

Курсовая работа по теме
«Методы оптимизации»

Дмитрий Джус

2008

Содержание

1. Введение	2
1.1. Предмет оптимизации	2
1.2. Выпуклость множеств и функций	2
1.2.1. Понятие выпуклости	2
1.2.2. Исследование функций и множеств на выпуклость . . .	3
1.2.3. Свойства выпуклых функций	5
1.3. Проблемы, возникающие в задачах оптимизации	5
1.3.1. Явление овражности	5
1.3.2. Явление многоэкстремальности	7
2. Симплекс-метод	8
2.1. Алгоритм симплекс-метода	8
2.1.1. Метод искусственных переменных	10
2.1.2. Геометрическая интерпретация	10
2.2. Практический пример	13
3. Безусловная оптимизация	15
3.1. Теоретические сведения	15
3.1.1. Постановка задачи	15
3.1.2. Градиентные методы	15
3.1.3. Функция релаксации	16
3.1.4. Метод ПГС	16
3.2. Описание метода RELCH	18
3.2.1. Использование полиномов Чебышёва в функции релаксации	18
3.2.2. Реализация метода	19
3.2.3. Выбор параметра метода	19
3.2.4. Трудности, возникающие при реализации RELCH	20
3.3. Тестовые задачи	22
3.3.1. Функция Розенброка	22
3.3.2. Функция Химмельблау	27
3.3.3. Экспоненциальная функция	30
3.3.4. Многопараметрическая оптимизация	35
3.4. Выводы	35
4. Условная оптимизация	36
4.1. Теорема Куна—Таккера	36
4.1.1. Теоретические сведения	36
4.1.2. Алгоритм метода	38
4.1.3. Нахождение точного аналитического решения	39
4.1.4. Использование метода	42
4.2. Метод Зойтендейка	43
4.2.1. Общая схема методов возможных направлений	43
4.2.2. Алгоритм метода Зойтендейка	43
4.2.3. Пример	45
4.2.4. Использование метода	47
4.3. Метод штрафных функций	48
4.3.1. Общие сведения	48
4.3.2. Метод внешних штрафных функций	48
4.3.3. Пример	49

А. Исходные тексты	52
А.1. Общая схема градиентных методов	52
А.2. Алгоритмы RELCH, GDRELCH, ПГС и ПГС ₂	53
Б. Информация о документе	56
В. Список литературы	56

Список иллюстраций

1	Выпуклость множеств	3
2	Выпуклая функция	4
3	Процесс поиска минимума функции	6
4	Овражная функция	6
5	Симплекс-метод	12
6	Подходящая функция релаксации	17
7	Функция релаксации метода ПГС	17
8	Функции релаксации на основе полиномов Чебышёва	18
9	Функция Розенброка	23
10	GDRELCH на функции Розенброка, $h = 0.003$	24
11	RELCH на функции Розенброка	26
12	Функция Химмельблау	28
13	RELCH на функции Химмельблау	29
14	Экспоненциальная функция	32
15	RELCH на экспоненциальной функции	33
16	GDRELCH на экспоненциальной функции	34
17	Стационарные точки задачи условной оптимизации	41
18	Задача условной оптимизации вблизи одной из стационарных точек	42
19	Метод Зойтендейка	47
20	Влияние коэффициента штрафа на вид целевой функции	50
21	Метод штрафных функций	51

Предмет работы

Настоящая курсовая работа посвящена методам оптимизации и экстремальным задачам.

В первой части рассмотрено решение задачи условной минимизации линейной функции с помощью симплекс-метода.

Кроме того, дано описание градиентного метода многопараметрической оптимизации с чебышёвскими функциями релаксации, предложена реализация метода на алгоритмическом языке и представлены результаты работы на тестовых функциях.

Последний раздел посвящён поиску экстремальных значений целевой функции при наличии ограничений с использованием теоремы Куна—Таккера, метода штрафных функций и метода возможных направлений Зойтендейка.

1. Введение

В данном разделе изложены предварительные сведения общего характера, которые будут использоваться в дальнейших частях работы.

1.1. Предмет оптимизации

Необходимость поиска экстремальных значений разнообразных функций встречается в практической деятельности повсеместно. В связи с этим разработка и исследование эффективных алгоритмов оптимизации, имеющих широкие области применения, является одним из важных направлений современной математики.

Обратимся к следующей общей формулировке задачи оптимизации:

$$f(x) \rightarrow \min \quad (1.1)$$

В данной работе рассматривается оптимизация действительных функций действительного переменного, $f: \mathbb{R}^n \rightarrow \mathbb{R}$.

Задача (1.1), где на возможные значения x не накладывается никаких ограничений, называется задачей *безусловной оптимизации*. В то же время, на практике приходится решать и задачи *условной оптимизации*:

$$f(x) \rightarrow \min \text{ при } x \in D \quad (1.2)$$

где D является некоторым подмножеством области определения $f(x)$.

ЗАМЕЧАНИЕ 1.1. Рассмотрение задачи на поиск *минимума* в качестве оптимизационной не ограничивает общности рассуждений в том смысле, что задача *максимизации* функции $f(x)$

$$f(x) \rightarrow \max$$

сводится к задаче минимизации противоположной функции $-f(x)$.¹

Рассмотрение задачи оптимизации и связанных с ней проблем будет продолжено после изложения важного понятия выпуклости.

1.2. Выпуклость множеств и функций

1.2.1. Понятие выпуклости

Интуитивно ясное понятие выпуклости опирается на наглядные геометрические представления, но вместе с тем имеет и чисто аналитическую формулировку.

Более подробное рассмотрение выпуклых множеств и функций представлено в [1], [7] и [9].

ОПРЕДЕЛЕНИЕ 1.2. Множество $D \subset \mathbb{R}^n$ называется *выпуклым*, если для $\forall x, y \in D, \forall \alpha \in [0; 1]$ выполняется

$$\alpha x + (1 - \alpha)y \in D$$

ОПРЕДЕЛЕНИЕ 1.3. Если в определении 1.2 для $\forall \alpha \in (0; 1)$ условие выполняется в смысле принадлежности ко *внутренности* D , то есть

$$\alpha x + (1 - \alpha)y \in D \setminus \partial D,$$

то множество D называется *строго выпуклым*.

¹Сходные соображения бывают справедливы и для ограничения $x \in D$, см. далее замечания 2.1, 4.1

ОПРЕДЕЛЕНИЕ 1.4. Множество называется *вогнутым*, если оно не является выпуклым.

Таким образом, отрезок, соединяющий две точки выпуклого множества, целиком принадлежит этому множеству, а две точки строго выпуклого множества соединяются отрезком, внутренние точки которого не принадлежат границе множества (рис. 1).

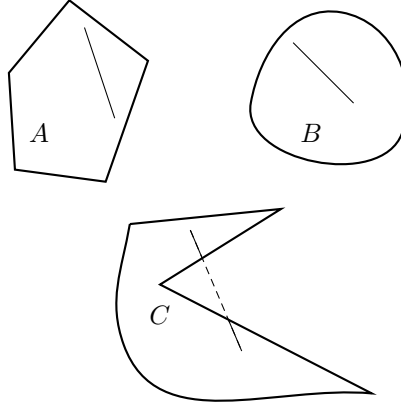


Рис. 1: Выпуклое множество A , строго выпуклое множество B и вогнутое множество C

ОПРЕДЕЛЕНИЕ 1.5. Функция $f(x), x \in \mathbb{R}^n$ называется *выпуклой*, если для $\forall x, y \in \mathbb{R}^n, \forall \alpha \in [0; 1]$ выполняется неравенство

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$$

ОПРЕДЕЛЕНИЕ 1.6. Если в определении 1.5 для $\forall \alpha \in (0; 1)$ неравенство выполняется строго, функция $f(x)$ называется *строго выпуклой*.

ОПРЕДЕЛЕНИЕ 1.7. Функция $f(x)$ называется *вогнутой*, если $-f(x)$ является выпуклой функцией.

Геометрический смысл определений 1.5 и 1.6 ясен: функция выпукла, если выпукла область над её графиком, и строго выпукла, если область над её графиком строго выпукла (рис. 2).²

ЗАМЕЧАНИЕ 1.8. Линейные функции выпуклы, но не строго выпуклы.

1.2.2. Исследование функций и множеств на выпуклость

Для доказательства выпуклости функций используются следующие теоремы.

ТЕОРЕМА 1.9. Для выпуклости дважды непрерывно-дифференцируемой функции одной переменной $f(x), x \in \mathbb{R}$ необходимо и достаточно, чтобы её вторая производная $f''(x)$ была ограничена и неотрицательна:

$$f''(x) \geq 0$$

При этом выполнение неравенства со знаком “ $>$ ” означает строгую выпуклость.

²Мнемоническое правило: выпуклые функции «улыбаются» \smile , а вогнутые — «хмурятся» \frown .

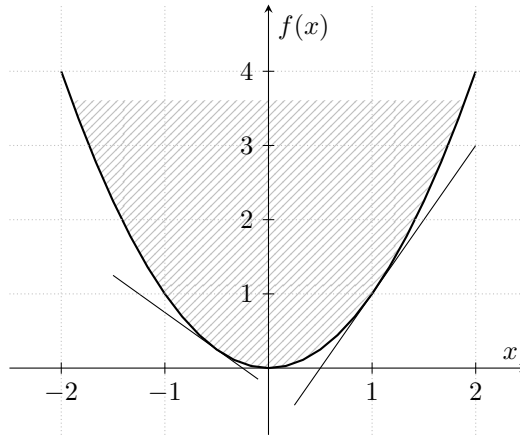


Рис. 2: Выпуклая функция $f(x) = x^2$ и касательные прямые к ней в точках $(-\frac{1}{2}, \frac{1}{4})$ и $(1, 1)$

На случай функции многих переменных теорема 1.9 обобщается следующим образом.

ТЕОРЕМА 1.10. *Для выпуклости дважды непрерывно-дифференцируемой функции нескольких переменных $f(x), x \in \mathbb{R}^n$ необходимо и достаточно, чтобы её матрица Гессе была ограничена и неотрицательно определена³. При этом положительная определённость матрицы Гессе означает строгую выпуклость функции.*

Для исследования симметричных матриц на определённость используются различные критерии:

- Для положительной определённости матрицы необходимо и достаточно, чтобы все её собственные числа были положительны.
- Критерий Сильвестра гласит, что матрица положительно определена тогда и только тогда, когда все её угловые миноры положительны. При этом, однако, неотрицательность угловых миноров не означает неотрицательной определённости всей матрицы.

ТЕОРЕМА 1.11. *Для выпуклости непрерывно-дифференцируемой функции $f(x)$ необходимо и достаточно, чтобы её график всеми точками лежал над любой своей касательной или на ней. Для строгой выпуклости функции её график должен лежать строго над любой своей касательной (кроме точки касания).*

Рисунок 2 служит иллюстрацией теоремы 1.11: график строго выпуклой функции $f(x) = x^2$ лежит над своими касательными.

В случае функции многих переменных в теореме 1.11 рассматривается касательная гиперплоскость.

ЗАМЕЧАНИЕ 1.12. До сих рассматривались функции, выпуклые на всей области определения. Вместе с тем можно рассматривать и понятие функции $f(x), x \in \mathbb{R}^n$, выпуклой лишь на подмножестве $D \subset \mathbb{R}^n$ своей области определения, то есть такой, что $\forall x, y \in D \subset \mathbb{R}^n, \forall \alpha \in [0; 1]$ выполняется неравенство

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$$

³Неотрицательно определённые матрицы также называют положительно полуопределёнными.

При этом все теоремы, относящиеся к выпуклым функциям, переформулируются для соответствующих множеств D .

Наконец, рассмотрим теоремы, которые оказываются полезным при доказательстве выпуклости допустимого множества в задачах условной оптимизации.

ТЕОРЕМА 1.13. *Пересечение любого числа выпуклых множеств есть выпуклое множество.*

Следующая теорема является прямым следствием замечания 1.8 и теоремы 1.13.

ТЕОРЕМА 1.14. *Если $g_j(x), j = \overline{1, m}$ — выпуклые функции, то множество D , заданное условиями $g_j(x) \leq b_j$, является выпуклым.*

1.2.3. Свойства выпуклых функций

ТЕОРЕМА 1.15. *Сумма выпуклых функций также выпукла.*

ТЕОРЕМА 1.16. *Сумма выпуклых и строго выпуклых функций строго выпукла.*

ТЕОРЕМА 1.17. *Функция сохраняет выпуклость при сложении с произвольной константой или умножении на положительную константу.*

Следующие теоремы позволяют судить о существовании, единственности и даже местоположении экстремальных точек выпуклой на множестве D функции, что особенно важно ввиду существования проблемы многоэкстремальности в задачах оптимизации (о ней пойдёт речь в разделе 1.3.2).

ТЕОРЕМА 1.18. *Если точка $x_0 \in D \subset \mathbb{R}^n$ — локальный минимум функции $f(x), x \in \mathbb{R}^n$, выпуклой на множестве D , то $\min_D f(x) = f(x_0)$. Если точка $x_0 \in D$ — локальный минимум функции $f(x)$, вогнутой на множестве D , то $\max_D f(x) = f(x_0)$.*

Таким образом, выпуклая на D функция не может иметь в этом множестве локальных минимумов, в которых достигаются разные значения функции. Поэтому для определения минимального значения выпуклой функции достаточно найти хотя бы одну точку её локального минимума. Аналогичное утверждение справедливо и для максимального значения вогнутой функции.

ТЕОРЕМА 1.19. *Если функция строго выпукла в D , то её точка минимума — единственная. Если функция строго вогнута в D , то её точка максимума — единственная.*

ТЕОРЕМА 1.20. *Если функция строго выпукла в D , то её точки максимума лежат на границе D . Если функция строго вогнута в D , то её точки минимума лежат на границе D .*

1.3. Проблемы, возникающие в задачах оптимизации

Обозначим трудности, проявляющиеся в ходе решения различных экстремальных задач.

1.3.1. Явление овражности

Большое число различных методов минимизации тем или иным образом используют тот факт, что антиградиент функции $-f'(x)$ указывает на направление её наискорейшего убывания.

Метод простого градиентного спуска⁴ (ПГС) реализует эту идею самым простым образом, а именно: поиск минимума функции осуществляется путём последовательного продвижения вдоль её антиградиента. На рисунке 3 демонстрируется, как ПГС, начиная движение из точки $(0, 0)$, находит локальный минимум функции⁵ в точке $(3, 2)$.

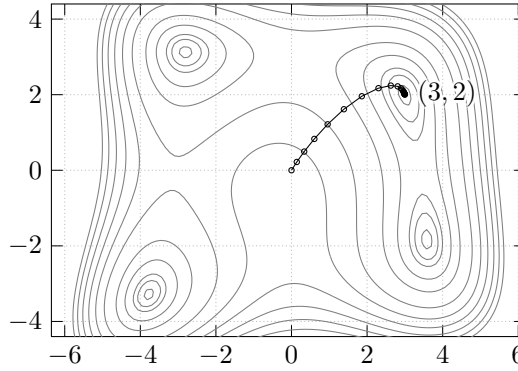


Рис. 3: Процесс поиска минимума функции

Тем не менее, на практике нередко случаи, когда следование направлению $-f'(x)$ почти никак не позволяет приблизиться к минимуму целевой функции.

Подобная ситуация изображена на рисунке 4. Линии уровня целевой функции сильно вытянуты, так что последовательные шаги метода ПГС постоянно оказываются то по одну, то по другую сторону получившегося «оврага», а процесс поиска минимума сильно замедляется.

В таком случае говорят об *овражности* или *плохой обусловленности* целевой функции. Не случайной будет аналогия с понятием плохо обусловленной матрицы. Связь терминов раскрывается в следующем определении, которое вводит численную характеристику овражности функции.

ОПРЕДЕЛЕНИЕ 1.21. *Степень овражности* выпуклой функции $f(x)$ есть отношение

$$\eta = \frac{\lambda_{\max}}{\lambda_{\min}}$$

где $\lambda_{\max}, \lambda_{\min}$ — максимальное и минимальное собственное значение матрицы Гессе этой функции, соответственно.

Овражная функция имеет $\eta \gg 1$.

Степень овражности, грубо говоря, характеризует разброс собственных значений матрицы Гессе.

Как будет показано в разделе 3.1.3, спектр гессиана целевой функции имеет весьма большое значение при анализе областей применимости различных методов оптимизации.

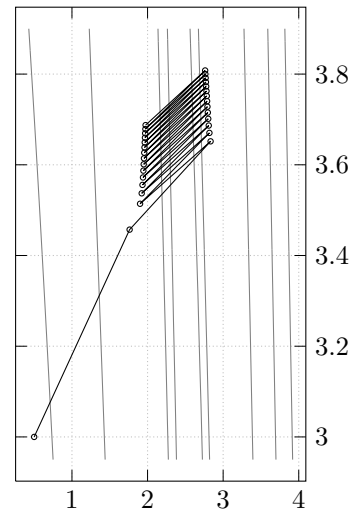


Рис. 4: Зацикливание ПГС на овражной функции

⁴ПГС более подробно рассмотрен далее, см. раздел 3.1.4.

⁵В качестве целевой здесь взята так называемая функция Химмельблау, которая рассматривается в разделе 3.3.2.

Более строгое определение понятия овражности дано в [3].

1.3.2. Явление многоэкстремальности

Функции, имеющие несколько локальных минимумов, называют *многоэкстремальными*. Такова, например, функция с рисунка 3, обладающая по крайней мере четырьмя минимумами. При этом функция может иметь не просто несколько, но бесконечно много локальных минимумов. Многокритериальность целевой функции осложняет решение задач, требующих точного определения *глобального* минимума.

Простейшая стратегия борьбы с многокритериальностью заключается в проведении в многократном процесса минимизации несколько раз при варьировании начального приближения. Если при этом получаются разные ответы, выбирают тот, в котором достигается минимальное значение целевой функции. Такой подход, вообще говоря, не гарантирует определение глобального минимума в отсутствие дополнительных соображений о выпуклости функции.

2. Симплекс-метод

Симплекс-метод позволяет решать задачи линейного программирования, заключающиеся в минимизации целевого линейного функционала при заданных линейных ограничениях. Основной идеей симплекс-метода является перебор вершин выпуклого многогранника в многомерном пространстве.

2.1. Алгоритм симплекс-метода

Приведём описание общей схемы метода. Более подробное изложение предложено, например, в [2].

Рассмотрим следующую задачу линейного программирования:

$$\begin{cases} f(x) = \sum_{i=1}^n c_i x_i + c_{n+1} \rightarrow \min \\ Ax = b, \quad A = (a_{ij}), \text{rg } A = m, b \in \mathbb{E}^n \\ x_i \geq 0, i = \overline{1, n} \end{cases} \quad (2.1)$$

Как видим, она состоит в поиске таких неотрицательных значений x_i , что достигается минимум линейной функции $f(x)$ при выполнении n линейных ограничений на набор x_i , которые задают *допустимую область решений* задачи.

ЗАМЕЧАНИЕ 2.1. Рассмотрение задачи с ограничениями лишь типа равенств не ограничивает общности рассуждений. Действительно, ограничение типа неравенства со знаком “ \leq ”

$$\sum_{i=1}^n a_{ij} x_i \leq b_j$$

может быть сведено к ограничению типа равенства путём добавления дополнительной неотрицательной переменной s_k , называемой *остаточной*:

$$\sum_{i=1}^n a_{ij} x_i + s_k = b_j$$

В случае знака “ \geq ” ограничение типа неравенства

$$\sum_{i=1}^n a_{ij} x_i \geq b_j$$

приводится к равенству путем вычитания *избыточной переменной*:

$$\sum_{i=1}^n a_{ij} x_i - s_k = b_j$$

В обоих случаях условие $s_k \geq 0$ добавляется к набору ограничений исходной задачи.

ЗАМЕЧАНИЕ 2.2. В силу теоремы 1.14 и замечаний 1.8, 2.1, множество ограничений задачи (2.1) является выпуклым. Известно, что локальный минимум (2.1) расположен хотя бы в одной из «вершин» допустимой области и может быть найден путем целенаправленного перебора⁶ *конечного* их числа.

⁶Геометрическая интерпретация простой задачи линейного программирования представлена на рис. 5 в разделе 2.1.2.

Пусть исходная задача может быть приведена к следующему виду:

$$\begin{cases} f(x) = -\sum_{i=m+1}^n \Delta_i x_i + \hat{c}_{n+1} \rightarrow \min \\ x_j + \sum_{i=m+1}^n \alpha_{ji} x_i = \beta_j, \beta_j \geq 0, j = \overline{1, m}, \\ x_i \geq 0, i = \overline{1, n} \end{cases} \quad (2.2)$$

ОПРЕДЕЛЕНИЕ 2.3. В задаче (2.2) переменные x_1, \dots, x_m , которым соответствуют единичные столбцы в новой матрице коэффициентов ограничений, называются *базисными*, а x_{m+1}, \dots, x_n — *свободными*.

ОПРЕДЕЛЕНИЕ 2.4. Решение задачи (2.2) при нулевых свободных переменных называется *базисным решением*. Если при этом все переменные принимают неотрицательные значения, решение называется *допустимым*, а в противном случае — *недопустимым*.

Итерационный алгоритм симплекс-метода для поиска оптимального решения перебирает ограниченное число допустимых базисных решений. В ходе работы алгоритма на каждой итерации определённая переменная исключается из состава базисных, а другая, наоборот, вводится в их число, что соответствует уменьшению значения целевой функции с помощью изменения значения одной из свободных переменных.

Для формулировки алгоритма удобно на каждой итерации использовать следующее матричное представление задачи:

$$\begin{bmatrix} & \bar{x}_1 & \bar{x}_2 & \dots & \bar{x}_m & \bar{x}_{m+1} & \dots & \bar{x}_n & \\ \bar{x}_1 & 1 & 0 & \dots & 0 & \bar{\alpha}_{1,m+1} & \dots & \bar{\alpha}_{1,n} & \bar{\beta}_1 \\ \bar{x}_2 & 0 & 1 & & \vdots & \bar{\alpha}_{2,m+1} & \dots & \bar{\alpha}_{2,n} & \bar{\beta}_2 \\ \vdots & \vdots & & \ddots & 0 & \vdots & \ddots & \vdots & \vdots \\ \bar{x}_m & 0 & \dots & 0 & 1 & \bar{\alpha}_{m,m+1} & \dots & \bar{\alpha}_{m,n} & \bar{\beta}_m \\ f & 0 & \dots & 0 & 0 & \bar{\Delta}_{m+1} & \dots & \bar{\Delta}_n & \hat{c}_{n+1} \end{bmatrix} \quad (2.3)$$

ЗАМЕЧАНИЕ 2.5. Поскольку наборы базисных и свободных переменных меняются с каждой итерацией, в данной формулировке с помощью $\bar{x}_1, \dots, \bar{x}_m$ и $\bar{x}_{m+1}, \dots, \bar{x}_n$ обозначены базисные и свободные переменные *на текущей итерации*. Аналогичные обозначения используются и для элементов матрицы (2.3). На практике подобное переупорядочивание переменных обычно не используется.

Приведём последовательность действий, выполняемых для решения задачи симплекс-методом.

- Шаг 1. Если все коэффициенты $\bar{\Delta}_{m+1}, \dots, \bar{\Delta}_n$ неположительны, то найдено оптимальное решение $\bar{x}_1 = \bar{\beta}_1, \dots, \bar{x}_m = \bar{\beta}_m, \bar{x}_{m+1} = \bar{x}_n = 0$. Если найдётся $\bar{\Delta}_s > 0$ такое, что в соответствующем столбце матрицы (2.3) все коэффициенты $\bar{\alpha}_{js} \leq 0$, то решений нет.
- Шаг 2. В качестве вводимой в состав базиса свободной переменной \bar{x}_s выбирается та, которой отвечает наибольший положительный коэффициент $\bar{\Delta}_s$. Соответствующий столбец называется *ведущим*.
- Шаг 3. Выбор исключаемой из базиса переменной \bar{x}_r производится таким образом, что достигается минимум симплексного отношения $\Theta = \frac{\bar{\beta}_r}{\bar{\alpha}_{rs}}$ для $\bar{\alpha}_{rs} > 0$. Соответствующая строка называется *ведущей*.

Шаг 4. Осуществляется переход к новому базису — ведущая строка делится на $\bar{\alpha}_{rs}$ и ведущий столбец путём элементарных преобразований приводится единичному виду.

Шаг 5. Процесс повторяется с первого шага для нового базиса

ЗАМЕЧАНИЕ 2.6. После того, как переменная была исключена из числа базисных, дальнейшее её рассмотрение в ходе работы алгоритма не имеет практического значения. В представленных далее примерах столбцы матрицы (2.3), соответствующие исключённым ранее переменным, будут опускаться.

Алгоритм симплекс-метода гарантирует допустимость базисного решения, получаемого на каждой итерации. При этом проблему может составлять построение *начального* допустимого базисного решения. Структура целевой функции и ограничений могут быть таковы, что сразу привести задачу (2.1) к виду (2.2) не удастся. Для этого пользуются различными методами для поиска начального решения, один из которых — метод искусственных переменных — изложен в следующем разделе.

ЗАМЕЧАНИЕ 2.7. Задачи линейного программирования, в которых все ограничения имеют вид неравенств с “ \leq ” с неотрицательной правой частью, приводятся к виду (2.2) за счёт добавления остаточных переменных (см. замечание 2.1), поэтому для них начальное допустимое решение получается сразу.

2.1.1. Метод искусственных переменных

В данном методе сначала рассматривается вспомогательная задача, решение которой затем используется в качестве начального для исходной задачи. Алгоритм метода имеет следующую схему:

Шаг 1. Пусть все ограничения исходной задачи (2.1) приведены к виду равенств. Для каждого ограничения, в котором не содержится остаточной переменной (см. замечание 2.1), вводится искусственная остаточная переменная.

Шаг 2. Решается вспомогательная задача линейного программирования *минимизации* суммы \tilde{f} введённых искусственных переменных при полученных ограничениях. Если минимальное значение новой целевой функции \tilde{f} больше нуля, то исходная задача решения не имеет. В противном случае полученное оптимальное решение используется в качестве начального допустимого базисного решения в исходной задаче.

2.1.2. Геометрическая интерпретация

Рассмотрим следующую простую задачу линейного программирования:

$$\begin{cases} f(x) = -x_1 - 2x_2 \rightarrow \min \\ -3x + 4y \leq 8 \\ 7x + 6y \leq 35 \\ x_1, x_2 \geq 0 \end{cases} \quad (2.4)$$

Проиллюстрируем на ней алгоритм симплекс-метода.

Согласно замечанию 2.1 преобразуем ограничения к равенствам, добавив остаточные переменные, и приведём задачу к виду (2.2):

$$\begin{cases} f(x) = -x_1 - 2x_2 \rightarrow \min \\ -3x + 4y + s_1 = 8 \\ 7x + 6y + s_2 = 35 \\ x_1, x_2, s_1, s_2 \geq 0 \end{cases}$$

Как и говорилось в замечании 2.7, в данном случае использование метода искусственных переменных не нужно, так как начальное допустимое решение находится сразу.

Перейдём к матричному представлению задачи.

$$\begin{bmatrix} & s_1 & s_2 & x_1 & x_2 & \\ s_1 & 1 & 0 & -3 & 4 & \mathbf{8} \\ s_2 & 0 & 1 & 7 & 6 & 35 \\ f & 0 & 0 & 1 & \mathbf{2} & 0 \end{bmatrix}$$

Максимальный положительный элемент в последней строке соответствует свободной переменной x_2 , поэтому её выберем в качестве вводимой. Минимум соотношения $\Theta = \frac{\beta_r}{\alpha_{rs}}$ достигается на первой строке (в ней $\Theta = \frac{8}{4} = 2$), поэтому первая базисная переменная s_1 выбирается в качестве исключаемой. Теперь путём элементарных преобразований приведём ведущий столбец к единичному виду:

$$\begin{bmatrix} & s_1 & s_2 & x_1 & x_2 & \\ s_1 & 1 & 0 & -3 & \mathbf{4} & 8 \\ s_2 & 0 & 1 & 7 & 6 & 35 \\ f & 0 & 0 & 1 & 2 & 0 \end{bmatrix} \begin{array}{l} \left[\begin{array}{c} \leftarrow -\frac{3}{2} \\ \leftarrow + \end{array} \right]^{-\frac{1}{2}} \mid \cdot \frac{1}{4} \\ \leftarrow + \end{array}$$

Таким образом переменная x_2 вводится в базис, а s_1 — исключается из него. На следующей итерации матрица задачи имеет вид:

$$\begin{bmatrix} & s_2 & x_1 & x_2 & \\ x_2 & 0 & -\frac{3}{4} & 1 & 2 \\ s_2 & 1 & \frac{23}{2} & 0 & \mathbf{23} \\ f & 0 & \frac{5}{2} & 0 & -4 \end{bmatrix}$$

На этот раз ведущим является второй столбец, соответствующий свободной переменной x_1 . Строка базисной переменной s_1 является ведущей. Введём выбранную переменную x_1 в состав базисных путём элементарных преобразований:

$$\begin{bmatrix} & s_2 & x_1 & x_2 & \\ x_2 & 0 & -\frac{3}{4} & 1 & 2 \\ s_2 & 1 & \frac{23}{2} & 0 & 23 \\ f & 0 & \frac{5}{2} & 0 & -4 \end{bmatrix} \begin{array}{l} \left[\begin{array}{c} \leftarrow + \\ \leftarrow \frac{3}{46} \end{array} \right]^{-\frac{5}{23}} \mid \cdot \frac{2}{23} \\ \leftarrow + \end{array}$$

В результате элементы рассматриваемой матрицы приобретут следующие значения:

$$\begin{bmatrix} & x_1 & x_2 & \\ x_2 & 0 & 1 & \frac{7}{2} \\ x_1 & 1 & 0 & 2 \\ f & 0 & 0 & -9 \end{bmatrix}$$

Отрицательность всех коэффициентов в последней строке свидетельствует о том, что найдено оптимальное допустимое решение. Соответствующие значения базисных переменных находятся в правом столбце матрицы: $x_1 = 2, x_2 = \frac{7}{2}$, свободные переменные приравняются к нулю.

Рассмотрим выполненные действия с геометрической точки зрения. На рисунке 5 изображены линии уровня функции $f(x) = -x_1 - 2x_2$, в минимизации которой на заданном множестве (—) и состоит задача (2.4).

На каждой итерации алгоритм симплекс-метода изменяет значение только одной небазисной переменной, не нарушая при этом ограничений задачи. В начальном базисном решении $s_1 = s_2 = 0$, откуда $x_1 = x_2 = 0$, так что движение начинается из точки начала координат.

После первой итерации переменная s_1 выводится из состава базисных взамен x_2 . В найденном базисном решении $x_2 = 2, s_2 = 23$, откуда $x_1 = 0$, так что метод приходит в точку $(0, 2)$, являющуюся пересечением границ двух ограничений $-3x_1 + 4x_2 \leq 8$ и $x_1 \geq 0$.

В результате второй итерации исключается уже переменная s_2 , и метод приходит в очередную вершину допустимого множества — $(2, 3.5)$, которая оказывается оптимальным решением задачи. Действительно, судя по ориентации линии уровня 9, проходящей через найденную точку, и направлению убывания $-f'(x) = (\frac{1}{2})$ (—►) целевой функции ясно, что дальнейшее уменьшение значения функции при наложенных ограничениях невозможно.

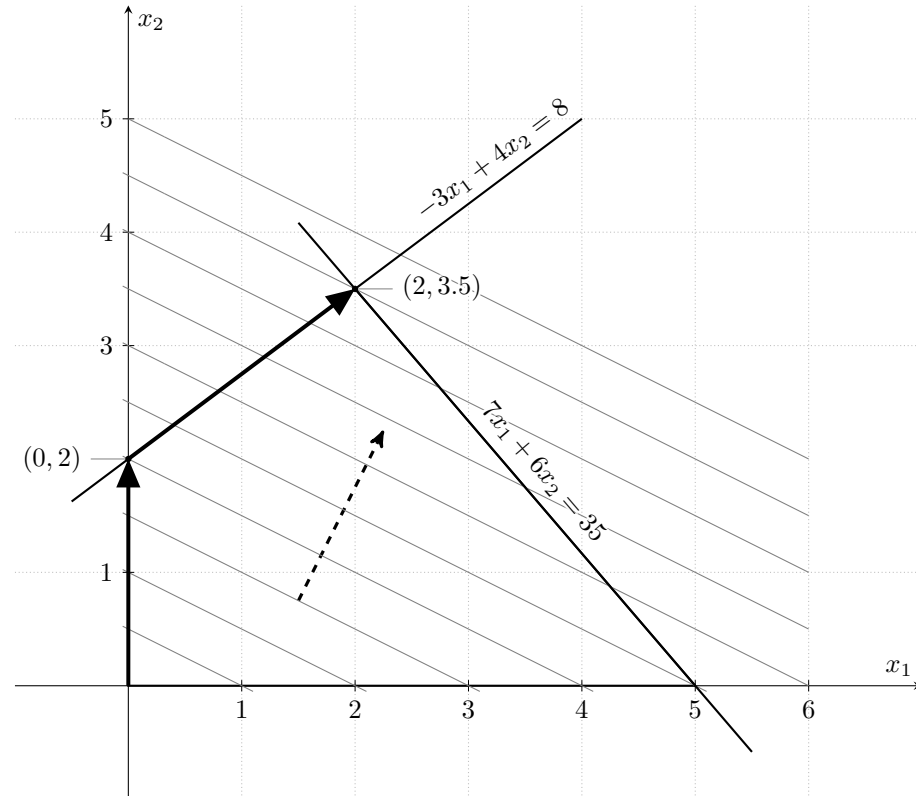


Рис. 5: Ход решения задачи (2.4) симплекс-методом

2.2. Практический пример

К решению предлагается следующая задача:

$$\begin{cases} f(x) = -3x_1 + 2x_2 - 2x_3 + 2x_4 - x_5 \rightarrow \min \\ -x_1 + x_2 - x_3 = 1 \\ -x_2 + x_3 + x_4 = 1 \\ x_2 + x_3 + x_5 = 2 \\ x_i \geq 0, i = \overline{1, 5} \end{cases} \quad (2.5)$$

Воспользуемся методом искусственных переменных для построения начального допустимого базисного решения. Добавим в первое ограничение переменную x_6 так, что оно примет вид

$$-x_1 + x_2 - x_3 + x_6 = 1$$

Заменим минимизируемую функцию на $\tilde{f}(x) = x_6$ и выразим её через свободные переменные x_1, x_2, x_3 , получив $\tilde{f}(x) = x_1 - x_2 + x_3 + 1$. Перейдём к рассмотрению вспомогательной задачи

$$\begin{cases} \tilde{f}(x) = x_1 - x_2 + x_3 + 1 \rightarrow \min \\ -x_1 + x_2 - x_3 + x_6 = 1 \\ -x_2 + x_3 + x_4 = 1 \\ x_2 + x_3 + x_5 = 2 \\ x_i \geq 0, i = \overline{1, 6} \end{cases} \quad (2.6)$$

Решим её симплекс-методом. Составим расширенную матрицу из коэффициентов ограничений и целевой функции:

$$\begin{bmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & \\ x_6 & -1 & 1 & -1 & 0 & 0 & 1 & \mathbf{1} \\ x_4 & 0 & -1 & 1 & 1 & 0 & 0 & 1 \\ x_5 & 0 & 1 & 1 & 0 & 1 & 0 & 2 \\ \tilde{f} & -1 & \mathbf{1} & -1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

Базисными переменными являются x_4, x_5, x_6 . Для дальнейшего решения необходимо исключить искусственную переменную x_6 из состава базисных.

Выберем в качестве ведущего столбец с максимальным элементом в последней строке — это столбец переменной x_2 , поскольку $\max\{1, -1, 0\} = 1$. С помощью симплексного отношения выберем первую строку в качестве ведущей.

Учитывая выбранный ведущий элемент, переведём переменную x_2 в состав базисных взамен x_6 :

$$\begin{bmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & \\ x_6 & -1 & \mathbf{1} & -1 & 0 & 0 & 1 & 1 \\ x_4 & 0 & -1 & 1 & 1 & 0 & 0 & 1 \\ x_5 & 0 & 1 & 1 & 0 & 1 & 0 & 2 \\ \tilde{f} & 1 & 1 & -1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{array}{l} \left[\begin{array}{c} \leftarrow + \\ \leftarrow + \\ \leftarrow + \end{array} \right]^{-1} \\ \leftarrow + \end{array}$$

Получим матрицу⁷ с базисными переменными x_2, x_4, x_5 :

$$\begin{bmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 & \\ x_2 & -1 & 1 & -1 & 0 & 0 & 1 \\ x_4 & -1 & 0 & 0 & 1 & 0 & 2 \\ x_5 & 1 & 0 & 2 & 0 & 1 & 1 \\ \tilde{f} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.8)$$

⁷Столбец исключённой переменной опущен, см. замечание 2.6.

При этом значение функции $\tilde{f}(x)$ равно нулю, что свидетельствует о том, что найденное решение вспомогательной задачи является допустимым для исходной.

Отбросим искусственную переменную x_6 и вернёмся к начальной задаче (2.5). Исключим из исходной целевой функции $f(x)$ базисные переменные x_2, x_4 и x_5 , используя ограничения из матрицы (2.8):

$$\begin{cases} -x_1 + x_2 - x_3 = 1 \\ -x_1 + x_4 = 2 \\ x_1 + 2x_3 + x_5 = 1 \end{cases}$$

Тогда $f(x)$ принимает вид

$$f(x) = 2x_1 + 2x_3 + 5$$

Используем эту целевую функцию в матрице ограничений (2.8):

$$\begin{bmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 & \\ x_2 & -1 & 1 & -1 & 0 & 0 & 1 \\ x_4 & -1 & 0 & 0 & 1 & 0 & 2 \\ x_5 & 1 & 0 & 2 & 0 & 1 & 1 \\ f & -2 & 0 & -2 & 0 & 0 & 5 \end{bmatrix}$$

В нижнем ряду под столбцами переменных все коэффициенты отрицательны, что свидетельствует о том, что найдено оптимальное решение:

$$x = (0, 1, 0, 2, 0)$$

3. Безусловная оптимизация

В этом разделе представлена общая схема градиентных методов, рассмотрено понятие функции релаксации и описан метод многопараметрической оптимизации с чебышёвскими функциями релаксации RELCH, предложенный И. Г. ЧЕРНОРУЦКИМ в [3].

3.1. Теоретические сведения

3.1.1. Постановка задачи

Настоящий раздел посвящён решению следующей задачи *безусловной* минимизации:

$$f(x) \rightarrow \min, \quad x \in \mathbb{R}^n, \quad f(x) \in C^2(\mathbb{R}^n) \quad (3.1)$$

Рассматриваемые итерационных методов построены по общей схеме, выражаемой следующей рабочей формулой, которая определяет способ перехода к новому приближению x^{k+1} точки минимума на очередной итерации:

$$x^{k+1} = \varphi_k(x^k) \quad (3.2)$$

При этом используются следующие критерии останова процесса (3.2) на k -м шаге:

- 1) Близость к нулю⁸ нормы градиента функции:

$$\|f'(x^k)\| < \varepsilon$$

- 2) Близость соседних приближений:

$$\|x^{k+1} - x^k\| < \varepsilon$$

- 3) Близость значений целевой функции $f(x)$ в точках соседних приближений:

$$\|f(x^{k+1}) - f(x^k)\| < \varepsilon$$

- 4) Остановка процесса после выполнения предельного количества итераций:

$$k = k_{\max}$$

3.1.2. Градиентные методы

ОПРЕДЕЛЕНИЕ 3.1. *Градиентными* называются итерационные методы оптимизации со следующей рабочей формулой:

$$x^{k+1} = x^k - H_k(G_k, h_k) g_k \quad (3.3)$$

здесь H_k — некоторая функция от матрицы Гессе $G_k = G(x^k) = f''(x^k)$ и параметра h_k , а $g_k = g(x^k) = f'(x^k)$ — градиент функции в точке x_k .

При постоянных значениях параметров на различных шагах итерации соответствующие индексы k в формуле (3.3) опускаются.

Предполагается, что в некоторой ε_k -окрестности $\{\|x - x^k\| < \varepsilon_k\}$ точки x^k функция $f(x)$ аппроксимируется гиперboloидом:

$$f(x) \approx \frac{1}{2} \langle G_k x, x \rangle - \langle a_k, x \rangle + b_k \approx \frac{1}{2} \langle G_k x, x \rangle \quad (3.4)$$

⁸Здесь и в дальнейшем, $\varepsilon > 0$ — некоторое наперёд заданное малое число.

Ставится задача построения таких матричных функций H_k , при которых выполняется *условие релаксации* процесса

$$f(x^{k+1}) < f(x^k) \quad (3.5)$$

При этом требуется, чтобы величина нормы $\|x^{k+1} - x^k\|$ была ограничена сверху лишь параметром ε_k , который характеризует область справедливости локальной квадратичной модели (3.4).

3.1.3. Функция релаксации

ОПРЕДЕЛЕНИЕ 3.2. *Функцией релаксации* называется скалярная функция

$$R_h(\lambda) = 1 - H(\lambda, h)\lambda, \quad \lambda, h \in \mathbb{R} \quad (3.6)$$

где $H(\lambda, h)$ — скалярный аналог матричной функции $H(G, h)$ из формулы (3.3).

В дальнейшем индекс h у функции релаксации $R_h(\lambda)$ иногда будем опускать.

ЗАМЕЧАНИЕ 3.3. Функция релаксации с необходимостью является аналитической.

ОПРЕДЕЛЕНИЕ 3.4. *Множителями релаксации* для точки x^k называются значения функции релаксации на спектре матрицы Гессе:

$$R_h(\lambda_i), \lambda_i \in \text{Sp } G_k \quad (3.7)$$

Благодаря следующей теореме, функция релаксации используется для анализа различных градиентных методов.

ТЕОРЕМА 3.5. *Для выполнения условия релаксации (3.5) при любых x^k необходимо и достаточно, чтобы*

$$\begin{aligned} |R(\lambda_i)| &\geq 1 & \lambda_i < 0 \\ |R(\lambda_i)| &\leq 1 & \lambda_i > 0 \\ i &= \overline{1, n} \end{aligned} \quad (3.8)$$

Скорость релаксации может быть оценена с использованием следующего соотношения:

$$2|f(x^{k+1}) - f(x^k)| = \sum_{\lambda_i^+ > 0} \{\xi_{i,k}^2 \lambda_i^+ [1 - R^2(\lambda_i^+)]\} + \sum_{\lambda_i^- < 0} \{\xi_{i,k}^2 |\lambda_i^-| [R^2(\lambda_i^-) - 1]\} \quad (3.9)$$

здесь λ_i^+ и λ_i^- — положительные и отрицательные собственные значения матрицы G_k . Коэффициенты $\xi_{i,k}$ происходят из разложения $x^k = \sum_i \xi_{i,k} u^i$

по собственным векторам u^i матрицы Гессе.

Таким образом, эффективными оказываются методы, функция релаксации которых в положительной области значений λ как можно *меньше* уклоняется от нуля, а при отрицательных λ становится как можно *больше* по модулю (см. рис. 6).

3.1.4. Метод пгс

В качестве примера рассмотрим⁹ метод *простого градиентного спуска* (ПГС), рабочая формула которого имеет вид:

$$x^{k+1} = x^k - hg_k \quad (3.10)$$

⁹Этот метод уже упоминался в разделе 1.3.1.

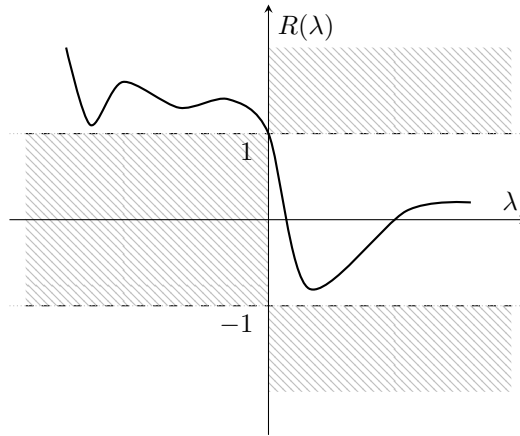


Рис. 6: Функция релаксации, удовлетворяющая условиям теоремы 3.5

где $h \in \mathbb{R}$ — некоторое фиксированное число (обычно малое), называемое *шагом*.

Рассмотрим функцию релаксации метода ПГС:

$$R(\lambda) = 1 - \lambda h \quad (3.11)$$

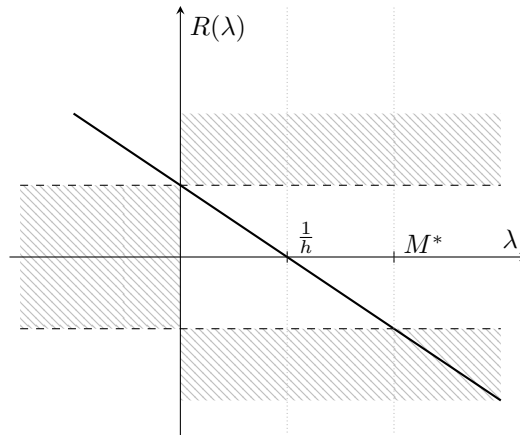


Рис. 7: Функция релаксации (3.11) метода простого градиентного спуска

Из её графика на рисунке 7 видно, что метод применим лишь в том случае, когда собственные значения матрицы Гессе оптимизируемой функции не превосходят некоторого критического значения $M^* = \frac{2}{h}$. Кроме того, в области значений λ , близких к нулю или M^* , согласно (3.9) скорость релаксации сильно снижается.

Так объясняется не самая широкая область применимости ПГС и проблемы градиентного спуска, обозначенные в разделе 1.3.1.

Имеются различные модификации метода ПГС. Одной из наиболее простых является метод ПГС₂ — простой градиентный спуск с *дроблением шага*. В ПГС₂ изначально h полагается достаточно большим, но в случае нарушения условия релаксации (3.5) на какой-либо итерации h делится пополам до тех пор, пока (3.5) вновь не станет выполняться.

3.2. Описание метода RELCH

3.2.1. Использование полиномов Чебышёва в функции релаксации

Рассмотрим смещённые полиномы Чебышёва второго рода, которые определяются согласно рекуррентному соотношению:

$$P_0(\lambda) = 0 \quad (3.12)$$

$$P_1(\lambda) = 1 \quad (3.13)$$

$$P_k(\lambda) = 2(1 - 2\lambda)P_{k-1}(\lambda) - P_{k-2}(\lambda), \quad k \geq 2 \quad (3.14)$$

Функциональная последовательность таких полиномов обладает важным свойством, а именно

$$\lim_{s \rightarrow \infty} \frac{P_s(\lambda)}{s} \rightarrow 0 \text{ на } (0; 1) \quad (3.15)$$

причём сходимость равномерная.

Предположим, что собственные числа матрицы Гессе целевой функции $f(x)$ задачи (3.1) в положительной области спектра не превосходят 1 (для этого достаточно в рассмотрении считать градиент и матрицу Гессе нормированными). Тогда использование следующей функции в качестве релаксационной:

$$R_s(\lambda) = \frac{P_s(\lambda)}{s} \quad (3.16)$$

позволяет обеспечить, согласно (3.9), сколь угодно быструю релаксацию. На иллюстрации 8 приведены графики функции $R_s(\lambda)$ вблизи $[0; 1]$ для нескольких значений s .

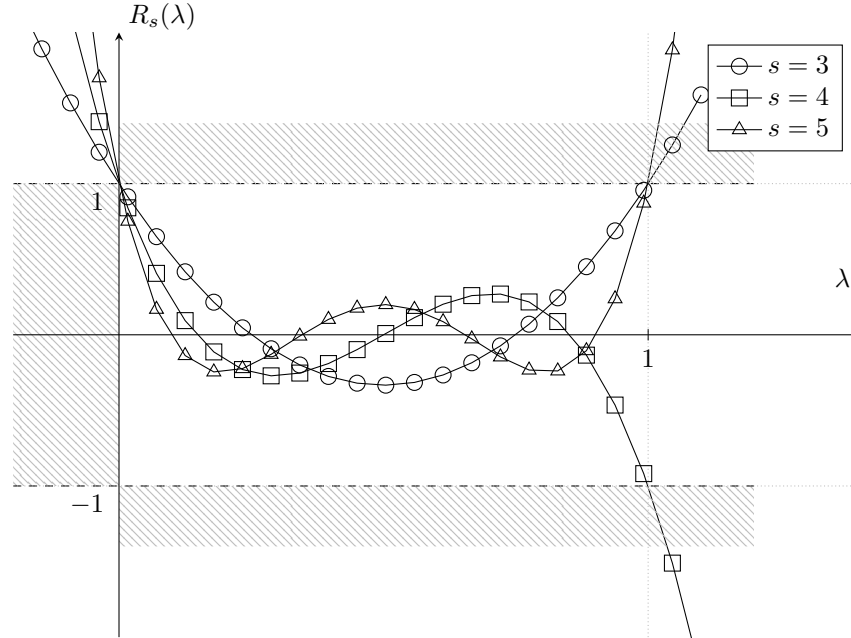


Рис. 8: Функции релаксации на основе полиномов Чебышёва

Уже при $s = 8$ значение $R_s(\lambda)$ не превосходит по модулю 0.23 на отрезке $[0.025; 0.975]$, обеспечивая хорошее подавление слагаемых (3.9), соответствующих большому диапазону положительных собственных чисел.

При этом стремление $R_s(\lambda)$ к $+\infty$ в отрицательной части спектра также соответствует требованиям к хорошей функции релаксации.

3.2.2. Реализация метода

Согласно (3.6), выбранной функции релаксации соответствует зависимость

$$H(\lambda) = \frac{1 - \frac{P_s(\lambda)}{s}}{\lambda}$$

откуда с учётом (3.12) получим следующие рекуррентные соотношения уже для функции $H(\lambda)$:

$$\begin{aligned} H_1 &= 0 \\ H_2 &= 2 \\ H_{s+1} \cdot (s+1) &= 2s \cdot (1 - 2\lambda) \cdot H_s - (s-1) \cdot H_{s-1} + 4s \end{aligned} \quad (3.17)$$

Параметр метода s равен степени используемых полиномов Чебышёва. Соображения по выбору s приведены ниже.

Подстановка полученного соотношения (3.17) в (3.3) даёт следующую рабочую формулу

$$\begin{aligned} x^{k+1} &= x^k - H_{s+1}g_k = \\ &= x^k - \frac{2s}{s+1}(E - 2G_k)H_s g_k + \frac{s-1}{s+1}H_{s-1}g_k - \frac{4s}{s+1}g_k \end{aligned} \quad (3.18)$$

Таким образом, вектор смещения $d_{s+1} = x^{k+1} - x^k$ при выбранном значении параметра s на каждом шаге k вычисляется по следующей рекуррентной формуле:

$$\begin{aligned} d_1 &= 0 \\ d_2 &= -2g_k \\ d_{s+1} &= \frac{2s}{s+1}(E - 2G_k)d_s - \frac{s-1}{s+1}d_{s-1} - \frac{4s}{s+1}g_k \end{aligned} \quad (3.19)$$

После вычисления d_{s+1} применяется регулировка шага путём деления его пополам:

$$d_{s+1} = \frac{d_{s+1}}{2} \quad (3.20)$$

причём регулировка (3.20) последовательно продолжается до тех пор, пока не будет обеспечено условие релаксации (3.5) процесса оптимизации. Подобное ограничение шага d_{s+1} используется в целях предотвращения выхода из области справедливости локальной квадратичной модели (3.4). На идее такой регулировки также построен метод ПГС₂, описанный ранее в разделе 3.1.4.

3.2.3. Выбор параметра метода

Для применения на практике рассматриваемый метод минимизации требует задания параметра s . Обратимся к способам выбора s .

Автором метода предлагается выбирать значение s из следующего соотношения:

$$s = 1.3\sqrt{\eta} \quad (3.21)$$

где η — оценка овражности минимизируемой функции. Таким образом, возникает необходимость определения η перед применением метода. Рассмотрим предлагаемые для этого способы.

ИСПОЛЬЗОВАНИЕ ГЕССИАНА

Можно определять степень овражности непосредственно из определения 1.21, исследуя аналитические выражения для собственных чисел гессиана функции или же вычисляя их численно. При алгоритмическом построении спектра матрицы Гессе и делении на λ_{\min} могут возникнуть проблемы численного переполнения, вызванные нехваткой машинной точности.

ПРЕДВАРИТЕЛЬНОЕ ПРИМЕНЕНИЕ ПРОСТОГО ГРАДИЕНТНОГО СПУСКА

Данный подход предусматривает применение метода простого градиентного спуска (см. раздел 3.1.4) для оценки η .

Для этого сначала запускается решение исходной задачи минимизации (3.1) с помощью ПГС. Когда метод простого градиентного спуска заикливается (см. 1.3.1), отношение норм градиентов целевой функции в точках соседних приближений стабилизируется около некоторого μ :

$$\frac{\|f'(x^{k+1})\|}{\|f'(x^k)\|} \approx \mu \quad (3.22)$$

Тогда оценка овражности может быть найдена из следующего соотношения:

$$\eta = \frac{2}{|1 - \mu|} \quad (3.23)$$

После этого из (3.21) определяется s , и метод RELCH начинает работу из последней найденной методом ПГС точки приближения.

Для точности оценки s необходимо в качестве шага h в рабочей формуле (3.10) метода ПГС выбирать как можно большее значение, при котором обеспечивается стабилизация отношения (3.22). При этом для определения h , удовлетворяющего такому свойству, могут потребоваться предварительные попытки решения задачи с различными пробными значениями h .

Отметим, что в случае функции малой овражности достаточно хорошее решение может получиться уже на этапе применения ПГС.

При алгоритмическом построении s нужно учитывать возможность получения столь большого значения параметра, что на вычисления будет затрачиваться непозволительно долгое время. Поэтому при начальном рассмотрении задачи может потребоваться искусственное ограничение параметра s сверху.

Двухэтапную модификацию метода RELCH в дальнейшем будем обозначать как GDRELCH.

ЗАМЕЧАНИЕ 3.6. При увеличении параметра s на некоторое ограниченное значение рост скорости релаксации, вообще говоря, не гарантируется.

В разделе 3.3 представлены результаты применения GDRELCH на тестовых функциях.

3.2.4. Трудности, возникающие при реализации RELCH

В ходе тестирования RELCH были выявлены случаи, когда метод попадает в такую точку, что норма градиента и значение гессиана функции приводят к численному переполнению в результате многократных вычислений по формуле (3.19), так что условие релаксации нарушается, несмотря на регулировку (3.20).

Для борьбы с таким поведением использовалась следующая *эвристическая* техника: после вычисления (3.19) при обнаружении бесконечного

значения $\|d_{s+1}\|$ следующее приближение выбиралось в случайном направлении на расстоянии \sqrt{n} от текущего. В общем случае невозможно доказать результативность таких действий, однако подход продемонстрировал успех на ряде тестовых задач.

При выборе начального приближения вдали от точки минимума многократные вычисления по формуле (3.19) также могут привести к численному переполнению и из-за большого по модулю значения целевой функции. В определённых случаях проблемы можно избежать, используя на первом этапе метод ПГС или ПГС₂ для начального снижения значения целевого функционала.

3.3. Тестовые задачи

В данном разделе рассмотрено несколько тестовых функций с их анализом на выпуклость и овражность, а также приведены результаты применения метода RELCH для решения задач их минимизации.

Для анализа выпуклости функций использовалась теорема 1.10, применимая в силу того, что все рассматриваемые функции дважды непрерывно-дифференцируемы.

Овражность оценивалась с помощью определения 1.21, эвристически и на основе информации о применимости метода ПГС к целевой функции (см. разделы 3.1.4, 1.3.1).

Все численные значения приведены с точностью до двух значащих цифр.

3.3.1. Функция Розенброка

Одним из классических тестов для различных алгоритмов оптимизации является тест Розенброка, заключающийся в минимизации следующей функции:

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2 \quad (\rho\text{-}3.23)$$

Точку $(-1.2, 1)$ предлагается взять в качестве начального приближения.

Выпуклость

Матрица Гессе этой функции имеет вид:

$$\begin{pmatrix} 1200x^2 - 400y + 2 & -400x \\ -400x & 200 \end{pmatrix} \quad (3.24)$$

Применим критерий Сильвестра к матрице (3.24) для исследования её на определённости. Запишем условия положительности угловых миноров:

$$\begin{aligned} \Delta_1 &= 1200x^2 - 400y + 2 > 0 \iff y < 3x^2 + \frac{1}{200} \\ \Delta_2 &= 80\,000x^2 - 80\,000y + 400 > 0 \iff y < x^2 + \frac{1}{200} \end{aligned}$$

При этом второе ограничение является более сильным. Значит, граница области выпуклости функции Розенброка проходит вдоль кривой

$$y = x^2 + \frac{1}{200} \quad (3.25)$$

Нижне этой кривой функция Розенброка строго выпукла. Согласно теореме 1.19, она имеет в своей области выпуклости единственную точку минимума.

При значениях (x, y) , лежащих выше кривой (3.25), функция Розенброка вогнута, а потому в силу теоремы 1.20 не может иметь внутри этой области точек минимума.

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ

Найдём минимум функции ($\rho\text{-}3.23$) аналитически.

Из необходимых условий стационарности точки имеем:

$$\begin{aligned} \frac{\partial f}{\partial x} &= 400x^3 - 400yx + 2x - 2 = 0 \\ \frac{\partial f}{\partial y} &= 200y - 200x^2 = 0 \end{aligned}$$

откуда $x = y = 1$. Найденная точка $A(1, 1)$ лежит в области выпуклости, а потому является глобальным минимумом функции Розенброка.

ОВРАЖНОСТЬ

В точке начального приближения, лежащей в области выпуклости, собственные значения матрицы Гессе функции Розенброка приблизительно равны 23.63 и 1506.37, откуда согласно (1.21) получается степень овражности

$$\eta = \frac{1506.37}{23.63} \approx 63.7 \quad (3.26)$$

Вычисление значения η в A даёт

$$\eta = \frac{1001.60}{0.4} \approx 2507.76 \quad (3.27)$$

Исследуемая функция имеет кривую перегиба $y = x^2 + \frac{1}{200}$, разделяющую её области выпуклости и вогнутости. Вдоль этой кривой степень овражности функции $\eta = \infty$.

На иллюстрации 9 представлены линии уровня функции Розенброка, которые сильно вытягиваются вдоль всей кривой $y = x^2$ за счёт близости перегиба и быстрого роста функции в выпуклой области.

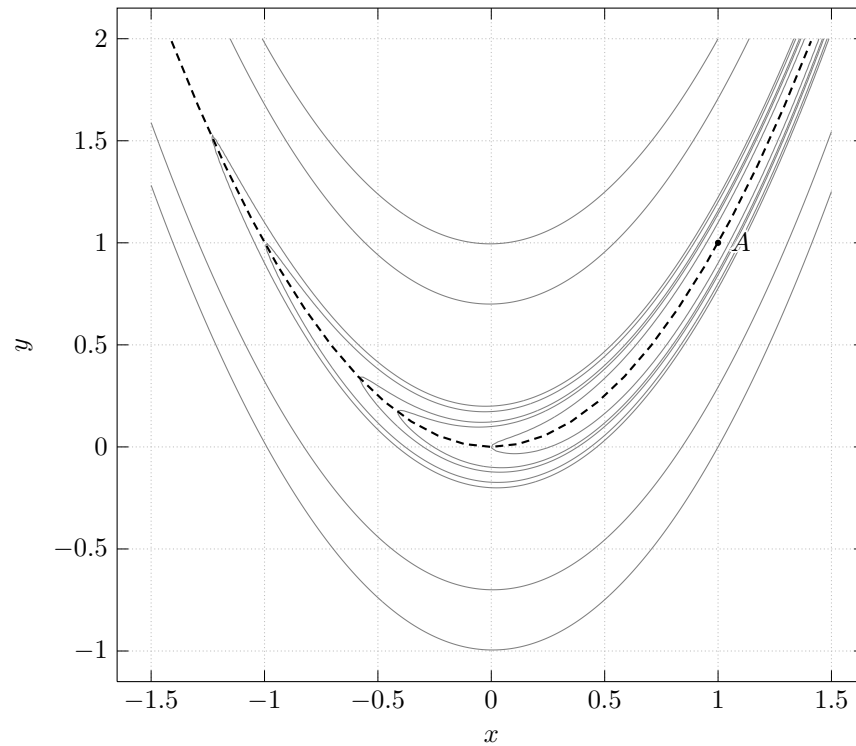


Рис. 9: Линии уровней 1, ... 100 функции Розенброка (р-3.23), её кривая перегиба $y = x^2 + \frac{1}{200}$ и глобальный минимум в точке $A(1, 1)$

С учётом наблюдаемой картины, а также перепада в значениях степени овражности (3.26) и (3.27) можно утверждать, что в ходе работы алгоритму придётся бороться с сильной овражностью целевой функции.

МИНИМИЗАЦИЯ С ПОМОЩЬЮ RELCH ПРИ РУЧНОМ ВЫБОРЕ ПАРАМЕТРА

В качестве пробных оценок η для применения в формуле (3.21) выберем значения (3.26) и (3.27), тогда:

$$s_1 = 1.3\sqrt{64} \approx 10$$

$$s_2 = 1.3\sqrt{2507} \approx 66$$

Из эмпирических соображений в качестве s выбиралось ближайшее к $1.3\sqrt{\eta}$ чётное число (в ходе тестов алгоритм демонстрировал наилучшую сходимость именно при чётных s).

На иллюстрации 11 представлены результаты работы RELCH на функции Розенброка. Видно, что при малом значении $s = 10$ RELCH быстро попадает в окрестность минимума, но испытывает трудности при продвижении по дну оврага. С ростом s до 66 алгоритм крупными шагами стремительно движется вдоль оврага к точке минимума. Протестирована также работа алгоритма с $s = 200$. Высокая скорость сходимости с таким значением соответствует ожиданиям.

ПРИМЕНЕНИЕ GDRELCH

Описанный в разделе 3.2.3 двухэтапный метод GDRELCH применим к функции Розенброка. На иллюстрации 10 показано, как сначала метод ПГС закликивается в овраге, что соответствует стабилизации (3.22), после чего включается обычный RELCH со значением параметра, определённым на первом этапе. В данном случае стабилизацию (3.22) удалось обеспечить при шаге $h = 0.003$. В таблице 1, содержащей информацию о ходе вычисления, замечен участок в окрестности точки $(-0.48, 0.23)$, у которой происходит стабилизация нормы градиента целевой функции на уровне ≈ 2.13 . После этого по формуле (3.23) определяется s и включается алгоритм RELCH, который быстро сводит значение функции до уровня порядка 10^{-13} .

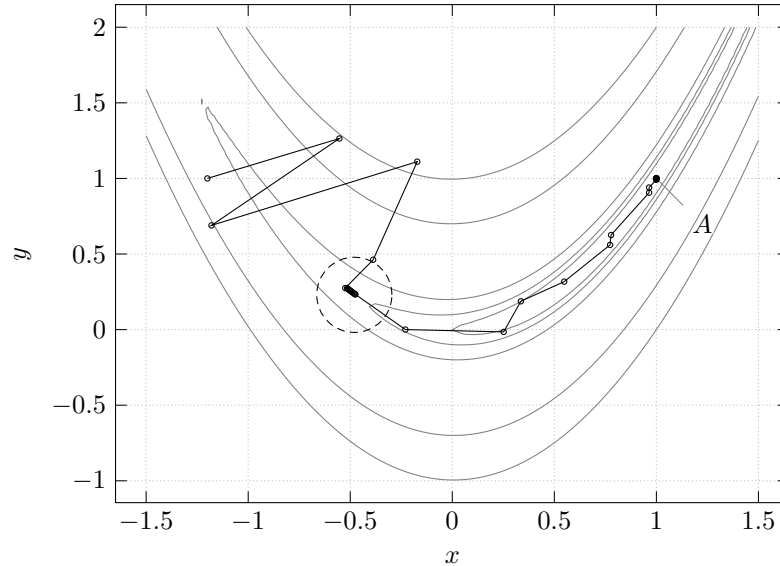


Рис. 10: Минимизация функции Розенброка алгоритмом GDRELCH при значении шага $h = 0.003$.

x_1	x_2	$f(x)$	$\ f'(x)\ $
-1.2	1	94.18	232.87
-0.55	1.26	54.14	283.44
-1.18	0.69	118.31	364.22
-0.17	1.11	11.65	227.93
-0.39	0.46	2.32	77.23
-0.52	0.28	2.31	3.05
-0.52	0.27	2.29	2.18
-0.51	0.27	2.28	2.12
-0.51	0.27	2.26	2.12
-0.5	0.26	2.25	2.12
-0.5	0.26	2.24	2.12
-0.49	0.25	2.22	2.13
-0.49	0.25	2.21	2.13
-0.48	0.24	2.2	2.13
-0.48	0.24	2.18	2.14
-0.48	0.23	2.17	2.14
-0.48	0.23	1.79	2.14
-0.23	$3.13 \cdot 10^{-4}$	1.18	12.78
0.25	$-1.51 \cdot 10^{-2}$	1	16.95
0.34	0.19	0.23	18.75
0.55	0.32	0.19	5.63
0.77	0.56	$9.14 \cdot 10^{-2}$	13.28
0.78	0.63	$5.89 \cdot 10^{-2}$	7.96
0.96	0.91	$1.17 \cdot 10^{-2}$	10.36
0.96	0.94	$6.63 \cdot 10^{-3}$	4.49
1	0.99	$4.14 \cdot 10^{-5}$	3.64
1	1	$2.45 \cdot 10^{-5}$	0.27
1	1	$9.82 \cdot 10^{-7}$	0.22
1	1	$6.64 \cdot 10^{-7}$	$2.8 \cdot 10^{-2}$
1	1	$4.23 \cdot 10^{-7}$	$3.31 \cdot 10^{-2}$
1	1	$2.03 \cdot 10^{-8}$	$2.8 \cdot 10^{-2}$
1	1	$1.37 \cdot 10^{-8}$	$2.59 \cdot 10^{-3}$
1	1	$3.75 \cdot 10^{-9}$	$5.05 \cdot 10^{-3}$
1	1	$9.02 \cdot 10^{-10}$	$2.6 \cdot 10^{-3}$
1	1	$2.9 \cdot 10^{-10}$	$1.23 \cdot 10^{-3}$
1	1	$5.96 \cdot 10^{-11}$	$6.85 \cdot 10^{-4}$
1	1	$2.85 \cdot 10^{-11}$	$2.7 \cdot 10^{-4}$
1	1	$3.99 \cdot 10^{-12}$	$2.07 \cdot 10^{-4}$
1	1	$3.86 \cdot 10^{-12}$	$3.14 \cdot 10^{-5}$

Таблица 1: Минимизация функции Розенброка алгоритмом GDRELCH при $h = 0.003$. Зацикливание ПГС происходит в окрестности точки $(-0.48, 0.23)$.

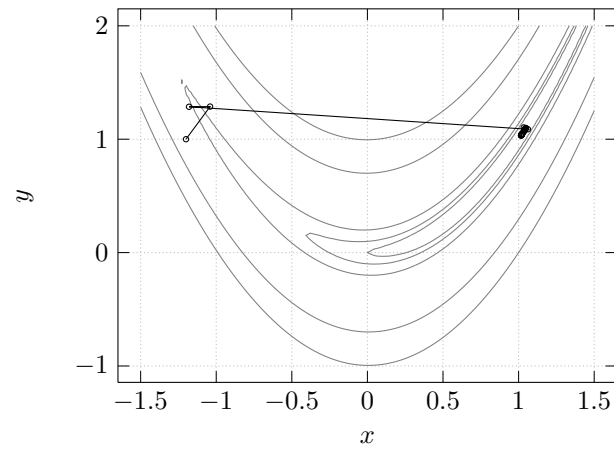
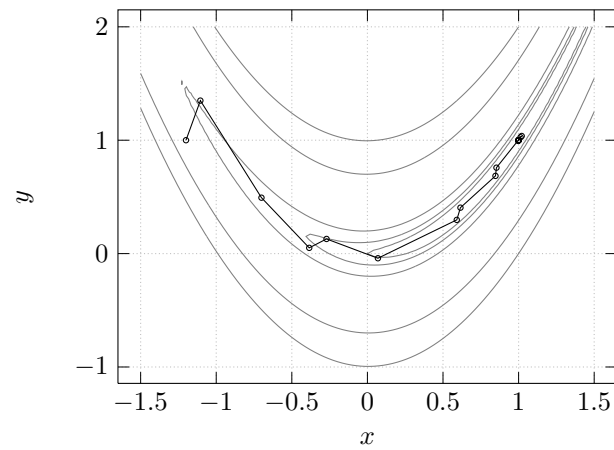
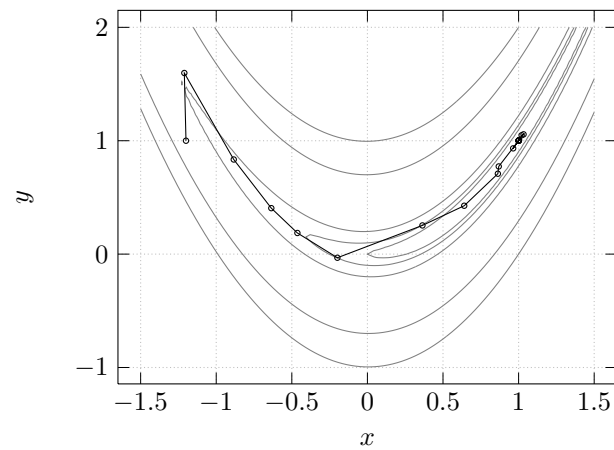
а) $s = 10$ б) $s = 66$ в) $s = 200$

Рис. 11: Минимизация функции Розенброка алгоритмом RELCH при различных значениях параметра s

3.3.2. Функция Химмельблау

Рассмотрим уже виденную ранее¹⁰ функцию Химмельблау, которая задаётся следующим образом:

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \quad (\chi\text{-}3.27)$$

ВЫПУКЛОСТЬ

Матрица Гессе функции Химмельблау имеет вид:

$$\begin{pmatrix} 4y + 12x^2 - 42 & 4y + 4x \\ 4y + 4x & 12y^2 + 4x - 26 \end{pmatrix} \quad (3.28)$$

Исследуем её на определённую с помощью критерия Сильвестра, записав условие положительности угловых миноров:

$$\Delta_1 = 4y + 12x^2 - 42 > 0 \iff y > 10\frac{1}{2} - 3x^2 \quad (3.29)$$

$$\Delta_2 = 4(12y^3 + (36x^2 - 130)y^2 - (4x + 26)y + 12x^3 - 82x^2 - 42x + 273) > 0 \quad (3.30)$$

Первое условие имеет простой вид, в то время как границей решения второго является кривая четвёртого порядка. Геометрический образ условий (3.29) вместе с линиями уровня функции Химмельблау представлен на рисунке 12.

Во всей области под гиперболой $10\frac{1}{2} - 3x^2$ (---) функция вогнута, несмотря на то, что на графическом изображении второго условия вокруг начала координат виден скруглённый прямоугольник, внутри которого выполняется условие $\Delta_2 > 0$. В этом отношении функция Химмельблау отличается от функции Розенброка, где условие $\Delta_1 > 0$ было безусловно слабее условия $\Delta_2 > 0$.

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ

Известно, что локальные минимумы¹¹ функции (χ-3.27) расположены в точках $A = (-3.78, -3.28)$, $B = (-2.80, 3.13)$, $C = (3.58, -1.85)$, $D = (3.00, 2.00)$, и в каждой из них достигается значение 0. Все они находятся в областях выпуклости функции, чего и следует ожидать.

ОВРАЖНОСТЬ

Как было продемонстрировано ещё в разделе 1.3.1, даже метод ПГС с фиксированным шагом может локализовать точку минимума функции Химмельблау. Исходя из этого, можно предположить, что высокой степенью овражности эта функция не обладает.

Действительно, даже в точках минимумов η не превосходит *четырёх*:

$$\begin{aligned} \eta(A) &\approx 1.9 & \eta(B) &\approx 3.18 \\ \eta(C) &\approx 3.65 & \eta(D) &\approx 3.2 \end{aligned} \quad (3.31)$$

С учётом данных соображений можно заключить, что предпосылок для выбора очень большого значения s при использовании RELCH для оптимизации функции нет. Необходимость применения метода GDRELCH в данном случае также ничем не обоснована.

¹⁰См. раздел 1.3.1

¹¹Численные значения координат A, B, C, D приведены в литературе, см. [6]. Их аналитические выражения имеют весьма сложный вид.

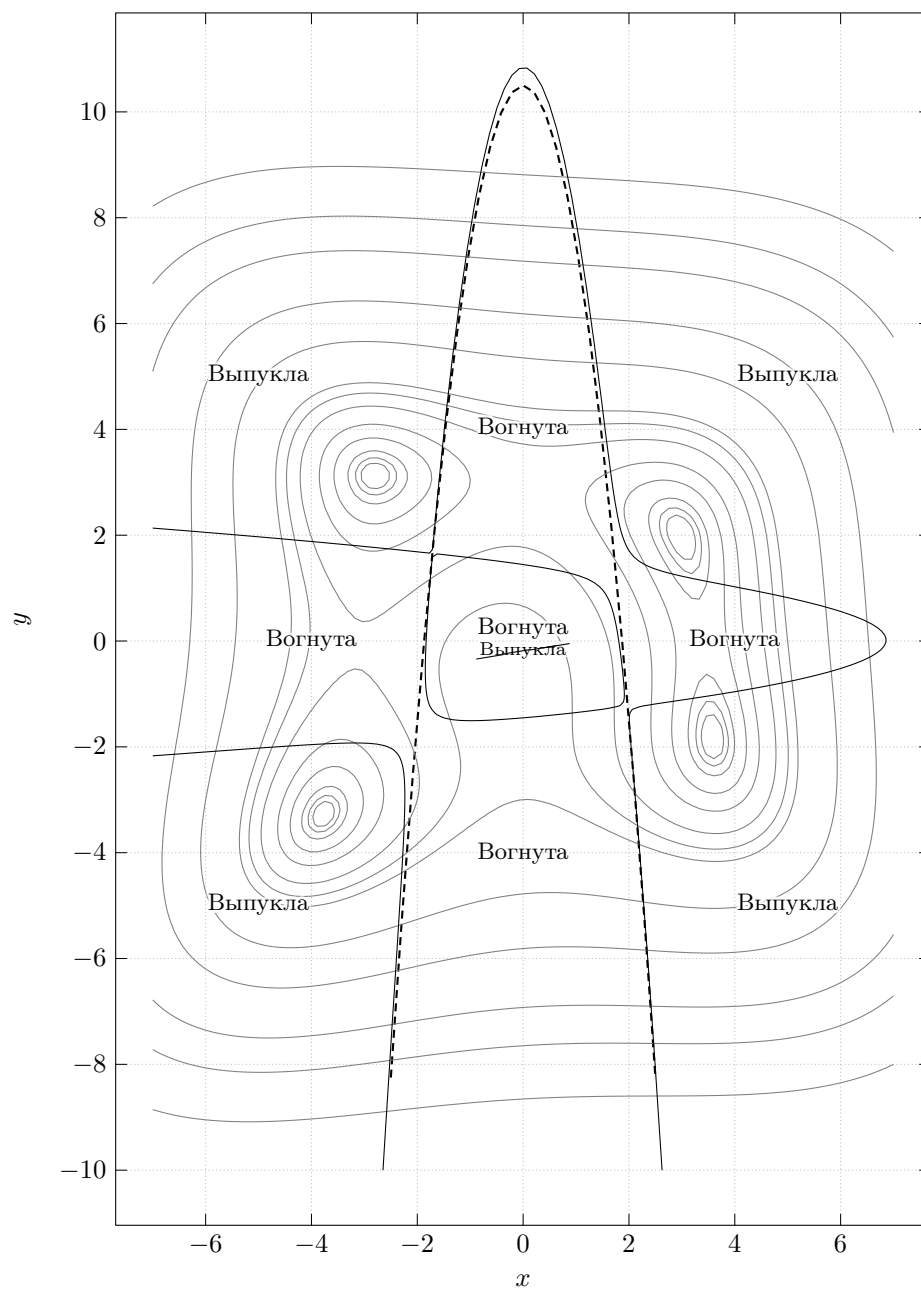


Рис. 12: Линии уровня функции Химмельблау (χ -3.27) и границы областей выпуклости

ИСПОЛЬЗОВАНИЕ RELCH

На рисунке 13 представлены результаты работы алгоритма RELCH для решения задачи минимизации функции (χ -3.27). Во всех случаях использовалось значение $s = 50$. На иллюстрации также видно, что из-за многоэкстремальности целевой функции результат работы алгоритма зависит от выбора начального приближения.

В отсутствие эталонных данных о минимумах проверить найденные решения на единственность можно было бы, воспользовавшись информацией об областях выпуклости функции и теоремами из раздела 1.2.1.

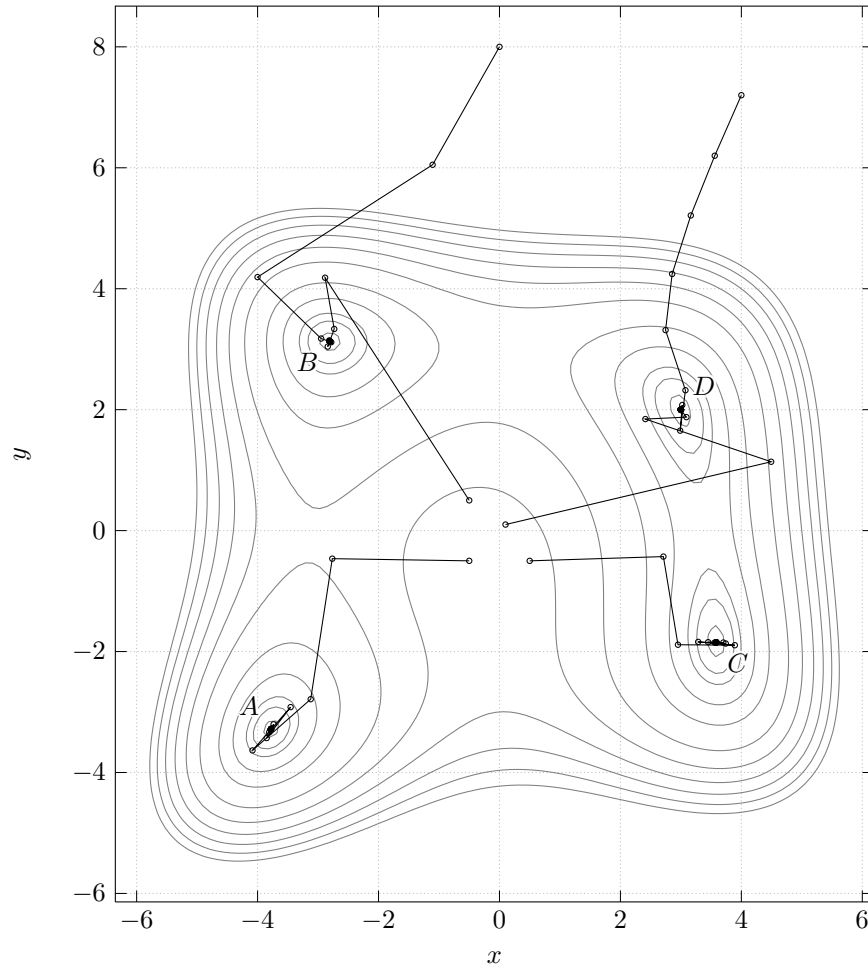


Рис. 13: Результаты работы алгоритма с функцией Химмельблау (χ -3.27) в зависимости от выбора начальной точки

3.3.3. Экспоненциальная функция

Рассмотрим функцию

$$f(x, y) = \sum_{a \in A} [e^{-xa} - e^{-ya} - (e^{-a} - e^{-10a})]^2 \quad (e-3.31)$$

Здесь суммирование происходит по значениям $a \in A = \{0.1, 0.2, \dots, 1\}$.

Выпуклость

Сложность функции (e-3.31) не позволяет провести анализ её выпуклости так легко, как это было сделано для функции Розенброка. Матрица Гессе в точке (x, y) имеет вид:

$$\sum_{a \in A} \begin{bmatrix} 2a^2 \times \begin{pmatrix} e^{-ax}(2e^{-ax} - e^{-ay} - c_a) & -e^{-a(x+y)} \\ -2a^2 e^{-a(x+y)} & e^{-ay}(e^{-ax} - c_a) \end{pmatrix} \end{bmatrix} \quad (3.32)$$

где $c_a = e^{-a} - e^{-10a}$. В силу положительности a , $c_a > 0$.

Попробуем использовать для исследования её на определённости критерий Сильвестра. Угловые миноры должны быть положительны:

$$\begin{aligned} \Delta_1 &= \sum_{a \in A} [2a^2 e^{-ax}(2e^{-ax} - e^{-ay} - c_a)] > 0 \\ \Delta_2 &= \sum_{a \in A} [e^{-ax}(2e^{-ax} - e^{-ay} - c_a)] \times \sum_{a \in A} [2a^2 e^{-ay}(2e^{-ay} - e^{-ax} + c_a)] - \\ &\quad - \left(\sum_{a \in A} [2a^2 e^{-ax-ay}] \right)^2 > 0 \end{aligned}$$

Решить это неравенства аналитически нелегко. Обратимся к численным методам. На рисунке 14 изображены найденные численно границы областей выпуклости, в которых миноры Δ_1 и Δ_2 обращаются в ноль. Оказывается, что функция выпукла лишь внутри узкой области, заключённой между границами $\Delta_2 = 0$.

Аналитическое решение

Найдём минимальное значение (e-3.31) аналитически. Из условий стационарности имеем

$$\frac{\partial f}{\partial x} = -\frac{\partial f}{\partial y} = \sum_{a \in A} [2ae^{-ay}(e^{-ax} - e^{-ay} - (e^{-a} - e^{-10a}))] = 0$$

откуда определяется единственное решение — точка $A(1, 10)$. Точка A попадает в область выпуклости функции (см. рис. 14), а потому является глобальным минимумом.

Овражность

Несмотря на видимую вытянутость линий уровня и быстрое стремление к бесконечности при $x < 0, y < 0$, рассматриваемая функция не обладает высокой степенью овражности даже в окрестности точки минимума:

$$\begin{aligned} \eta(0, 1) &\approx 140.57 & \eta(1, 5) &\approx 40.5 \\ \eta(1, 9) &\approx 257.93 & \eta(1.5, 11.5) &\approx 142.5 \end{aligned} \quad (3.33)$$

Как оказывается, и в точке A степень овражности невысока:

$$\eta(A) = \frac{1.75}{0.004} \approx 437.5$$

Поэтому использование высоких s в RELCH не обязательно.

Минимизация с помощью RELCH

На рисунке 15 представлены результаты работы RELCH с рассматриваемой функцией при $s = 20$. Алгоритм стабильно определяет минимум в точке $(1, 10)$.

ПРИМЕНЕНИЕ GDRELCH

Как видно из таблицы 2, при начальной точке в $(2, 6)$ и шаге $h = 0.001$ метод GDRELCH стабилизируется уже в окрестности *начальной точки* с $\|f'(x)\| = 0.86$, после чего быстро минимизирует функцию до порядка 10^{-10} .

Аналогичная ситуация имеет место при другом выборе начального приближения, см. рис. 16.

x_1	x_2	$f(x)$	$\ f'(x)\ $
2	6	0.75	0.86
2	6	0.75	0.86
2	6	0.75	0.86
2	6	0.75	0.86
2	6	0.75	0.86
2	6	0.75	0.86
1.99	6	0.75	0.86
1.99	6	0.14	0.86
1	5.23	$3.51 \cdot 10^{-2}$	0.36
0.8	8.55	$3.97 \cdot 10^{-3}$	0.37
1.06	9.73	$1.77 \cdot 10^{-3}$	0.11
0.95	9.91	$2.2 \cdot 10^{-4}$	$8.08 \cdot 10^{-2}$
1.02	9.97	$1.59 \cdot 10^{-4}$	$2.74 \cdot 10^{-2}$
0.99	10	$1.81 \cdot 10^{-6}$	$2.38 \cdot 10^{-2}$
1	10	$1.47 \cdot 10^{-7}$	$2.51 \cdot 10^{-3}$
1	10	$2.55 \cdot 10^{-9}$	$7.16 \cdot 10^{-4}$
1	10	$3.14 \cdot 10^{-11}$	$9.31 \cdot 10^{-5}$

Таблица 2: Минимизация функции (e-3.31) алгоритмом GDRELCH при шаге $h = 0.001$.

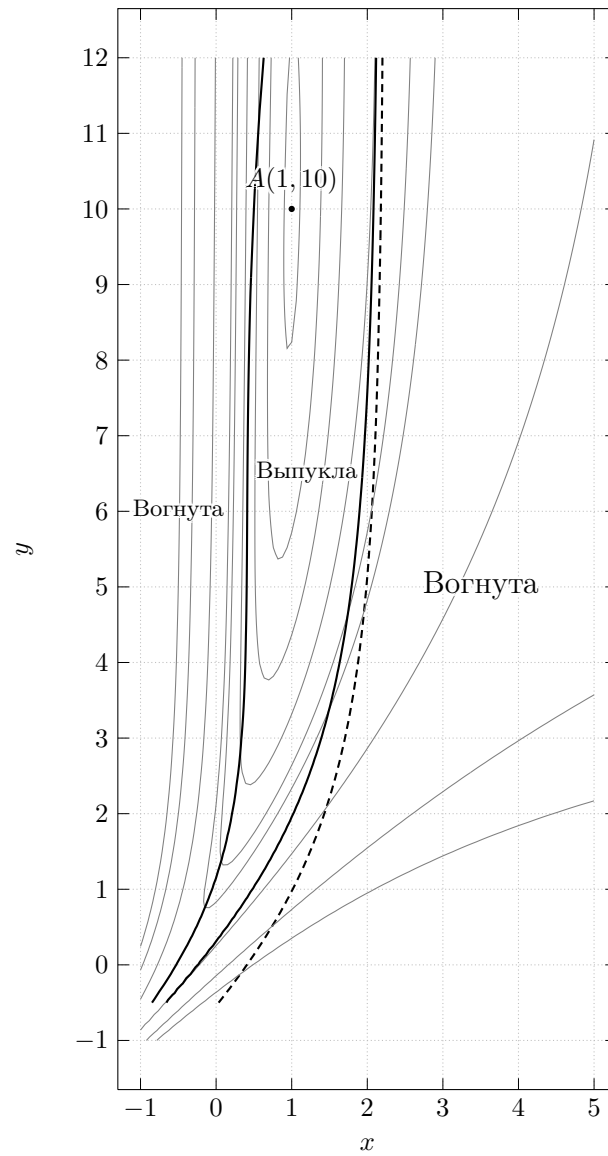
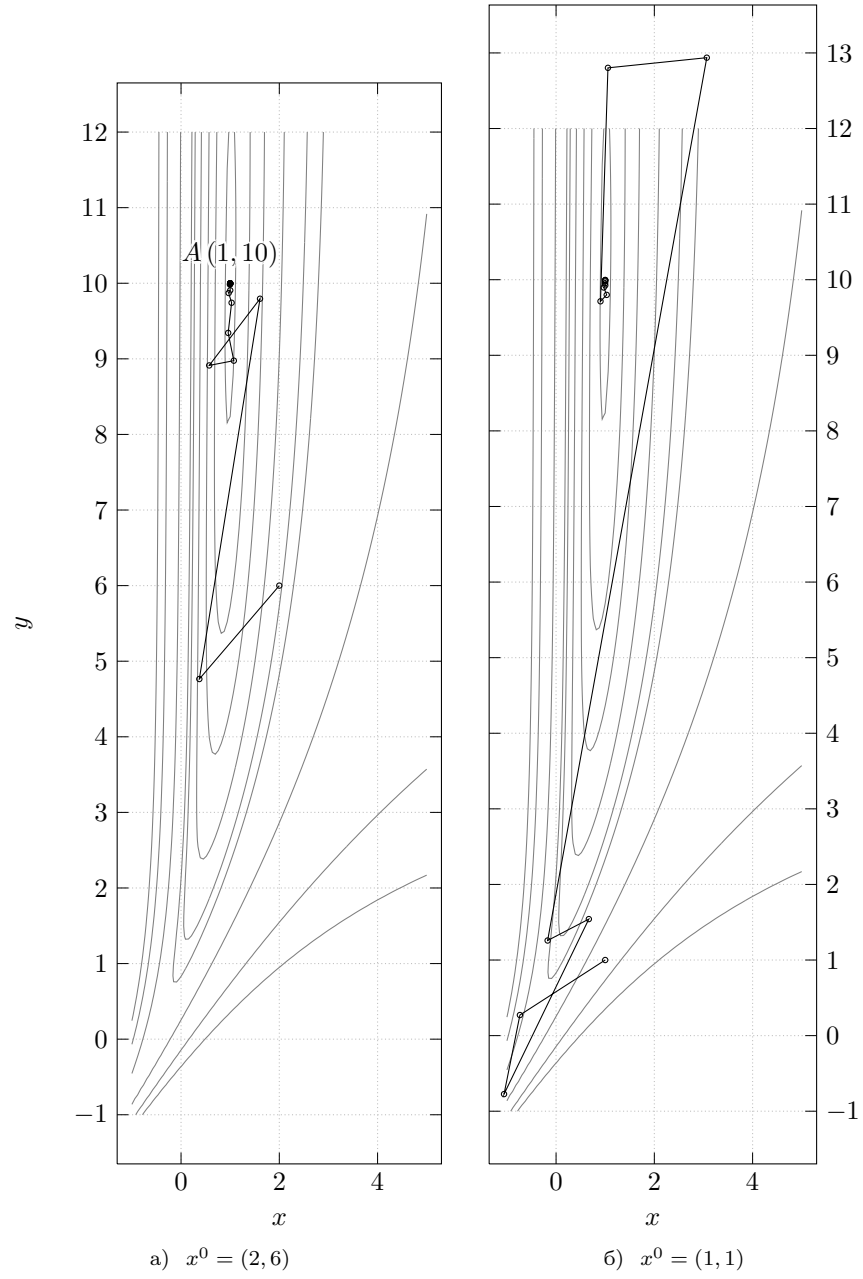


Рис. 14: Линии уровней $0.01, \dots, 2$ функции (e-3.31), границы области выпуклости $\Delta_1 = 0$, $\Delta_2 = 0$ и глобальный минимум в точке $A(1, 10)$.

Рис. 15: Минимизация функции (e-3.31) алгоритмом RELCH при $s = 20$.

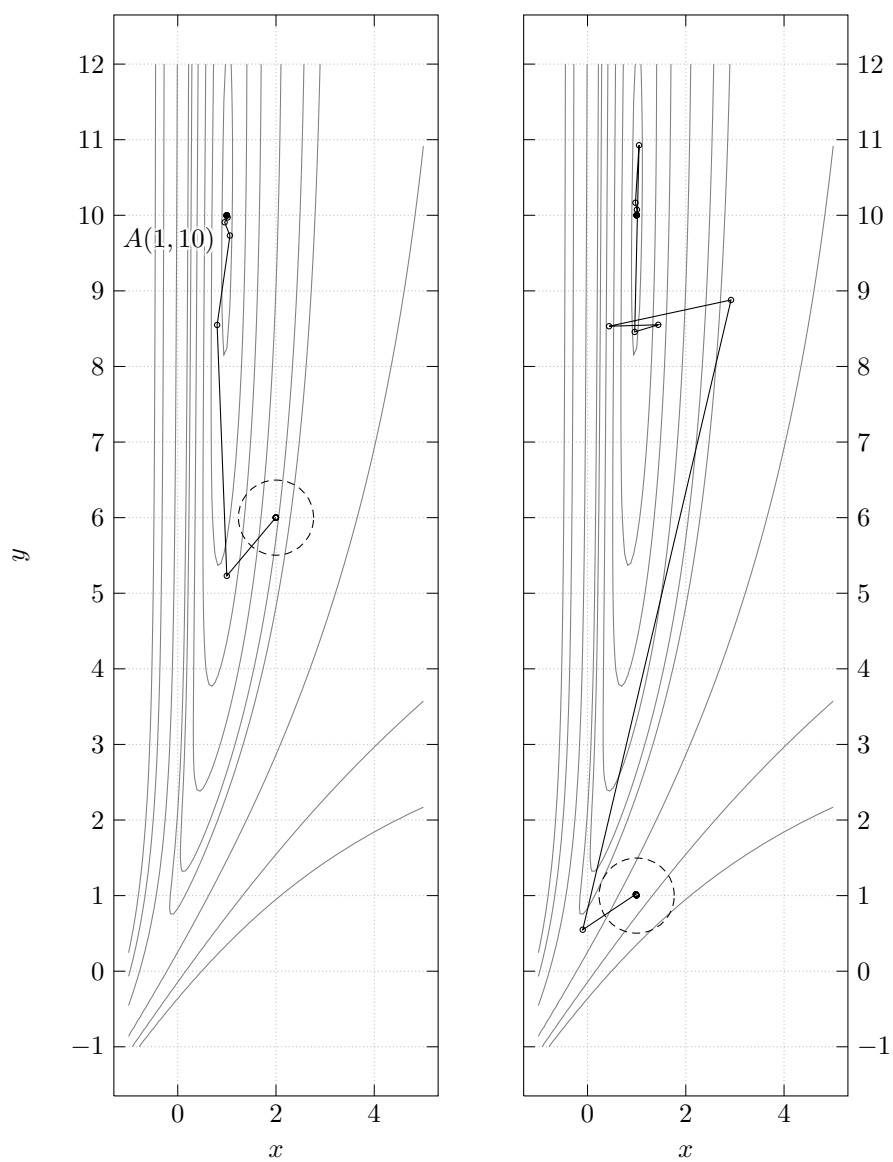
а) $x^0 = (2, 6)$, $h = 0.001$ (табл. 2)б) $x^0 = (1, 1)$, $h = 0.001$

Рис. 16: Минимизация функции (е-3.31) алгоритмом GDRELCH.

3.3.4. Многопараметрическая оптимизация

В качестве дополнительного примера приведём результат оптимизации функции четырёх переменных, заданной следующим образом:

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4 \quad (3.34)$$

Особенность функции (3.34) состоит в том, что она имеет вырожденную матрицу Гессе в точке минимума.

x_1	x_2	x_3	x_4	$f(x)$	$\ f'(x)\ $
-3	-1	-3	-1	102.91	1 154.69
-2.79	1.04	-0.65	-1.85	83.06	224.79
0.29	-0.89	-0.78	0.56	4.34	172.79
-0.54	0.25	$-7.48 \cdot 10^{-2}$	-0.38	0.35	39.81
-0.13	$-4.22 \cdot 10^{-2}$	-0.13	$-4.36 \cdot 10^{-2}$	$7.96 \cdot 10^{-3}$	11.19
-0.13	$2.1 \cdot 10^{-2}$	$-5.99 \cdot 10^{-2}$	$-7.25 \cdot 10^{-2}$	$6.29 \cdot 10^{-3}$	1.66
$-3.72 \cdot 10^{-2}$	$-3.77 \cdot 10^{-3}$	$-2.78 \cdot 10^{-2}$	$-1.63 \cdot 10^{-2}$	$2.34 \cdot 10^{-5}$	1.51
$-2.97 \cdot 10^{-2}$	$3.42 \cdot 10^{-3}$	$-1.52 \cdot 10^{-2}$	$-1.58 \cdot 10^{-2}$	$5.73 \cdot 10^{-7}$	$9.07 \cdot 10^{-2}$
$-1.98 \cdot 10^{-2}$	$1.93 \cdot 10^{-3}$	$-1.03 \cdot 10^{-2}$	$-1.03 \cdot 10^{-2}$	$5.02 \cdot 10^{-8}$	$9.4 \cdot 10^{-3}$
$-1.03 \cdot 10^{-2}$	$1.04 \cdot 10^{-3}$	$-5.35 \cdot 10^{-3}$	$-5.36 \cdot 10^{-3}$	$2.56 \cdot 10^{-8}$	$3.14 \cdot 10^{-3}$
$-3.98 \cdot 10^{-3}$	$3.82 \cdot 10^{-4}$	$-1.95 \cdot 10^{-3}$	$-1.94 \cdot 10^{-3}$	$3.87 \cdot 10^{-10}$	$3.11 \cdot 10^{-3}$
$-3.7 \cdot 10^{-3}$	$3.7 \cdot 10^{-4}$	$-1.83 \cdot 10^{-3}$	$-1.83 \cdot 10^{-3}$	$8.22 \cdot 10^{-11}$	$1.39 \cdot 10^{-5}$

Таблица 3: Минимизация функции (3.34) методом RELCH при $s = 100$

Получаемый результат близок к эталонному — точке $(0, 0, 0, 0)$, приведённой в [6].

3.4. Выводы

Описанный и протестированный алгоритм RELCH обладает следующими особенностями:

- Теоретически доказана бесконечная скорость релаксации RELCH.
- Метод RELCH обладает устойчивостью к овражности целевого функционала.
- Возможно алгоритмическое определение значения параметра s с помощью предварительного использования метода ПГС.
- Метод относительно просто реализуется, в алгоритме нет концептуально сложных вычислений.
- Автором метода в [3] также указывается на возможность применения алгоритма для оптимизации многопараметрических систем, в которых гессиан целевого функционала представляет собой разреженную матрицу большой размерности.

В разделе А представлены исходные тексты реализации RELCH и GDRELCH на языке Scheme.

4. Условная оптимизация

Рассмотрим задачу условной оптимизации на поиск экстремума целевой функции $f(x)$ в области D , заданной ограничениями типа неравенств:

$$\begin{cases} f(x) \rightarrow \text{extr} \\ g_j(x) \leq 0, j = \overline{1, m} \\ x \in \mathbb{R}^n \end{cases} \quad (4.1)$$

ЗАМЕЧАНИЕ 4.1. Ограничение типа равенства

$$g(x) = 0$$

эквивалентно паре ограничений типа неравенств

$$\begin{aligned} g(x) &\leq 0 \\ g(x) &\geq 0 \end{aligned}$$

Далее рассматриваются различные методы решения задач типа (4.1). Теоретическое описание каждого метода иллюстрируется практическим решением модельной задачи (4.3).

4.1. Теорема Куна—Таккера

На основе теоремы Куна—Таккера построен *аналитический* метод решения задачи (4.1), позволяющий найти её решения точно.

Более подробно тема освещена в [5], [8], [2] и [9].

4.1.1. Теоретические сведения

ОПРЕДЕЛЕНИЕ 4.2. *Функцией Лагранжа* задачи (4.1) называют функцию

$$\mathcal{L}(x, \lambda_0, \lambda) = \lambda_0 f(x) + \sum_{j=1}^m \lambda_j g_j(x) \quad (4.2)$$

где λ — вектор $\lambda_1, \dots, \lambda_m$. При $\lambda_0 = 1$ функция Лагранжа называется *классической*.

ОПРЕДЕЛЕНИЕ 4.3. Ограничение $g_k(x)$ задачи (4.1) называется *активным* в точке x_0 , если $g_k(x_0) = 0$.

ТЕОРЕМА 4.4 (Куна—Таккера). Пусть точка \hat{x} является решением задачи (4.1) с функцией Лагранжа (4.2) при соответствующих λ_0 и λ . Тогда выполнены следующие условия:

- а) $\frac{\partial \mathcal{L}}{\partial x_i} = 0, i = \overline{1, n}$ (условие стационарности функции Лагранжа)
- б) $\lambda_j \geq 0, j = \overline{1, m}$, если \hat{x} — точка минимума, и $\lambda_j \leq 0$, если это точка максимума.
- в) $\lambda_j \cdot g_j(x) = 0, j = \overline{1, m}$ (условие дополняющей нежёсткости)
- г) $g_j(x) \leq 0, j = \overline{1, m}$
- д) $\lambda_0^2 + \|\lambda\|^2 > 0$ (условие нетривиальности решения)

ОПРЕДЕЛЕНИЕ 4.5. Точка x^* , удовлетворяющая условиям теоремы 4.4, называется *условно-стационарной*. Если при этом $\|\lambda\| \neq 0$, то x^* называется *регулярной* условно-стационарной точкой.

Грубо говоря, регулярные условно-стационарные точки образуются за счёт вида самой целевой функции, в то время как существование нерегулярных обеспечиваются лишь структурой ограничений.

При использовании теоремы 4.4 для поиска условно-стационарных точек рассматривают два варианта значений λ_0 в функции Лагранжа: $\lambda_0 = 0$ и $\lambda_0 \neq 0$. В последнем случае без ограничения общности обычно полагают $\lambda_0 = 1$.

ТЕОРЕМА 4.6 (Условие регулярности). *Для того, чтобы в условно-стационарной точке x^* выполнялось неравенство $\lambda_0 \neq 0$, достаточно линейной независимости градиентов активных в этой точке ограничений*

$$g'_k(x^*), g'_{k+1}(x^*), \dots, g'_{k+l}(x^*)$$

Если удаётся показать, что во *всех* допустимых точках задачи (4.1) выполнено условие регулярности, случай $\lambda_0 = 0$ можно исключить из рассмотрения. Для этого также удобно использовать следующее условие.

ТЕОРЕМА 4.7 (Условие Слейтера). *Для $\lambda_0 \neq 0$ в условиях теоремы 4.4 достаточно существования такой точки x_s , в которой все неравенства ограничений выполняются строго: $g_j(x_s) < 0$, $j = \overline{1, m}$.*

Введённые в разделе 1.2.1 понятия выпуклости и вогнутости функций и множеств используются в следующей теореме, в которой говорится о случае *достаточности* условий теоремы 4.4.

ТЕОРЕМА 4.8. *Для точек минимума необходимые условия Куна—Таккера становятся достаточными в случае, когда целевая функция $f(x)$ и ограниченное неравенствами $g_j(x)$ множество выпуклы.*

В случае точек максимума достаточность достигается при вогнутости функции $f(x)$ и выпуклости множества допустимых решений задачи.

Может оказаться, что существование в определённой точке экстремума не удаётся доказать, опираясь лишь на теоремы 4.4 и 4.8. В таком случае задачу исследуют с применением условий высших порядков, которые приведены далее.

В следующих теоремах используются полные дифференциалы функции Лагранжа $d^2\mathcal{L}$ и ограничений dg_j , определяемые следующим образом:

$$d^2\mathcal{L} = \sum_{p=1}^n \sum_{r=1}^n \frac{\partial^2 \mathcal{L}}{\partial x_p \partial x_r} dx_p dx_r$$

$$dg_j = \sum_{p=1}^n \frac{\partial g_j}{\partial x_p} dx_p$$

ТЕОРЕМА 4.9 (Необходимое условие экстремума второго порядка). *Пусть точка \hat{x} — регулярный экстремум в задаче (4.1), удовлетворяющий условиям теоремы 4.4 при соответствующем λ . Тогда*

$$d^2\mathcal{L}(\hat{x}) \geq 0 \quad \text{для минимума}$$

$$d^2\mathcal{L}(\hat{x}) \leq 0 \quad \text{для максимума}$$

при $dx \neq 0$ таких, что

$$dg_j(\hat{x}) = 0 \quad \forall j : \lambda_j \neq 0$$

$$dg_j(\hat{x}) \leq 0 \quad \forall j : \lambda_j = 0$$

ТЕОРЕМА 4.10 (Достаточное условие экстремума первого порядка). Пусть x^* является регулярной условно-стационарной точкой задачи (4.1), а число активных в x^* ограничений равно числу переменных n . Тогда для наличия в x^* регулярного экстремума достаточно выполнения одного из следующих наборов неравенств при $\forall j : g_j(\hat{x}) = 0$:

$$\begin{aligned}\lambda_j &> 0 && \text{для минимума} \\ \lambda_j &< 0 && \text{для максимума}\end{aligned}$$

ТЕОРЕМА 4.11 (Достаточное условие экстремума второго порядка). Пусть x^* является регулярной условно-стационарной точкой задачи (4.1). Если при $dx \neq 0$ таких, что

$$\begin{aligned}dg_j(x^*) &= 0 && \forall j : \lambda_j \neq 0 \\ dg_j(x^*) &\leq 0 && \forall j : \lambda_j = 0\end{aligned}$$

выполняется неравенство $d^2\mathcal{L}(x^*) \neq 0$, то x^* — точка регулярного экстремума, причём

$$\begin{aligned}d^2\mathcal{L}(x^*) &> 0 && \text{для минимума} \\ d^2\mathcal{L}(x^*) &< 0 && \text{для максимума}\end{aligned}$$

4.1.2. Алгоритм метода

Пользуясь введёнными в предыдущем разделе определениям и теоремами, сформулируем алгоритм решения задачи (4.1). В нём после применения теоремы Куна—Таккера точки добавляются к решению или исключаются из дальнейшего рассмотрения по мере выполнения для них различных достаточных условий или невыполнения необходимых, соответственно.

- Шаг 1. Составить функцию Лагранжа (4.2). При выполнении условий теорем 4.6 или 4.7 положить $\lambda_0 = 0$.
- Шаг 2. Определить условно-стационарные точки из необходимых условий Куна—Таккера теоремы 4.4.
- Шаг 3. Добавить к решению точки условного минимума или максимума, если для них условия Куна—Таккера оказываются достаточными согласно теореме 4.8.
- Шаг 4. Для оставшихся точек проверить достаточное условие первого порядка (теорема 4.10). Точки, для которых оно выполняется, добавляются к решению. Затем к решению добавляются точки, для которых выполняется достаточное условие второго порядка (теорема 4.11).
- Шаг 5. Для оставшихся точек проверить необходимое условие второго порядка (теорема 4.9). Точки, для которых оно не выполняется, решениями задачи быть не могут и исключаются из рассмотрения.
- Шаг 6. Оставшиеся точки (то есть те, для которых выполнены все необходимые, но ни одно из достаточных условий) требуют дальнейшего исследования.

4.1.3. Нахождение точного аналитического решения

Рассмотрим методику применения изложенных в предыдущем разделе теоретических соображений на практическом примере.

К решению предлагается следующая задача:

$$\begin{cases} f(x) = (x_1 + 4)^2 + (x_2 - 4)^2 \rightarrow \text{extr} \\ 2x_1 - x_2 \leq 2 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases} \quad (4.3)$$

После приведения к каноническому виду она примет вид

$$\begin{cases} f(x) = (x_1 + 4)^2 + (x_2 - 4)^2 \rightarrow \text{extr} \\ g_1(x) = 2x_1 - x_2 - 2 \leq 0 \\ g_2(x) = -x_1 \leq 0 \\ g_3(x) = -x_2 \leq 0 \end{cases} \quad (4.4)$$

Заметим, что выполняется условие теоремы 4.7, так как в качестве соответствующей точки x_s можно взять, например, точку $(1, 1)$. Значит, в данной задаче достаточно рассмотреть лишь случай классической функции Лагранжа с $\lambda_0 = 1$.

Составим функцию Лагранжа:

$$\begin{aligned} \mathcal{L}(x, \lambda) = (x_1 + 4)^2 + (x_2 - 4)^2 + \\ + \lambda_1(2x_1 - x_2 - 2) + \lambda_2(-x_1) + \lambda_3(-x_2) \end{aligned} \quad (4.5)$$

Запишем необходимые условия стационарности точки x^* при соответствующем векторе λ :

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial x_1} = 2(x_1^* + 4) + 2\lambda_1 - \lambda_2 = 0 \\ \frac{\partial \mathcal{L}}{\partial x_2} = 2(x_2^* - 4) - \lambda_1 - \lambda_3 = 0 \end{cases} \quad (4.6a)$$

$$\text{sgn}(\lambda_1) = \text{sgn}(\lambda_2) = \text{sgn}(\lambda_3) \quad (4.6б)$$

$$\begin{cases} \lambda_1 \cdot g_1(x^*) = \lambda_1 \cdot (2x_1^* - x_2^* - 2) = 0 \\ \lambda_2 \cdot g_2(x^*) = \lambda_2 \cdot (-x_1^*) = 0 \\ \lambda_3 \cdot g_3(x^*) = \lambda_3 \cdot (-x_2^*) = 0 \end{cases} \quad (4.6в)$$

$$\begin{cases} g_1(x^*) = 2x_1^* - x_2^* - 2 \leq 0 \\ g_2(x^*) = -x_1^* \leq 0 \\ g_3(x^*) = -x_2^* \leq 0 \end{cases} \quad (4.6г)$$

Рассмотрим $2^3 = 8$ вариантов удовлетворения условий дополняющей нежёсткости (4.6в).

I) $\lambda_1 = 0$

В этом случае уравнения (4.6а) принимают вид:

$$\begin{cases} 2x_1^* + 8 - \lambda_2 = 0 \\ 2x_2^* - 8 - \lambda_3 = 0 \end{cases} \quad (4.7)$$

а) $\lambda_2 = 0$

Из (4.7) имеем $2x_1^* + 8 = 0 \iff x_1^* = -4$, что не удовлетворяет условию $g_2(x^*) = -x_1^* \leq 0$ из (4.6г).

б) $\lambda_2 \neq 0$

В этом случае из (4.6в) следует, что $g_2(x^*) = 0 \iff x_1^* = 0$, откуда согласно (4.7) получаем $\lambda_2 = 8$.

1) $\lambda_3 = 0$

Из (4.7) следует $x_2^* = 4$. Получаем точку $A = (0, 4)$.

2) $\lambda_3 \neq 0$

В данном случае согласно (4.6в) находим $x_2^* = 0$, поэтому из (4.7) следует, что $\lambda_3 = -8$. С учётом $\lambda_2 = 8$ заметим, что не выполняются условия (4.6б).

II) $\lambda_1 \neq 0$

Согласно условию (4.6в), в данном случае

$$2x_1^* - x_2^* - 2 = 0 \quad (4.8)$$

а) $\lambda_2 = 0$

Из (4.6а) получим

$$2x_1^* + 8 + 2\lambda_1 = 0 \quad (4.9)$$

1) $\lambda_3 = 0$

Второе уравнение системы (4.6а) даёт $2x_2^* - 8 - \lambda_1 = 0$. Сложим это уравнение с (4.9) и рассмотрим его вместе с (4.8), получив

$$\begin{cases} 2x_1^* + 4x_2^* - 8 = 0 \\ 2x_1^* - x_2^* - 2 = 0 \end{cases} \quad (4.10)$$

Таким образом получим $4x_2^* - 8 = -x_2^* - 2 \iff x_2^* = \frac{6}{5}$.

Значение $x_1^* = \frac{8}{5}$ определяется из (4.8). После этого из уравнения (4.9) найдём значение $\lambda_1 = -\frac{28}{5}$. Итак, получена ещё одна точка $B = (\frac{8}{5}, \frac{6}{5})$.

2) $\lambda_3 \neq 0$

Согласно (4.6в), в данном случае $x_2^* = 0$, поэтому из (4.8) следует $x_1^* = 1$. Тогда из (4.9) определим $\lambda_1 = -5$. Подставив найденные значения x_2^* и λ_1 в (4.6а), получим $\lambda_3 = -3$. Найдена очередная точка $C = (1, 0)$.

б) $\lambda_2 \neq 0$

Из (4.6в) получаем $x_1^* = 0$, откуда с учётом (4.8) следует значение $x_2^* = -2$, не удовлетворяющее условию $g_3(x^*) = -x_2^* \leq 0$ из (4.6г).

Итак, найдены три точки, для которых выполнены необходимые условия теоремы 4.4. Тип возможного экстремума определяется согласно знаку компонент λ .

- $A = (0, 4)$, $\lambda = (0, 8, 0)$, минимум
- $B = (\frac{8}{5}, \frac{6}{5})$, $\lambda = (-\frac{28}{5}, 0, 0)$, максимум
- $C = (1, 0)$, $\lambda = (-5, 0, -3)$, максимум

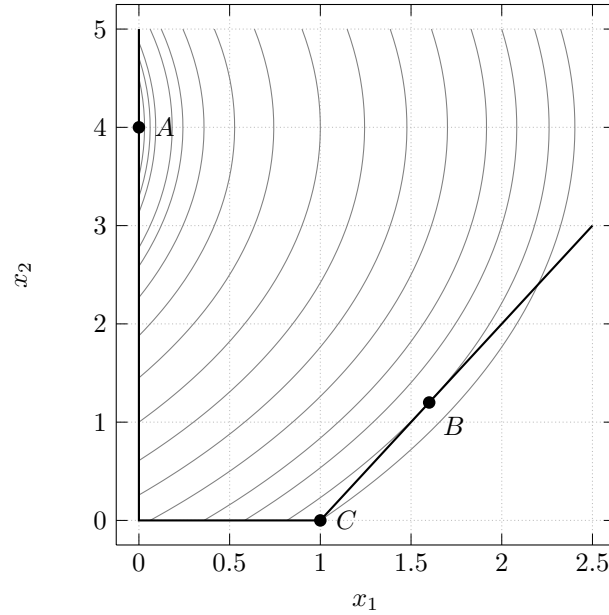


Рис. 17: Границы допустимого множества, линии уровня целевой функции и условно-стационарные точки задачи (4.4)

Поскольку целевая функция обладает положительно определённой матрицей Гессе

$$\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \quad (4.11)$$

она является выпуклой в силу теоремы 1.10. Рассматриваемое множество (см. рис. 17) также является выпуклым, так как функции ограничений $g_j(x)$ выпуклы в силу замечания 1.8, а потому выполняются условия теоремы 1.14.

Таким образом, согласно теореме 4.8, для точки A условия Куна—Таккера являются *достаточными*, и она является точкой *локального минимума*. Линии уровня функции $f(x)$, сходящиеся к A на рисунке 17, служат дополнительным подтверждением этому.

В то же время, для B и C условия Куна—Таккера не являются достаточными. Воспользуемся для их исследования условиями высших порядков.

Второй дифференциал $d^2\mathcal{L}$ во всех точках одинаковый и имеет вид

$$d^2\mathcal{L} = 2dx_1^2 + 2dx_2^2 \quad (4.12)$$

- $B = (\frac{8}{5}, \frac{6}{5})$

В данной точке активно лишь ограничение g_1 , так что воспользоваться достаточным условием первого порядка нельзя. Проверим необходимое условие второго порядка. Приравнивая к нулю дифференциал dg_1 активного ограничения и выбирая дифференциалы неактивных ограничений неположительными, получим

$$\begin{aligned} dg_1 &= 2dx_1 - dx_2 = 0 \iff dx_2 = 2dx_1 \\ dg_2 &= -dx_1 \leq 0 \\ dg_3 &= -dx_2 \leq 0 \end{aligned}$$

и с учётом этого рассмотрим (4.12) при условии $dx \neq 0$:

$$d^2\mathcal{L} = 2dx_1^2 + 8dx_1^2 = 10dx_1^2$$

Полученная форма очевидно больше нуля. Поскольку в B значения $\lambda < 0$, необходимое условие второго порядка теоремы 4.9 не выполняется, так что B не является точкой экстремума.

- $C = (1, 0)$

Активными являются ограничения g_1 и g_3 , их количество равно числу переменных $n = 2$, поэтому согласно теореме 4.10 выполняется достаточное условие экстремума первого порядка, и C есть точка *локального максимума*.

Действительно, из рисунка 18 видно, что любое допустимое перемещение из точки C приводит к переходу с линии уровня $f(C) = 41$ на линии меньших уровней. Вектор градиента $f'(C) = \begin{pmatrix} 5 \\ -4 \end{pmatrix}$ в данной точке образует с границами области тупые углы, так что нельзя увеличить значение функции, не выходя за пределы допустимого множества.

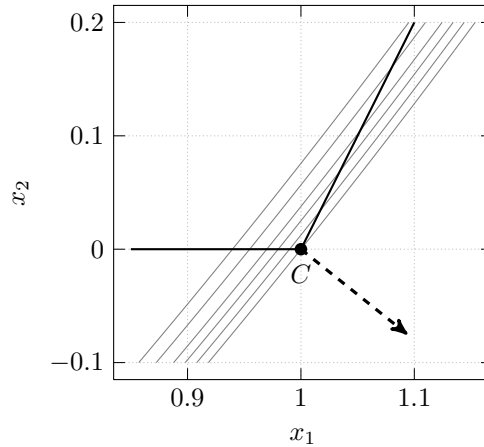


Рис. 18: Направление вектора градиента $f'(C)$, границы допустимого множества и линии уровня целевой функции $f(x)$ вблизи точки C

Эта ситуация напоминает уже виденную ранее (см. раздел 2.1.2, рис. 5).

4.1.4. Использование метода

Аналитический метод решения, построенный на использовании теоремы Куна—Таккера 4.4, позволяет точно определить условные экстремумы.

В то же время, реализация данного метода на алгоритмическом языке сопряжена с рядом проблем, основной из которых является концептуальная сложность алгоритмов логического вывода, которые придётся использовать при проверке условий теорем необходимости и достаточности. Кроме того, в общем случае метод также требует решения систем нелинейных уравнений.

Реализация аналитических машинных алгоритмов традиционно не является простой задачей, и в большинстве случаев на практике оказывается более выгодным (во всех отношениях) использование численных методов.

4.2. Метод Зойтендейка

4.2.1. Общая схема методов возможных направлений

Итерационные методы возможных направлений, к которым относится и метод Зойтендейка, реализуют решение задачи (4.1) по общей схеме, согласно которой поиск начинается в допустимой точке пространства решений и продолжается по траектории, обеспечивающей улучшение значения целевой функции, но не выходящей за границы допустимой области.

На каждом шаге очередное приближение к точке минимума находится по формуле

$$x^{k+1} = x^k + s_k \cdot d^k \quad (4.13)$$

Различие между методами возможных направлений заключается в стратегии выбора направления d^k и шага s_k .

4.2.2. Алгоритм метода Зойтендейка

В методе Зойтендейка на каждой итерации направление и шаг выбираются так, чтобы обеспечить наименьшую минимизирующую поправку к целевой функции без нарушения какого-либо ограничения.

Пусть в (4.1) функции ограничений $g_j(x)$ линейны или линеаризованы (с помощью разложения в ряд Тейлора) так, что получена следующая задача (рассматривается случай минимизации):

$$\begin{cases} f(x) \rightarrow \min \\ \langle a^j, x \rangle \leq b_j, j = \overline{1, m} \\ x_i \geq 0, i = \overline{1, n} \\ x \in \mathbb{R}^n \end{cases} \quad (4.14)$$

Данный метод не позволяет решать задачи с ограничениями типа равенств.

Теперь сформулируем алгоритм метода Зойтендейка, описанный в [6].

Начальное приближение x^1 выбирается из числа точек, удовлетворяющих ограничениям задачи (4.14).

- 1) Определяются множества номеров активных ограничений I^k — для ограничений вида $x_i \geq 0$ и J^k — для ограничений вида $\langle a^j, x \rangle \leq b_j$.
- 2) Если $I^k = J^k = \emptyset$, то $d^k = -f'(x^k)$. В противном случае решают задачу линейного программирования по минимизации скалярного произведения

$$\langle f'(x^k), d^k \rangle \rightarrow \min$$

при наличии ограничений

$$\langle a^j, d^k \rangle \leq 0, j \in J^k$$

что соответствует поиску такого допустимого направления, вдоль которого целевая функция убывает быстрее всего.

Для этого в векторе d^k каждую i -ю координату представляют¹² как d_i для $i \in I^k$ и $d_{i+} - d_{i-}$ для $i \notin I^k$, так что решаемая задача приобретёт

¹²Верхние индексы у d_i, d_{i+}, d_{i-} опущены для компактности.

следующий вид:

$$\begin{cases} \langle f'(x^k), d^k \rangle = \sum_{i \in I^k} \frac{\partial f(x^k)}{\partial x_i} d_i + \sum_{i \notin I^k} \frac{\partial f(x^k)}{\partial x_i} (d_{i+} - d_{i-}) \rightarrow \min \\ \langle a^j, d^k \rangle = \sum_{i \in I^k} a_i^j d_i + \sum_{i \notin I^k} a_i^j (d_{i+} - d_{i-}) \leq 0, j \in J^k \\ \sum_{i=1}^n |d_i^k| = \sum_{i \in I^k} d_i + \sum_{i \notin I^k} (d_{i+} + d_{i-}) \leq 1 \\ d_{i+} \geq 0, d_{i-} \geq 0, d_{i+} \cdot d_{i-} = 0, i \notin I^k \end{cases} \quad (4.15)$$

3) Если выполняется условие

$$\langle f'(x^k), d^k \rangle = 0 \quad (4.16)$$

то дальнейшее улучшение решения невозможно и процесс прерывается.

В противном случае определяют величину шага

$$s_k = \min\{s_*, \hat{s}_1, \dots, \hat{s}_n, \bar{s}_1, \dots, \bar{s}_m\} \quad (4.17)$$

где компоненты под \min находятся по следующим правилам:

- s_* — решение задачи безусловной одномерной оптимизации

$$f(x^k + s_* \cdot d^k) \rightarrow \min \quad (4.18)$$

- \hat{s}_i — наибольший шаг, при котором ограничение

$$x^k + \hat{s}_i d_i^k \geq 0$$

остаётся удовлетворённым. Если компонента найденного вектора направления $d_i^k \geq 0$, то $\hat{s}_i = +\infty$ (так что соответствующая компонента \hat{s}_j согласно (4.17) никак не ограничивает выбор s_k), иначе

$$\hat{s}_i = -\frac{x_i^k}{d_i^k}$$

- \bar{s}_j — наибольший шаг, при котором при движении из x^k по направлению d^k ограничение

$$\langle a^j, x^k + \bar{s}_j d^k \rangle \leq b_j$$

остаётся выполненным. Если $\langle a^j, d^k \rangle \leq 0$, то ограничение будет удовлетворено при любом шаге, поэтому $\bar{s}_j = +\infty$. В противном случае, \bar{s}_j определяется как

$$\bar{s}_j = \frac{b_j - \langle a^j, x^k \rangle}{\langle a^j, d^k \rangle}$$

Отметим, что $s_k > 0$.

4) С учётом шага s_k и направления d^k согласно (4.13) полагают новое приближение равным

$$x^{k+1} = x^k + s_k \cdot d^k$$

и переходят к следующей итерации.

Таким образом, критерием останова процесса оптимизации k -й итерации является выполнение равенства (4.16), что равносильно невозможности дальнейшего уменьшения значения целевой функции при заданных ограничениях.

4.2.3. Пример

Рассмотрим работу метода Зойтендейка на примере задачи (4.3). Будем искать точку минимума. Ограничения уже линейны, поэтому дополнительная линеаризация не требуется. Задача уже приведена к виду (4.14):

$$\begin{cases} f(x) = (x_1 + 4)^2 + (x_2 - 4)^2 \rightarrow \min \\ 2x_1 - x_2 \leq 2 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases} \quad (4.19)$$

Итерация 1. В качестве первого приближения x^1 возьмём точку $(1, 1)$, очевидно удовлетворяющую всем ограничениям.

Шаг 1. Ни одно из ограничений не активно в данной точке, поэтому множества $I^k = J^k = \emptyset$.

Шаг 2. Направление движения определим как противоположное вектору градиента:

$$d^1 = -f'(x^1) = \begin{pmatrix} -10 \\ 6 \end{pmatrix}$$

Шаг 3. Определим коэффициенты s_* , \hat{s}_1 , \hat{s}_2 , \bar{s}_1 .

- Минимум функции $f(x^1 + s_* \cdot d^1) = (5 - 10s_*)^2 + (6s_* - 3)^2$ достигается при $s_* = \frac{1}{2}$.
- $d_1^1 = -10 \not\geq 0 \Rightarrow \hat{s}_1 = -\frac{x_1^1}{d_1^1} = -\frac{1}{-10} = \frac{1}{10}$
- $d_2^1 = 6 \geq 0 \Rightarrow \hat{s}_2 = +\infty$
- $\langle a^1, d^1 \rangle = -26 \leq 0 \Rightarrow \bar{s}_1 = +\infty$

Находим величину шага: $s_1 = \min\{\frac{1}{2}, \frac{1}{10}\} = \frac{1}{10}$.

Шаг 4. Вычисляем новое приближение x^2 :

$$x^2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{1}{10} \begin{pmatrix} -10 \\ 6 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{8}{5} \end{pmatrix}$$

Итерация 2. Продолжим поиск минимума из точки

$$x^2 = \begin{pmatrix} 0 \\ \frac{8}{5} \end{pmatrix}$$

Шаг 1. В x^2 активно лишь ограничение $x_1 \geq 0$, поэтому $I^2 = \emptyset$, $J^2 = \{1\}$.

Шаг 2. Составим задачу линейного программирования:

$$\begin{cases} \langle f'(x^2), d^2 \rangle = \tilde{f}_2 = 8d_1 - \frac{3}{5}(d_{2+} - d_{2-}) \rightarrow \min \\ d_1 + (d_{2+} + d_{2-}) \leq 1 \\ d_1, d_{2+}, d_{2-} \geq 0 \\ d_{2+} \cdot d_{2-} = 0 \end{cases}$$

Приведём её к каноническому виду, получив в первом ограничении равенство путём добавления переменной x_1 :

$$\begin{cases} \tilde{f}_2 = 8d_1 - \frac{3}{5}(d_{2+} - d_{2-}) \rightarrow \min \\ d_1 + (d_{2+} + d_{2-}) + x_1 = 1 \\ d_1, d_{2+}, d_{2-}, x_1 \geq 0 \\ d_{2+} \cdot d_{2-} = 0 \end{cases}$$

Решим задачу симплекс-методом. Составим матрицу коэффициентов ограничений и целевой функции, затем исключим x_1 из состава базисных переменных:

$$\begin{bmatrix} \dagger & d_1 & d_{2+} & d_{2-} & x_1 & \diamond \\ x_1 & 1 & 1 & 1 & 1 & 1 \\ \tilde{f}_2 & -8 & \frac{3}{5} & -\frac{3}{5} & 0 & 0 \end{bmatrix} \begin{array}{l} \leftarrow -\frac{3}{5} \\ \leftarrow + \end{array} \quad (4.20)$$

Получим матрицу с одними отрицательными коэффициентами в нижнем ряду:

$$\begin{bmatrix} \dagger & d_1 & d_{2+} & d_{2-} & x_1 & \diamond \\ x_1 & 1 & 1 & 1 & 1 & 1 \\ \tilde{f}_2 & -\frac{43}{5} & 0 & -\frac{6}{5} & -\frac{3}{5} & -\frac{3}{5} \end{bmatrix}$$

При этом условие $d_{2+} \cdot d_{2-} = 0$ выполнено. Итак, найдено оптимальное решение задачи линейного программирования:

$$\begin{cases} d_1 = 0 \\ d_{2+} = 1 \\ d_{2-} = 0 \end{cases}$$

с учётом которого построим вектор d^2 :

$$d^2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Шаг 3. Определим коэффициенты s_* , \hat{s}_1 , \hat{s}_2 , \bar{s}_1 .

- Минимум $f(x^2 + s_* \cdot d^2) = 16 + (s_* - \frac{32}{5})^2$ достигается при $s_* = \frac{32}{5}$.
- $d_1^2 = 0 \geq 0 \implies \hat{s}_1 = +\infty$
- $d_1^2 = 1 \geq 0 \implies \hat{s}_1 = +\infty$
- $\langle a^1, d^2 \rangle = -1 \leq 0 \implies \bar{s}_1 = +\infty$ Итак, ограничения никак не влияют на величину шага, поэтому $s_k = \frac{32}{5}$.

Шаг 4. Найдём приближение x^3 :

$$x^3 = \begin{pmatrix} 0 \\ \frac{8}{5} \end{pmatrix} + \frac{32}{5} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix}$$

Итерация 3. Как известно из точного аналитического решения (см. раздел 4.1), точка $(0, 4)$ является точным решением задачи (4.19). Значит, удовлетворение (4.16) на этой итерации должно сигнализировать об этом. Выполним первые два шага алгоритма, что убедиться в этом.

Шаг 1. Как и на прошлой итерации, $I^2 = \emptyset$, $J^2 = \{1\}$.

Шаг 2. Для нахождения компонент вектора d^k решим задачу линейного программирования:

$$\begin{cases} \langle f'(x^3), d^3 \rangle = \tilde{f}_3 = 8d_1 \rightarrow \min \\ d_1 + (d_{2+} + d_{2-}) + x_1 = 1 \\ d_1, d_{2+}, d_{2-}, x_1 \geq 0 \\ d_{2+} \cdot d_{2-} = 0 \end{cases}$$

оптимальным решением которой является тривиальный набор

$$\begin{cases} d_1 = 0 \\ d_{2+} = 0 \\ d_{2-} = 0 \end{cases}$$

Таким образом,

$$d^3 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Очевидно, что выполняется критерий останова (4.16). Значит, далее уменьшать значение $f(x)$ в (4.19) не представляется возможным.

Итак, с помощью метода возможных направлений Зойтендейка найдено решение задачи условной минимизации (4.19), которое согласуется с решением, найденным в разделе 4.1.

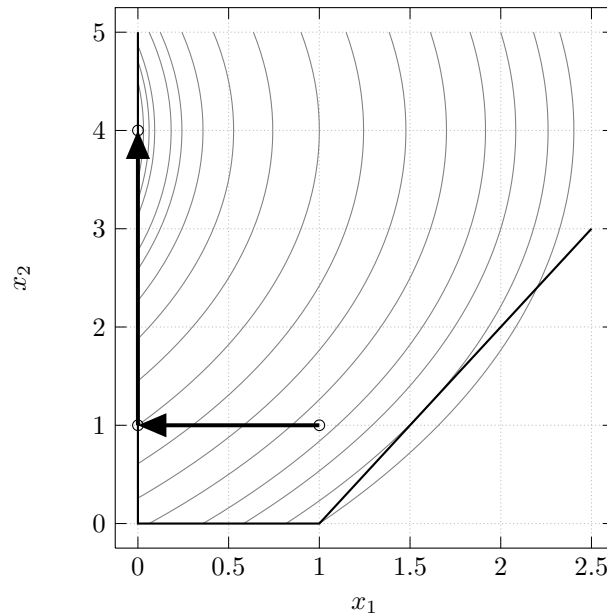


Рис. 19: Ход решения задачи (4.19) методом Зойтендейка

4.2.4. Использование метода

Метод Зойтендейка подходит для реализации на компьютере, поскольку не содержит принципиально сложных шагов. Единственным проблемным этапом может быть одномерная минимизация (4.18) для нахождения значения s_* .

Линеаризация ограничений может вносить определённую погрешность в решение.

Основным же недостатком методом Зойтендейка является невозможность решения задач с ограничениями типа равенств.

4.3. Метод штрафных функций

4.3.1. Общие сведения

Методы, построенные на применении штрафных функций, позволяют решать задачи на поиск минимума при наличии ограничений *численно*.

Общая схема таких методов заключается в замене задачи условной минимизации в некоторой области D , заданной ограничениями типа неравенств

$$\begin{cases} f(x) \rightarrow \min \\ g_j(x) \leq 0, j = \overline{1, m} \\ x \in \mathbb{R}^n \end{cases} \quad (4.21)$$

на задачу безусловной минимизации

$$P(x, r) = f(x) + \varphi(x, r) \rightarrow \min \quad (4.22)$$

где *штраф* $\varphi(x, r)$ удовлетворяет свойству

$$\begin{aligned} \varphi(x, r) &= 0 & x \in D \\ \varphi(x, r) &\gg 0 & x \notin D \end{aligned}$$

Таким образом, штрафная часть значительно возрастает при выходе за пределы допустимого множества D .

Метод штрафных функций позволяет сводить задачи оптимизации при наличии ограничений к задачам безусловной оптимизации, снижая таким образом структурную сложность решаемой задачи.

Один из подходов к использованию штрафных функций заключается в последовательном решении задач вида 4.22, параметризованных номером итерации k :

$$P_k(x, r) = f(x) + \varphi(x, p_k) \rightarrow \min \quad (4.23)$$

где функция $\varphi(x, p_k)$ масштабируется на каждой итерации в сторону увеличения или уменьшения.

4.3.2. Метод внешних штрафных функций

Методами внешних штрафных функций называют методы, использующие схему (4.23) таким образом, что $\varphi(x, p_k)$ с каждой итерацией увеличивается, и положения вне допустимой области с ростом k становятся всё менее «выгодными». При этом начальное приближение x^1 к точке минимума выбирается *вне* D . Благодаря этому приближение постепенно «скатывается» в область допустимых значений.

Один из таких методов, изложенный в [6] и [4], предусматривает построение штрафной функции для задачи (4.1) в виде:

$$\varphi(x, p_k) = p_k \sum_{j=1}^m [g_j^+(x)]^2 \quad (4.24)$$

где $g_j^+(x) = \max\{0, g_j(x)\}$, так что штрафная часть отлична от нуля и возрастает вне границ, заданных ограничениями $g_j(x)$, а $\lim_{k \rightarrow \infty} p_k = +\infty$.

В качестве критерия останова процесса (4.23) можно выбрать условие

$$\varphi(x^k, p_k) < \varepsilon$$

где x^k — приближение к точке минимума на k -ом шаге.

О сходимости метода внешних штрафных функций говорит следующая теорема.

ТЕОРЕМА 4.12. Если x^* — локальный минимум в задаче (4.21), то для достаточно больших p_k существует такая зависимость $x^*(p_k)$, определяющая безусловный локальный минимум в задаче (4.23) при штрафе (4.24), что

$$\lim_{p_k \rightarrow +\infty} x^*(p_k) = x^* \quad (4.25)$$

ЗАМЕЧАНИЕ 4.13. Теорема (4.12), вообще говоря, не гарантирует возможность определения глобального минимума задачи (4.21).

4.3.3. Пример

В разделе 4.1 была решена задача условной оптимизации (4.4):

$$\begin{cases} f(x) = (x_1 + 4)^2 + (x_2 - 4)^2 \rightarrow \text{extr} \\ g_1(x) = 2x_1 - x_2 - 2 \leq 0 \\ g_2(x) = -x_1 \leq 0 \\ g_3(x) = -x_2 \leq 0 \end{cases}$$

Среди условных экстремумов функции был найден минимум в точке $A = (0, 4)$. Найдём его с помощью штрафных функций.

ПОСТРОЕНИЕ НОВОЙ ЦЕЛЕВОЙ ФУНКЦИИ

Преобразуем задачу к виду (4.23), выбирая штрафную часть в виде (4.24):

$$P(x) = f(x) + p \sum_{j=1}^3 [g_j^+(x)]^2 \rightarrow \min \quad (4.26)$$

Отметим, что (4.26) строго выпукла¹³, поэтому теорема 4.12 для неё выполняется и в смысле глобального минимума в силу теоремы 1.19.

На рисунке 20 представлены линии уровня функции $P(x)$ задачи (4.26) при различных значениях p (при $p = 0$ получается исходная функция $f(x)$ задачи (4.4)). Видно, что с ростом p влияние штрафной части вне D (----) значительно возрастает, тогда как в допустимой области оптимизируемая функция не претерпевает никаких изменений.

ИСПОЛЬЗОВАНИЕ RELCH

Для численного решения задачи (4.26) теперь уже безусловной минимизации воспользуемся описанным ранее (см. раздел 3) градиентным методом с чебышёвскими функциями релаксации, выбирая в качестве начального приближения точку вне области D .

Индекс k в (4.26) опущен не случайно: опыт показывает, что при достаточно большом начальном значении параметра p_0 использование итерационного процесса (4.23) может и не понадобиться, поскольку точка минимума A с достаточной точностью локализуется уже на первом шаге. Это проиллюстрировано на рисунке 21. Трассировка в таблице 4 показывает, что положение точки безусловного минимума немного смещено из $(0, 4)$ в $(0.004, 4)$. Это объясняется тем, что сходимость метода штрафных функций достигается при $p \rightarrow +\infty$. При использовании подобного подхода стоит учитывать возможную высокую овражность получающегося функционала.

¹³ Действительно, $[g_1(x)]^2 = (2x_1 - x_2 - 2)^2$ выпукла в силу неотрицательной определённости своей матрицы Гессе $\begin{pmatrix} 8 & -4 \\ -4 & 2 \end{pmatrix}$, остальные слагаемые штрафа также очевидно выпуклы. Исходная функция строго выпукла (см. раздел 4.1.3, с. 41), а потому в силу теоремы 1.16 выпукла и вся функция (4.26).

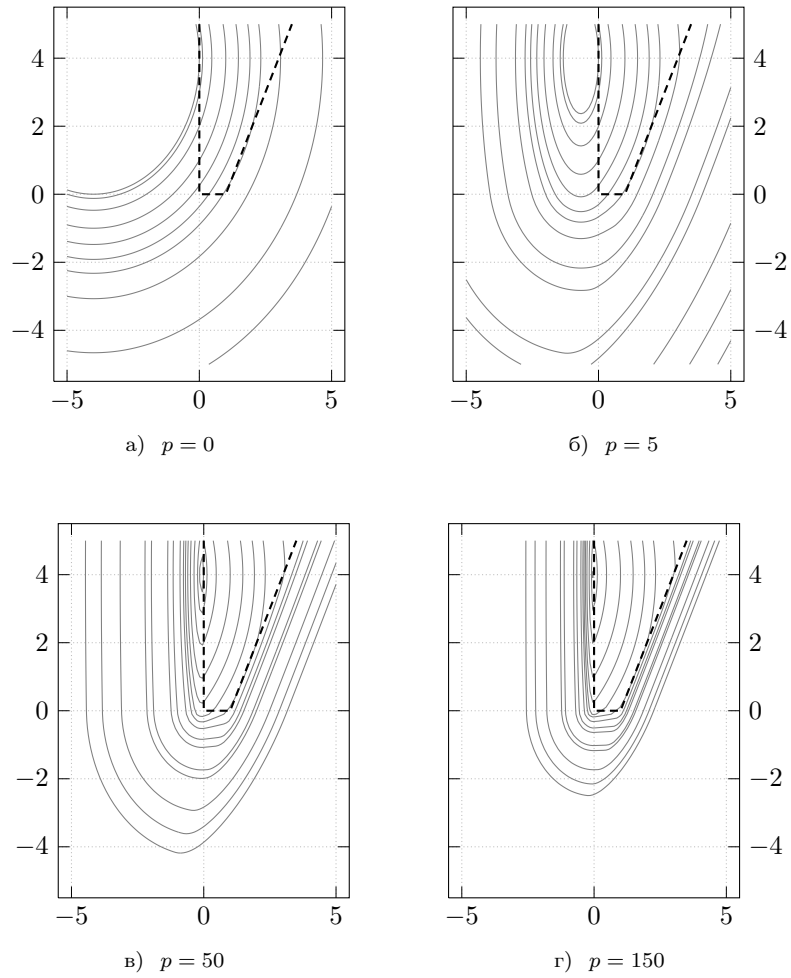


Рис. 20: Линии уровня функции $P(x)$ при различных значениях коэффициента штрафа p

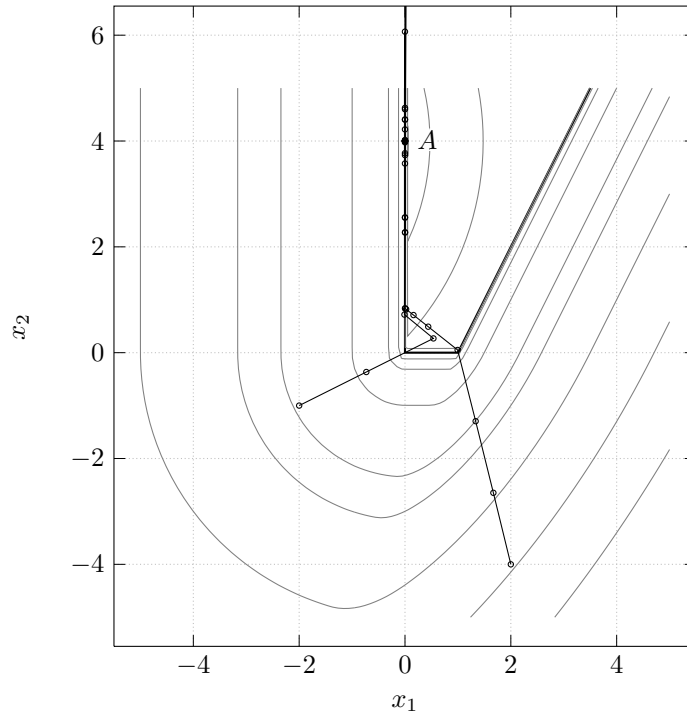


Рис. 21: Ход процесса решения задачи (4.26) с $p = 10^5$ методом RELCH с параметром $s = 200$ в зависимости от выбора начальной точки

x_1	x_2	$f(x)$	$\ f'(x)\ $
2	-4	$2.29 \cdot 10^5$	$3.12 \cdot 10^5$
1.67	-2.65	55 440.37	$2.08 \cdot 10^5$
1.34	-1.29	40.54	$1.02 \cdot 10^5$
0.99	$5.1 \cdot 10^{-2}$	32.02	12.73
0.44	0.49	28.13	11.32
0.16	0.71	26.28	10.61
$2.15 \cdot 10^{-2}$	0.82	26.05	10.25
$4.1 \cdot 10^{-3}$	0.83	26	10.21
$-2.47 \cdot 10^{-4}$	0.84	20.27	7.02
$-6.72 \cdot 10^{-4}$	6.07	18.09	6.83
$1.06 \cdot 10^{-4}$	2.55	18.09	8.51
$-5.46 \cdot 10^{-4}$	2.55	16.36	4.1
$-1.99 \cdot 10^{-4}$	4.6	16.05	4.19
$-6.67 \cdot 10^{-4}$	3.77	16	5.36
$-1.8 \cdot 10^{-4}$	4.02	16	4.39
$-6.69 \cdot 10^{-4}$	3.99	16	5.38
$-1.8 \cdot 10^{-4}$	4	16	4.39
$-4.25 \cdot 10^{-4}$	4	16	0.49
$-3.94 \cdot 10^{-4}$	4	16	0.12
$-4.02 \cdot 10^{-4}$	4	16	$3.44 \cdot 10^{-2}$

Таблица 4: Минимизация функции (4.26) при $p = 10^5$ методом RELCH с параметром $s = 200$ при начальном приближении $(2, -4)$.

А. Исходные тексты

А.1. Общая схема градиентных методов

Содержимое файла `gradient-methods.ss`

```
#lang scheme
```

```
(require (planet wmfarr/simple-matrix:1:0/matrix)
         "shared.ss"
         pyani-lib/matrix
         pyani-lib/vector
         pyani-lib/function-ops)

(define function? (flat-contract procedure?))
(define point? (flat-contract vector?))
(define shift? (flat-contract vector?))
(define gradient? (flat-contract vector?))
(define hessian? (flat-contract matrix?))
(define iteration-count? (flat-contract integer?))
(define epsilon? (and/c real? positive?))

(define listener-reply? (or/c void point? cons?))
(define listener? (point? shift? point? gradient? matrix? . -> . listener-reply?))

(define gradient-method?
  (->* (procedure? point? iteration-count?)
        (listener?)
        (or/c point? listener-reply?)))

(define shift-chooser? (point? vector? matrix? . -> . vector?))

(define stop-condition? (function? point? point? gradient? hessian? . -> . boolean?))

(provide/contract [gradient-method
                  (shift-chooser? stop-condition? . -> . gradient-method?)]
                  [zero-gradient-condition
                  (epsilon? . -> . stop-condition?)])

(provide/contract [function? contract?]
                  [point? contract?]
                  [iteration-count? contract?]
                  [epsilon? contract?]
                  [listener? contract?]
                  [listener-reply? contract?])

;;  $x^{k+1} = x^k - H_k(G_k, h_k)g_k$ 
(define (gradient-method choose-shift stop-condition)
  (define (optimize function x-start iterations [listener void])
    (if (<= iterations 0)
        x-start
        (let* ((g (@ (gradient function) x-start))
                (G (@ (hessian function) x-start))
                (shift (choose-shift x-start g G))
                (candidate-x-new (vector-add x-start shift))
                (listener-result (listener x-start
                                           shift
                                           candidate-x-new
                                           g G)))
              (if (cons? listener-result)
                  listener-result
                  (let ((x-new (if (vector? listener-result)
                                   listener-result
                                   candidate-x-new)))
                    (if (stop-condition function x-start x-new g G)))))))
    x-start))
```

```

                                x-new
                                (optimize function x-new
                                      (sub1 iterations)
                                      listener))))))
optimize)

;; ||f'(x)|| < ε
(define (zero-gradient-condition eps)
  (lambda (f x-start x-new g G)
    (<= (p-vector-norm g) eps)))

```

А.2. Алгоритмы RELCH, GDRELCH, ПГС и ПГС₂

Содержимое файла relch.ss

```

#lang scheme

(require "shared.ss"
        "gradient-methods.ss"
        (planet wmfarr/simple-matrix:1:0/matrix)
        pyani-lib/vector
        pyani-lib/matrix
        pyani-lib/function-ops)

(define parameter? (and/c integer? positive?))

(define optimization-method?
  (->* (function? point? iteration-count? epsilon? parameter?)
        (listener?)
        (or/c point? listener-reply?)))

(provide/contract
 [relch-optimize optimization-method?]
 [gd-optimize optimization-method?]
 [sgd-optimize optimization-method?]
 [gdrelch-optimize optimization-method?])

;;
;;
;; d ←  $\frac{d}{2}$ 
(define (enforce-relaxation shift f x-start [iter 1])
  (define (decrease-shift shift)
    (vector-scale shift 1/2))
  (define (better-minimum? f x-new x)
    (< (@ f x-new) (@ f x)))
  (let ((x-new (vector-add x-start shift)))
    (if (or (better-minimum? f x-new x-start)
            (> iter 1000))
        shift
        (enforce-relaxation (decrease-shift shift) f x-start
                              (add1 iter)))))

(define (enforce-relaxation-filtered shift f x-start)
  (define (random-vector length)
    (build-vector length (lambda (x) (* 2 (- (random 2) 1/2)))))
  (if (< (p-vector-norm shift) +inf.0)
      (enforce-relaxation shift f x-start)
      (random-vector (vector-length shift))))

(define (make-sgd-regulate factor)
  (lambda (shift f x-start)
    (vector-scale shift factor)))

(define (make-gd-optimize regulate-shift stop-condition)

```



```

(lambda (f x-start
        iterations eps
        [unused #f]
        [listener void])
  (define (choose-shift x-start g G)
    (let ((shift (vector-scale g -1)))
      (regulate-shift shift f x-start)))
  ((gradient-method choose-shift (stop-condition eps))
   f x-start iterations listener)))

(define (make-relch-optimize regulate-shift stop-condition)
  (lambda (f x-start
            iterations eps
            degree
            [listener void])
    (define (choose-shift x-start g G)
      (define (relch-shift L g G)
        (let* ((n (matrix-size G))
               (d1 (zero-vector n))
               (d2 (vector-scale g -2)))
          (define (sub2 x) (sub1 (sub1 x)))
          (define (approach-target-degree [prev d2] [preprev d1] [degree 3])
            (let ((current-degree-shift
                  (vector-add
                   (vector-add
                    (vector-scale
                     (matrix-vector-mul
                      (matrix-add (identity-matrix n) (matrix-scale G -2))
                      prev)
                     (/ (* 2 (sub1 degree)) degree))
                   (vector-scale preprev
                                (* -1 (/ (sub2 degree) degree))))
                  (vector-scale g
                                (/ (* -4 (sub1 degree)) degree))))))
              (if (= degree L)
                  current-degree-shift
                  (approach-target-degree current-degree-shift prev (add1 degree))))))
        (cond ((= L 1) d1)
              ((= L 2) d2)
              (else (approach-target-degree d2 d1))))))
      (let* ((G (normalize-matrix G))
             (g (normalize-vector g))
             (shift (relch-shift degree g G)))
        (regulate-shift shift f x-start)))
    ((gradient-method choose-shift (stop-condition eps))
     f x-start iterations listener)))

(define relch-optimize
  (make-relch-optimize enforce-relaxation-filtered
                       zero-gradient-condition))

(define (sgd-optimize f x-start iterations eps step-factor [listener void])
  (let ((optimize (make-gd-optimize (make-sgd-regulate step-factor)
                                    zero-gradient-condition)))
    (optimize f x-start iterations eps step-factor listener)))

(define gd-optimize
  (make-gd-optimize enforce-relaxation-filtered
                    zero-gradient-condition))

(define (stabilization-listener eps max-count [printing-listener void])
  (let ((prev-g #f)
        (stable-ratio -1)
        (count 0))
    (lambda (x shift x-new g G)
      (printing-listener x shift x-new g G)
      (when prev-g

```

```

      (let ((new-ratio (/ (p-vector-norm g) (p-vector-norm prev-g))))
        (if (< (abs (- new-ratio stable-ratio)) eps)
            (set! count (add1 count))
            (begin
              (set! count 0)
              (set! stable-ratio new-ratio))))))
    (set! prev-g g)
    (when (>= count max-count) (cons 'stabilized (cons stable-ratio x)))))

;; s = 1.3 * sqrt(2 / (1 - mu))
(define (make-relch-degree gradient-ratio)
  (define (closest-even x)
    (let ((n (round x)))
      (- n (modulo n 2))))
  (closest-even
   (* 1.3 (sqrt (/ 2 (abs (- 1 gradient-ratio)))))))

(define (gdrelch-optimize f x-start iterations eps step-factor [listener void])
  (let ((sgd-result (sgd-optimize f x-start iterations eps step-factor
                                   (stabilization-listener eps 5 listener))))
    (if (cons? sgd-result)
        (let ((relch-degree (make-relch-degree (car (cdr sgd-result))))
              (x-start (cdr (cdr sgd-result))))
          (relch-optimize f x-start iterations eps relch-degree listener))
        sgd-result)))

```

Б. Информация о документе

Данный документ был подготовлен с использованием `LATEX`. В качестве реализации языка Scheme использовалась `PLT Scheme`. Иллюстрации созданы с помощью пакета `pgfplots` и `gnuplot`.

Автоматизация процесса сборки обеспечивалась утилитами `GNU Make` и `texdepend`.

Представленная работа выполнена в рамках программы пятого семестра обучения по специальности «Вычислительная математика и математическая физика» в МГТУ им. Н. Э. Баумана.

Дата компиляции настоящего документа: 22 марта 2010 г.

В. Список литературы

- [1] Фихтенгольц Г. М. Курс дифференциального и интегрального исчисления. — 8 изд. — М.: Физматлит, 2003. — Т. 2. — С. 329–336.
- [2] ТАХА Х. Введение в исследование операций. — М.: Вильямс, 2005. — 912 с.
- [3] ЧЕРНОРУЦКИЙ И. Г. Методы оптимизации в теории управления. — СПб.: Питер, 2004. — 256 с.
- [4] Гилл Ф., МЮРРЕЙ У., РАЙТ М. Практическая оптимизация. — М.: Мир, 1985. — С. 289–293.
- [5] АЛЕКСЕЕВ В. М., ТИХОМИРОВ В. М., ФОМИН С. В. Оптимальное управление. — М.: Физматлит, 2005. — С. 41–45.
- [6] ХИММЕЛЬБЛАУ Д. Прикладное нелинейное программирование. — М.: Мир, 1975.
- [7] КОЛМОГОРОВ А. Н., ФОМИН С. В. Элементы теории функций и функционального анализа. — М.: Наука, 1972. — С. 123.
- [8] ИЗМАИЛОВ А. Ф., СОЛОДОВ М. В. Численные методы оптимизации. — М.: Физматлит, 2005. — С. 38–44.
- [9] ПОЛОВИНКИН Е. С., БАЛАШОВ М. В. Элементы выпуклого и сильно выпуклого анализа. — М.: Физматлит, 2004. — С. 24, 52, 211–215.