнологических установок синтеза метанола* / А.М. Клер, Э.А. Тюрина. – Новосибирск: Наука, 1998. – 127 с.
137. *Местецкий Л.М.* Математические методы распознавания образов. – <http://www.intuit.ru/department/graphics/imageproc/>
138. *Методи негладкої оптимізації у спеціальних задачах класифікації* / П.І. Стецюк, О.А. Березовський, М.Г. Журбенко, Д.О. Кропотов – Київ, 2009. – 28 с. – (Препр./НАН України. Ін-т кібернетики ім. В.М. Глушкова; 2009–1).
139. *Методы оптимизации сложных теплоэнергетических установок* / А.М. Клер, Н.П. Деканова, Т.П. Щеголева и др. – Новосибирск: Наука. Сиб. изд. фирма РАН, 1993. – 116 с.
140. *Мину М.* Математическое программирование. – М.: Наука, 1990. – 486 с.
141. *Михалевич В.С., Трубин В.А., Шор Н.З.* Оптимизационные задачи производственно-транспортного планирования. – М.: Наука, 1986. – 260 с.
142. *Михалевич М.В., Сергиенко И.В.* Моделирование переходной экономики. Модели, методы, информационные технологии. – Киев: «Наукова думка», 2005. – 670 с.
143. *Нурминский Е.А.* Численные методы решения детерминированных и стохастических минимаксных задач/Е.А. Нурминский.– К.:Наука. думка, 1979. –161с.

144. *Нурминский Е.А.* Численные методы выпуклой оптимизации. – М.: Наука, 1991. – 168 с.
145. *Оптимизация развития и функционирования автономных энергетических систем / А.М. Клер, Н.П. Деканова, Б.Г. Санеев и др.* – Новосибирск: Наука, 2001. – 144 с.
146. *Паршин А.А., Митор В.В., Безгрешнов А.Н. и др.* Тепловые схемы котлов. – М.: Машиностроение, 1987. – 222 с.
147. *Петунин Ю.И., Шульдешов Г.А.* Проблемы распознавания образов с помощью линейных дискриминантных функций Фишера // Кибернетика. – 1979. – № 6. – С. 134–137.
148. *Петунин Ю.И., Рублев Б.В.* Распознавание образов с помощью квадратичных дискриминантных функций // Вычисл. и прикл. математика. – 1996. – Вып. 80. – С. 89–104.
149. *Петунин Ю.И., Ключин Д.А. и др.* Методы распознавания групповой принадлежности, основанные на доверительных границах, и их апробация в клинической онкологии // Кибернетика и системный анализ. – 2009. – № 4. – С. 19–30.
150. *Поляк Б.Т.* Введение в оптимизацию. – М.: Наука, 1983. – 384 с.
151. *Попырин Л.С.* Математическое моделирование и оптимизация теплоэнергетических установок. – М.: Энергия, 1978. – 416 с.
152. *Попырин Л.С., Самусев В.И., Эпельштейн В.В.* Автоматизация математического моделирования теплоэнергетических установок. – М.: Наука, 1981. – 236 с.
153. *Пшеничный Б.Н., Данилин Ю. М.* Численные методы в экстремальных задачах. – М.: Наука, 1975. – 319 с.
154. *Пшеничный Б.Н.* Выпуклый анализ и экстремальные задачи. – М.: Наука, 1980. – 319 с.

155. *Пшеничный Б.Н.* Метод линеаризации. – М.: Наука. – 1983. – 136 с.
156. *Пшеничный Б.Н., Ненахов Э.И., Кузьменко В.Н.* Комбинированный метод решения общей задачи выпуклого программирования // Кибернетика и систем. анализ. – 1998. – № 4 – С. 121–134.
157. *Ржевский С. В.* Монотонные методы выпуклого программирования. – Киев: Наук. думка, 1993. – 316 с.
158. *Рублев Б.В., Петунин Ю.И., Литвинко П.Г.* Структура гомотетичных линейно разделимых множеств в n -мерном евклидовом пространстве. – Кибернетика и системный анализ. Ч.1 – 1992. – № 1. – С. 3–15; Ч.2 – 1992. – № 2. – С. 23–33.
159. *Сергиенко И. В.* Математические модели и методы решения задач дискретной оптимизации. – Наукова думка, 1988.
160. *Сергиенко И. В., Шило В. П.* Задачи дискретной оптимизации: проблемы, методы исследования, решения //К.: Наук. думка. – 2003.
161. *Сергиенко И. В., Гуляницкий Л. Ф.* Метаэвристический метод деформированного многогранника в комбинаторной оптимизации //Кибернетика и системный анализ. – 2007. – №. 6. – С. 70-79.
162. *Сергиенко И. В., Шило В. П.* Проблемы дискретной оптимизации: сложные задачи, основные подходы к их решению //Кибернетика и системный анализ. – 2006. – №. 4. – С. 3-25.
163. *Система Крокус* – автоматизированное проектирование, комплексные расчеты, оптимизация котельных установок / Волковицкая П.И., Журбенко Н.Г., Лаптин Ю.П. и др. // Энергетика и электрификация. – 2001. – № 7. – С. 45–48.
164. *Системные исследования проблем энергетики* / Л.С. Беляев, Б.Г. Санеев, С.П. Филиппов и др.; Под ред. Н.И. Воропая. – Новосибирск: Наука. Сиб. изд. фирма РАН, 2000. – 558 с.

165. *Соломон Д.И.* Дробное программирование и недифференцируемая оптимизация. – Кишинэу: Эврика, 2010. – 554 с.
166. *Стецюк П.И.* Метод центров тяжести простых тел // Кибернетика и системный анализ. – 1996. – № 5. – С. 117–138.
167. *Стецюк П.И., Нурминский Е.А.* Негладкий штраф и субградиентные алгоритмы для решения задачи проекции на политоп // Кибернетика и системный анализ. — 2010. — № 1. — С. 59–63.
168. *Стецюк П.И.* Методы эллипсоидов и r -алгоритмы. – Кишинэу: Эврика, 2014. – 488 с.
169. *Стохастическое программирование и его приложения.* Под ред. П.С. Кнопова, В.И. Зоркальцева. – Иркутск, 2012. – 492 с.
170. *Стрекаловский А. С., Груздева Т. В., Орлов А. В.* Численное решение задачи о полиэдральной отделимости // Автоматика и телемеханика. – 2015. – №. 10. – С. 113-130.
171. *Тепловой расчет* котельных агрегатов (нормативный метод). – М.: Энергия, 1973. – 295 с.
172. *Теплосиловые системы: Оптимизационные исследования.* – Новосибирск: Наука, 2005. – 236 с.
173. *Цурков В. И.* Декомпозиция в задачах большой размерности. – Наука. Гл. ред. физ.-мат. лит., 1981.
174. *Шлезингер М., Главач В.* Десять лекций по статистическому и структурному распознаванию. – Киев: Наук. думка, 2004. – 545 с.
175. *Шлейфер Б.М., Литвак Д.Б.* Оптимальные решения при проектировании промперегревателей. // Энергомашиностроение. – 1978. № 2. – С.4–6.

176. Шор Н. З., Журбенко Н. Г. Метод минимизации, использующий операцию растяжения пространства в направлении разности двух последовательных градиентов // Кибернетика. – 1971. – № 3. – С. 51–59.
177. Шор Н.З. О классе почти-дифференцируемых функций и одном методе минимизации функций этого класса // Кибернетика. – 1972. – № 4. – С. 9–17.
178. Шор Н.З. Метод отсечения с растяжением пространства для решения задач выпуклого программирования // Кибернетика. – 1977.– № 1. – С. 94–95.
179. Шор Н.З., Гершович В.И. Об одном семействе алгоритмов для решения задач выпуклого программирования // Кибернетика. – 1979. – № 4. – С. 62–67.
180. Шор Н.З. Методы минимизации недифференцируемых функций и их приложения. – Киев: Наукова думка, 1979. – 199 с.
181. Шор Н.З., Соломон Д.И. Декомпозиционные методы в дробно-линейном программировании. – Кишинев: Штиинца, 1989. – 204 с.
182. Шор Н.З., Стеценко С.И. Квадратичные экстремальные задачи и недифференцируемая оптимизация. – Киев: Наук. думка, 1989 – 208 с.
183. Шор Н.З., Березовский О.А. Построение эллипсоида максимального объема, вписанного в многогранник, с использованием последовательного растяжения пространства // Кибернетика и вычислительная техника. – Вып. 93. – Киев: «Наукова думка», 1992. – С. 1–6.
184. Shor N. Z. Nondifferentiable Optimization and Polynomial Problems. – Amsterdam / Dordrecht / London: Kluwer Academic Publishers, 1998.–381 p.
185. Юдин Д. Б., Гольштейн Е. Г. Линейное программирование. Теория, методы и приложения. – М.: Наука, 1969. – 424 с.
186. Юдин Д.Б., Немировский А.С. Информационная сложность и эффективные методы решения выпуклых экстремальных задач //

- Экономика и математические методы. – 1976. – Т. 12 – Вып. 2. – С. 357–369.
187. *Лаптин Ю.П., Журбенко Н.Г., Кузьменко В.Н.* Использование декомпозиции по переменным для нелинейных задач оптимизации // Декомпозиционные методы в математическом моделировании и информатике. Тезисы докладов 2-й Московской конференции (21–24 июня 2004 г.). – Москва: ВЦ РАН, 2004. – С. 64–65.
188. *Лаптин Ю. П., Журбенко Н. Г., Кузьменко В. Н.* Декомпозиция блочных задач оптимизации со связывающими переменными // Матеріали VII Міжнародної науково-технічної конференції “Системний аналіз та інформаційні технології.” (28 червня – 02 липня 2005 р., м. Київ, Україна). С. 49.
189. *Лаптин Ю.П., Журбенко М.Г., Коваленко Д.А., Левін М.М., Волковицька П.И.* Оптимальное проектирование сложных объектов энергетики // Системний аналіз та інформаційні технології. Матеріали XI Міжнародної науково-технічної конференції (26-30 травня 2009 р.), Київ, 2009. – С. 137.
190. *Лаптин Ю.П.* Некоторые подходы к решению нелинейных задач оптимизации с ограничениями // Праці міжнародного симпозіуму «Питання оптимізації обчислень (ПОО-XXXV)», том 1, Київ. – 2009. – С.436–440.
191. *Лаптин Ю.П., Лиховид А.П., Стрюкова Н.Н.* Сценарный подход к планированию инвестиций в электроэнергетику в условиях неопределенности // Системний аналіз та інформаційні технології: матеріали 12-ї Міжнародної науково-технічної конференції SAIT 2010, Київ. 25–29 травня 2010. – С. 277.
192. *Лаптин Ю.П., Журавлев Ю.И., Виноградов А.П.* Задачи построения линейных классификаторов в случае многих классов // Российская конференция «Дискретная оптимизация и исследование операций»:

- Материалы конференции (Алтай, 27 июня – 3 июля 2010). — Новосибирск: Изд-во Ин-та математики, 2010. – С. 187.
193. *Лаптин Ю.П., Журавлев Ю.И., Виноградов А.П.* Задачи построения линейных и нелинейных классификаторов в случае многих классов // Труды конференции ММРО-15, г. Петрозаводск, 11–17 сентября 2011 г. – С. 72–74.
194. *Лаптин Ю.П., Бардадым Т.А., Щетинин Е.И.* Использование некоторых точных вспомогательных функций в задачах оптимизации // Труды III международной конференции „Математическое моделирование, оптимизация и информационные технологии”, г. Кишинев, Республика Молдова, 19–23.03.2012. – С. 394–404.
195. *Laptin Yu. P., Bardadym T.A.* Some Approaches for Construction Exact Auxiliary Functions in Optimization Problems // Международная конференция «Конструктивный негладкий анализ и смежные вопросы», тезисы докладов, Санкт-Петербург, 18–23 июня 2012 г. – С. 102–104.
196. *Лаптин Ю.П., Шор Е.И.* Некоторые подходы к формированию точных штрафных функций // Праці міжнародної наукової конференції „Питання оптимізації обчислень (ПОО-XL)”, Київ: Інститут кібернетики імені В.М. Глушкова НАН України, 2013. – С. 143–144.
197. *Лаптин Ю.П., Бардадым Т.А.*, Использование выпуклых продолжений при формировании точных штрафных функций // Праці міжнародної наукової конференції „Питання оптимізації обчислень (ПОО-XL)”, Київ: Інститут кібернетики імені В.М. Глушкова НАН України, 2013. – С. 141–142.
198. *Лаптин Ю.П.* Определение коэффициентов штрафа и формирование точных штрафных функций // VII Московская международная конференция по исследованию операций (ORM2013), Труды, Том 2, Москва, 2013. – С. 39–41.