Численные аспекты и реализация субградиентных алгоритмов с преобразованием пространства.

Н.Г. ЖУРБЕНКО

Институт кибернетики им. В.М. Глушкова НАН Украины, Киев, Украина
zhurbnick@gmail.com

Моделирование и Оптимизация в Транспорте и Логистике", Киев 31 октября 2023 г.

1 ноября 2023 г.

OUTLINE

- Дмитрий Соломон.
- Параметры r-алгоритма, прцедура выбора шаговых множителей.
- Регуляризация матрицы преобразования пространства.
- Квазиньютоновские методы.
- Простейшие ограничения.
- Точные штрафные функции.
- Декомпозиция в задачах блочной структуры.
- Программные реализации для сложных прикладных задач оптимизации.

Дмитрий Соломон.

Дмитрий Соломон (1951 - 2022)

Дмитрий Соломон (???? - 2023?)

1976 г.
Соломон Д.И. Дробное программирование и недифференцируемая оптимизация. –Кишенэу:Эврика. 2010. – 420 с.

Академия транспорта, информатики и коммуникаций.

Решение транспортной задачи $\Rightarrow$ алгоритм негладкой оптимизации Н.З.Шор (1962 г.)

Параметры r-алгоритма, процедура выбора шаговых множителей.

Теоретически обоснованный вариант $\alpha(\varepsilon$-алгоритм. (Баку)
1. h_0 – начальный шаг. $h_0 \approx |x_0 - x^*|$.
2. α – коэффициент растяжения; $\alpha \approx 2.0$.
(Варианты с программным управлением коэффициентами растяжения $r(\sigma)$-алгоритмы.) (Львов)
3. $q_1, q_2, nStep$ параметры регулировки шага :
$0 < q_1 \leq 1$; $q_2 \geq 1$; $nStep \geq 1$.
($r(\sigma)$-алгоритмы могут использоваться без процедуры одномерного спуска, с постоянным шагом.)

Регуляризация матрицы преобразования пространства.

На итерации r-алгоритма используется операция деления на $|B_k^* g_k|$ – норму субградиента в преобразованном пространстве. Ошибка «деление на нуль». (Задача с заданной точностью еще не решена, а величина $|B_k^* g_k|$ принимает недопустимо малое значение.)

Процедура регуляризации матрицы преобразования пространства, предназначенная для предотвращения ее вырождения.

Вместо оператора $R_\beta(\eta)$ будем использовать оператор

$$\tilde{R}_\beta(\eta) = (1/\sqrt[n]{\beta}) R_\beta(\eta),$$

для которого $\det(\tilde{R}_\beta(\eta)) = 1$ $(\det(R_\beta(\eta)) = \beta)$.

Регуляризация матрицы преобразования пространства.

Оператор $\tilde{R}_\alpha(\eta)$ фактически можно интерпретировать как «растяжение» пространства оператором $R_\alpha(\eta)$ и равномерное сжатием по всем направлениям с коэффициентом $\beta = 1/\sqrt[n]{\alpha}$.
Заметим, что это дополнительное сжатие не изменяет структуру поверхностей уровня функции – изменяется лишь их масштаб.
Квадратичная овражная функция.
Основная цель преобразования пространства превратить "вытянутые" эллипсоиды поверхностей уровня в сферические.
Кажется естественным равенство объемов этих эллипсоидов и соответствующих им шаров в преобразованном пространстве.
Алгоритм с использованием оператора $\tilde{R}_\alpha(\eta)$ эквивалентен алгоритму с оператором $R_\alpha(\eta)$.

Регуляризация матрицы преобразования пространства.

Аналогичная ситуация в методе эллипсоидов.
Преобразование эллипсоида локализации решения в шар. Но в шар какого радиуса? Теоретически неважно.
Юдин-Немировский (первая публикация метода): сохранение объемов ($det(B) = 1$, радиус уменьшается)
Шор: минимальный радиус при использовании оператора растяжения с $\alpha > 1$ ($det(B) < 1$,радиус увеличивается).
Хачиян: радиус константа ($det(B) < 1$)

Квазиньютоновские методы.

В. Давидон 1959 метод переменной метрики.
W. Davidon Variable metric method for minimization. Argonne National Laboratory. Research and Development Report 5990 (1959).
A. Rodomanov, Y. Nesterov New Results on Superlinear Convergence of Classical Quasi-Newton Methods (2020), arXiv:2004.14866
(https://www.youtube.com/watch?v=0j45FpsLKyI)

$$x_k = x_{k-1} - h_k H_k g_{k-1}, k = 1, 2, \ldots, . \tag{1}$$

$h_k > 0; \quad g(x_{k-1} = \nabla f(x_{k-1}); \quad H_k \succ 0.$ $H_{k+1} =$H$_k+\triangle\ H_k$.
Квазиньютоновское условие:

$$H_{k+1}(g_k - g_{k-1}) = -h_k H_k g_{k-1}.$$

$r(\propto)$ ($\beta = 0$) $\Rightarrow$ алгоритм сопряженных направлений. (2)

Квадратичная функция $f(x) = (Cx, x) + (b, x)$
$x_1 = x_0 + hp$ Преобразование : $Y = R_\alpha(\eta)X$.
$f(x) \Rightarrow \mathsf{f}^y(y) = 1/2(C^y y, y)$, где $C^y = R_\beta(\eta)CR_\beta(\eta)$, $\beta = 1/\alpha$.
$p \Rightarrow \mathsf{p}^y = R_\alpha(\eta)p$.
$R_\alpha(\eta) \Rightarrow p^y$ **собственный вектор матрицы** C^y :

$$C^y p^y = \lambda p^y,\ \lambda > 0 \tag{3}$$

$$R_{\alpha^2}(\eta)p = (1/\lambda)\xi_1, \tag{4}$$

где $\xi_1 = (g_1 - g_0)/h$.

Квазиньютоновские методы.

Обозначения:
$\theta-$ угол между векторами p и ξ_1, $0 < \theta < \pi/2$ ($tg(\theta) = |g_1|/|g_0|$);
$\psi-$ угол между векторами p и η.
Лемма 1.
Пусть $\alpha \geq 1$. Тогда оператор $R_\alpha(\eta)$ обеспечивает выполнение соотношения (4) тогда и только тогда, когда

$$\theta < \psi < \pi/2 \tag{5}$$

$$\alpha^2 = tg(\psi)/tg(\psi - \theta) \tag{6}$$

Выбор величины угла ψ однозначно определяет оператор $R_\alpha(\eta)$.

Квазиньютоновские методы.

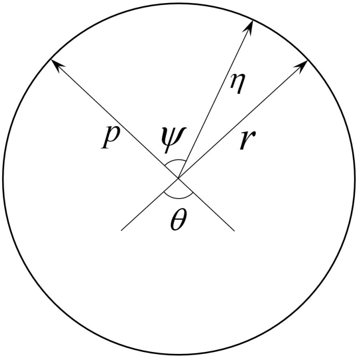
Из (5-6) следует, что если $\psi \to \theta$, или $\psi \to \pi/2$, то $\alpha \to \infty$. Поэтому особый интерес представляет выбор такого направления растяжения η^* (угла ψ^*), при котором достигается минимальное значение коэффициента растяжения α^*. Такой выбор определяется решением задачи минимизации α^2 по ψ при ограничении (5). Результат решения этой задачи:

$$\psi^* = \pi/4 + \theta/2; \quad \alpha^* = tg(\psi^*). \tag{7}$$

Квазиньютоновские методы.

Для квадратичной функции:
алгоритм является методом сопряженных направлений;
все направления движения в преобразованном пространстве взаимно ортогональны и являются собственными векторами матрицы C^y.
Квазиньютоновские алгоритмы $\Rightarrow$ дополнительное преобразование пространства, обеспечивающее равенство всех собственных чисел матрицы квадратичной формы.

Квазиньютоновские методы.

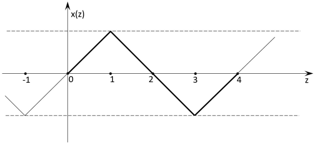


$$\psi \to 0 \quad \alpha \to \infty;$$
$$\psi \to \theta \quad \alpha \to \infty.$$

$$p = -g_0 / |g_0|;$$
$$r = (g_1 - g_0)/|g_1 - g_0|.$$

Простейшие ограничения.

Не отрицательность:
$x \geq 0 \Rightarrow x = |z|.$
Интервал:
$-1 \leq x \leq 1$



Точные штрафные функции.

$$minf(x)$$
$$g_k(x) \le 0, k = \overline{1,m}.$$

$$G(x) = max\{0; max\{g_k(x) | k = \overline{1,m}\}\}$$

$$\min\{F(x) = f(x) + RG(x)\}$$

$$R?$$

$$R > \sum_k u_k^*.$$

$$g(x) = \begin{cases} g_f(x),\ if\, G(x) \le 0 \\ g_G(x),\ if\, G(x) > 0 \end{cases},$$

где $g_f(x), g_G(x)$ субградієнти функцій $f(x), G(x)$.

Декомпозиция в задачах блочной структуры.

Блочные задачи математического программирования Н.З.Шор

$$\min f(x) \tag{8}$$

$$x \in D \tag{9}$$

$$g_i(x) \leq 0, \qquad i = \overline{1,m} \tag{10}$$

$L(x,u)$ – функция Лагранжа,

$$L(x,u) = f(x) + \sum_{i=1}^{m} u(i) g_i(x), \tag{11}$$

Двойственная задача относительно (10):

$$\max_{u \geq 0} \psi(u), \tag{12}$$

где

$$\psi(u) = \min_{x \in D} L(x,u). \tag{13}$$

Декомпозиция в задачах блочной структуры.

$\psi(u)$ вогнутая не дифференцируемая функция .
Ввнутренняя задача декомпозиции:

$$\min_{x \in D} L(x, u), \tag{14}$$

$x(u)$ – $\forall$ решение задачи (14) тогда

$$g(x(u)) \in \partial\psi(u). \tag{15}$$

$x(u)$ единственное решение $\Rightarrow$ $\psi(u)$ дифференцируема в т. u.
Блочная структура $f(x)$ и блочная структура D $\Rightarrow$ решение (14) сводится к автономному решению для отдельных блоков.
u^* – решение двойственной задачи.
Если $x(u^*)$ единственное решение задачи (14) $\Rightarrow$ $x(u^*) = x^*$ – решение исходной задачи.
Контроль точности: $f(x(u^*)) = \psi(u^*)$ + точность выполнения ограничений.

Декомпозиция в задачах блочной структуры.

1. Решение двойственной задачи.
Учет $u_k \geq 0 \Rightarrow u_k = |z_k|$.
"Несущественные" ограничения.
Старт $z_k = 0$ $(u_k = 0.)$
Если $g_k < 0 \Rightarrow g_k = 0$ (процедура "фиксация нуля").
Если $|z_k| < \varepsilon \Rightarrow z_k = 0$: включаем процедуру "фиксация нуля".
2. Решение исходной задачи. Если $x(u^*)$ единственное решение задачи (14) $\Rightarrow x(u^*) = x^*$ – решение исходной задачи. Поэтому на уровне построения модели стремимся к единственности ее решения.
Например обеспечить строгую выпуклость целевого функционала (малые квадратичные возмущения).

Декомпозиция в задачах блочной структуры.

Задача линейного программирования. Типично: число "связывающих" переменных $<<$ числа переменных (блоков) . Поэтому для большинства блоков решение подзадач однозначно.
И после их выведения из модели получаем задачу относительно небольшой размерности.

Метод усреднения (усреднение по Чезаро).

$u_0 = \widetilde{u}$,

$$u_{k+1} = u_k + h_k g(x(u_k)) \tag{16}$$

$$h_k > 0, \quad h_k \to 0 \quad \sum_{k=0}^{\infty} h_k = \infty. \tag{17}$$

$$z_N = (1/\sum_{k=0}^{N} h_k) \sum_{k=0}^{N} h_k x(u_k) \tag{18}$$

Любая предельная точка $\{z_N\}$ является решением исходной задачи .

На практике $h_k = h$ – constant:

z$_N = (1/N) \sum_{k=0}^{N} x(u_k)$

Программные реализации и прочее

C++
Базы данных.

Алгоритм с постоянным шагом (Шор Н.З.):
$x_{k+1} = x_k - hg_k/|g_k|$.
$D(r) \subset X^*$, $h < 2r \Rightarrow \exists\, k^* : x_{k*} \in X^*$.

ε-субградиенты.

СПАСИБО

zhurbnick@gmail.com