

Министерство образования и науки Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ
И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

ФАКУЛЬТЕТ ДИСТАНЦИОННОГО ОБУЧЕНИЯ (ФДО)

А. А. Мицель, А. А. Шелестов, В. В. Романенко

МЕТОДЫ ОПТИМИЗАЦИИ

Учебное пособие

Томск
2017

УДК [517.977.5:681.51.01 + 519.75](075.8)

ББК 22.183я73

М 701

Рецензенты:

А. Ю. Трифонов, д-р физ.-мат. наук, профессор, зав. кафедрой высшей математики и математической физики Национального исследовательского Томского политехнического университета;

Л. И. Бабак, д-р техн. наук, профессор кафедры компьютерных систем в управлении и проектировании Томского государственного университета систем управления и радиоэлектроники

Мицель А. А.

М 701 Методы оптимизации : учебное пособие / А. А. Мицель, А. А. Шелестов, В. В. Романенко. – Томск : ФДО, ТУСУР, 2017. – 198 с.

Учебное пособие включает шесть основных разделов лекционного курса по дисциплине «Методы оптимизации». В первом разделе дан анализ экстремальных задач. Во втором и третьем разделах рассмотрены методы безусловной оптимизации функции одной переменной и функций многих переменных. Четвертый, пятый и шестой разделы посвящены методам условной оптимизации для решения задач линейного и нелинейного программирования.

Для студентов специальностей 09.03.01, 09.03.03 и 01.03.02, а также аспирантов, преподавателей и инженеров любых специальностей, связанных с решением оптимизационных задач.

© Мицель А. А., Шелестов А. А.,
Романенко В. В., 2017

© Оформление.
ФДО, ТУСУР, 2017

Оглавление

Введение	6
1 Анализ экстремальных задач	11
1.1 Основные понятия и определения.....	11
1.2 Постановка и классификация задач оптимизации.....	21
1.3 Необходимые и достаточные условия существования экстремума	23
1.4 Характеристики алгоритмов оптимизации	26
1.5 Критерии останова.....	29
1.6 Численная аппроксимация градиентов.....	30
1.7 Классы алгоритмов оптимизации.....	31
2 Методы минимизации функции одной переменной	33
2.1 Классификация методов.....	33
2.2 Методы исключения интервалов.....	33
2.2.1 Метод равномерного поиска.....	34
2.2.2 Метод деления отрезка пополам (метод дихотомии)	36
2.2.3 Метод Фибоначчи.....	38
2.2.4 Метод золотого сечения.....	43
2.3 Полиномиальная аппроксимация и методы точечного оценивания	46
2.3.1 Квадратичная аппроксимация	46
2.3.2 Метод Пауэлла	47
2.4 Методы с использованием производных.....	49
2.4.1 Метод Ньютона – Рафсона.....	49
2.4.2 Другие итерационные методы поиска нулей функции.....	52
2.4.3 Метод средней точки (поиск Больцано).....	54
2.5 Метод поиска с использованием кубичной аппроксимации.....	56
2.6 Сравнение методов	58
3 Методы поиска экстремума функции многих переменных	65
3.1 Классификация методов.....	65
3.2 Методы прямого поиска.....	66
3.2.1 Симплексный метод	66
3.2.2 Метод поиска Хука – Дживса.....	71
3.2.3 Метод сопряженных направлений Пауэлла.....	74
3.3 Градиентные методы и методы второго порядка	80
3.3.1 Метод наискорейшего спуска (метод Коши).....	82
3.3.2 Метод Ньютона	84

3.3.3 Модифицированный метод Ньютона	85
3.3.4 Метод Марквардта	86
3.3.5 Методы сопряженных градиентов	87
3.3.6 Квазиньютоновские методы (методы с переменной метрикой).....	92
3.3.7 Обобщенный градиентный метод	102
3.4 Сравнение методов	103
4 Линейное программирование	107
4.1 Классификация методов	107
4.2 Разработка моделей линейного программирования.....	108
4.3 Формы записи задач линейного программирования.....	109
4.4 Основные определения ЛП	112
4.5 Поиск начального базиса	115
4.5.1 Метод Жордана – Гаусса.....	115
4.5.2 Метод искусственного базиса	115
4.6 Графическое решение ЗЛП	116
4.7 Основы симплекс-метода	121
4.8 Целочисленное программирование.....	126
4.8.1 Графический метод решения ЗЦП	127
4.8.2 Метод Гомори	128
5 Транспортная задача	132
5.1 Классификация методов	132
5.2 Понятия транспортной задачи и транспортной модели.....	133
5.3 Первоначальное закрепление потребителей за поставщиками.....	135
5.4 Решение транспортной задачи симплекс-методом	140
5.5 Решение транспортной задачи методом потенциалов	141
5.6 Задача о назначениях	143
5.7 Венгерский метод решения задачи о назначениях	145
6 Нелинейное программирование.....	147
6.1 Классификация методов	147
6.2 Задачи с ограничениями в виде равенств	149
6.2.1 Метод замены переменных.....	149
6.2.2 Метод множителей Лагранжа.....	150
6.2.3 Экономическая интерпретация множителей Лагранжа.....	152
6.3 Необходимые и достаточные условия оптимальности	154

6.3.1 Необходимые и достаточные условия оптимальности задач с ограничениями общего вида	155
6.3.2 Необходимые и достаточные условия оптимальности второго порядка	157
6.4 Методы штрафов.....	158
6.4.1 Понятие штрафных функций.....	160
6.4.2 Квадратичный штраф	161
6.4.3 Логарифмический штраф.....	162
6.4.4 Штраф типа обратной функции	163
6.4.5 Штраф типа квадрата срезки	164
6.4.6 Выбор штрафного параметра	165
6.4.7 Обобщенный алгоритм.....	166
6.5 Методы, основанные на линеаризации.....	167
6.5.1 Базовый метод линеаризации	167
6.5.2 Алгоритм Франка – Вульфа.....	169
6.5.3 Метод допустимых направлений Зойтендейка.....	172
6.5.4 Метод условного градиента.....	176
6.6 Метод проекции градиента	181
6.6.1 Случай линейных ограничений.....	181
6.6.2 Случай нелинейных ограничений.....	188
Заключение	192
Литература	193
Список условных обозначений и сокращений	195

Введение

В наиболее общем смысле теория оптимизации представляет собой совокупность фундаментальных математических результатов и численных методов, ориентированных на нахождение и идентификацию наилучших вариантов из множества альтернатив и позволяющих избежать полного перебора и оценивания возможных вариантов.

В настоящее время теория оптимизации, успешному применению которой способствует бурный прогресс в развитии средств вычислительной техники, вносит заметный вклад в ускорение научно-технического прогресса. Трудно назвать такую отрасль инженерной деятельности, где бы ни возникали задачи оптимизационного характера.

Важность и ценность теории оптимизации заключается в том, что она дает адекватные понятийные рамки для анализа и решения многочисленных задач [1]:

- в исследовании операций: оптимизация технико-экономических систем, транспортные задачи, управление запасами и т. д.;
- в численном анализе: аппроксимация, регрессия, решение линейных и нелинейных систем, численные методы, включая методы конечных элементов и т. д.;
- в автоматике: распознавание образов, оптимальное управление, фильтрация, управление производством, робототехника и т. д.;
- в математической экономике: решение больших макроэкономических моделей, моделей предпринимательства, теория принятия решений и теория игр.

Постановка каждой задачи оптимизации (ЗО) включает два объекта: множество допустимых решений и целевую функцию (функционал), которую следует минимизировать или максимизировать на указанном множестве. С этой общей точки зрения и рассматриваются различные классы экстремальных задач, составляющие предмет линейного, нелинейного, динамического программирования, вариационного исчисления и теории оптимального управления. Обычно наши действия в условиях неоднозначности выбора определяются некоторой целью, которую мы стремимся достичь наилучшим образом. Тем самым человеческая деятельность связана с постоянным (сознательным или бессознательным) решением оптимизационных задач. Об этом свидетельствуют

многочисленные древние легенды (о финикийской царице Дидоне, о возникновении г. Карфагена и др.), связанные с решением различных изопериметрических задач.

Можно было бы думать, что подобная распространенность ЗО давно должна была найти свое отражение в математике. Однако в действительности до середины XX столетия задачи на экстремум (extr) рассматривались в математике лишь эпизодически, развитая теория и методы решения подобных задач были созданы сравнительно недавно.

Наиболее простая задача безусловной минимизации функции многих переменных привлекла внимание математиков во времена, когда закладывались основы математического анализа. Она во многом стимулировала создание дифференциального исчисления, а необходимое условие существования extr (равенство градиента нулю), полученное П. Ферма в 1629 г., явилось одним из первых крупнейших результатов анализа. Позже в работах И. Ньютона и Г. В. Лейбница были по существу сформулированы условия экстремума 2-го порядка (т. е. в терминах вторых производных) для этой задачи.

Другой класс задач на экстремум, традиционно рассматривавшийся в математике, – это задачи вариационного исчисления. Следы интереса к ним можно найти и в античной математике (разного рода изопериметрические проблемы), однако подлинное рождение вариационного исчисления произошло в конце XVIII в., когда И. Бернулли сформулировал знаменитую задачу о брахистохроне (гр. «брахисто» – кратчайший, «хронос» – время).



..... Пример

Задача о брахистохроне. В плоскости (x, y) заданы две точки $A(0, 0)$ и $B(x_1, y_1)$, не лежащие на одной вертикальной прямой (рис. 1). Необходимо найти такую линию, соединяющую эти точки, при движении вдоль которой под действием силы тяжести P тяжелый материальный шарик должен достичь точки B за кратчайшее время. Силами трения пренебрегаем, начальная скорость v_A в точке A равна нулю (ответ: траекторией движения шарика является циклоида; решение задачи будет приведено ниже).

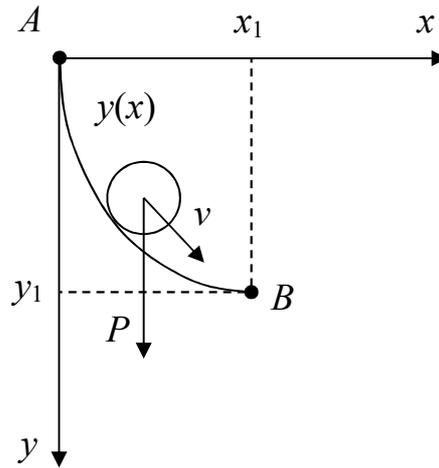


Рис. 1 – Задача о брахистохроне

.....

Условия экстремума 1-го порядка в вариационном исчислении были получены Л. Эйлером (уравнения Эйлера), а 2-го порядка – А. М. Лежандром и К. Г. Якоби. Важный вопрос о существовании решения в вариационном исчислении был впервые поставлен К. Вейерштрассом во второй половине XIX в.

Первые работы по экстремальным задачам при наличии ограничений общего вида (в виде равенств и неравенств) относятся к концу 30-х – началу 40-х гг. XX в. Здесь необходимо упомянуть имена американских ученых Чикагской школы: Г. Блисс, О. Больца, Е. Макшейн, Л. М. Грейвс и др.

Независимо от исследований американских ученых оптимизационная тематика развивалась и в СССР. Пионером в этой области был Л. В. Канторович, опубликовавший в 1939 г. книгу, содержащую математические постановки ряда экономических задач. Но работы Канторовича, как и результаты американских ученых, не привлекли внимание математиков и остались, по существу, незамеченными.

Время для этих задач настало несколько позже: в конце 40-х гг. XX в. Под влиянием прикладной тематики, которой приходилось заниматься в годы Второй мировой войны (проблема снабжения военных плацдармов войск второго фронта в Европе и Африке), Дж. Данциг в США стал изучать задачи минимизации при линейных ограничениях, получившие название задач линейного программирования (ЗЛП). Под влиянием работ Дж. фон Неймана по теории игр, Дж. Данциг, Д. Гейл, Г. Кун и А. Таккер создали теорию двойственности в линейном программировании (ЛП) – специфическую формулировку условий существования экстремума.

Существенный прогресс в развитии теории оптимизации произошел при изучении так называемых задач оптимального управления. Необходимые условия оптимальности для этих задач были сформулированы и доказаны Л. С. Понтрягиным, В. Г. Болтянским и Р. В. Гамкрелидзе в 1956–1958 гг. в форме «принципа максимума». В иной форме условия оптимальности для подобных задач были получены Беллманом на основе идей динамического программирования.

С необходимостью решения ЗО столкнулись и специалисты по теории автоматического управления (ТАУ). В трудах В. В. Казакевича, А. А. Фельдбаума, А. А. Первозванского в 50-х гг. XX столетия была создана теория экстремального регулирования и предложены специальные математические модели объектов, действующих в реальном масштабе времени.

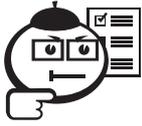
К середине 60-х гг. XX в. в рамках вычислительной математики сложилось самостоятельное направление, связанное с численными методами оптимизации (МО). С тех пор непрерывно шло интенсивное развитие этого направления как вширь (разработка новых методов, исследование новых классов задач и т. д.), так и вглубь (выработка единого аппарата для анализа сходимости и скорости сходимости, классификации методов). В настоящее время эта область вычислительной математики может считаться окончательно сформировавшейся, хотя существуют проблемы построения эффективных методов для некоторых специальных типов задач и отдельных специфических ситуаций. Так, например, для составления маршрута доставки тяжелой баллистической ракеты-носителя с завода в г. Сиэтле на северо-западе США до космодрома на мысе Каннаверал (полуостров Флорида) был специально разработан новый оптимизационный алгоритм, учитывающий массу, габариты груза, загруженность автомагистралей, грузоподъемность мостов, размеры туннелей и т. д.

Соглашения, принятые в учебном пособии

Для улучшения восприятия материала в данном учебном пособии используются пиктограммы и специальное выделение важной информации.



.....
Эта пиктограмма означает определение или новое понятие.



Эта пиктограмма означает задание. Здесь автор может дать указания для выполнения самостоятельной работы или упражнений, сослаться на дополнительные материалы.



В блоке «На заметку» автор может указать дополнительные сведения или другой взгляд на изучаемый предмет, чтобы помочь читателю лучше понять основные идеи.



Эта пиктограмма означает теорему.



Эта пиктограмма означает аксиому.



Эта пиктограмма означает доказательство.



Пример

Эта пиктограмма означает пример. В данном блоке автор может привести практический пример для пояснения и разбора основных моментов, отраженных в теоретическом материале.



Выводы

Эта пиктограмма означает выводы. Здесь автор подводит итоги, обобщает изложенный материал или проводит анализ.



Контрольные вопросы по главе

1 Анализ экстремальных задач

1.1 Основные понятия и определения

Математическое программирование в самом общем виде можно определить как задачу оптимизации с ограничениями в пространстве R^n [2]:

$$\begin{aligned} f(x) &\rightarrow \min_x, \\ g_j(x) &\geq 0, \quad j=1,2,\dots,J; \\ h_k(x) &= 0, \quad k=1,2,\dots,K; \\ x &\in D \subseteq R^n. \end{aligned} \tag{1.1}$$

Вектор $x \in D$ имеет компоненты x_1, x_2, \dots, x_n , которые являются неизвестными задачи (1.1). В дальнейшем, если функция $f(x)$ минимизируется по всем компонентам аргумента x , будем писать просто

$$f(x) \rightarrow \min.$$

Запись

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

означает поиск максимума целевой функции (ЦФ). Однако в силу очевидного равенства

$$\max f(x) = -\min(-f(x)), \tag{1.2}$$

не теряя общности, можно рассматривать только задачи на минимизацию. То есть задача максимизации функции сводится к задаче поиска минимума функции $F(x) = -f(x)$.

Запись

$$f(x) \rightarrow \text{extr}$$

означает поиск любого экстремума ЦФ.



.....
 Функция $f(x)$ называется **целевой функцией** (ЦФ, функцией качества, критерием оптимальности), а множество условий $g_j(x)$, $h_k(x)$ и $x \in D$ – **ограничениями задачи**.

В качестве D часто выступает выпуклое множество.

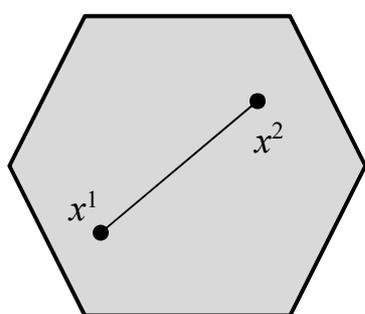


Выпуклым множеством $D \subseteq R^n$ называют такое множество, которое вместе с любыми двумя точками $x^1, x^2 \in D$ и для произвольного числа $\mu \in [0, 1]$ содержит также все точки x , удовлетворяющие соотношению

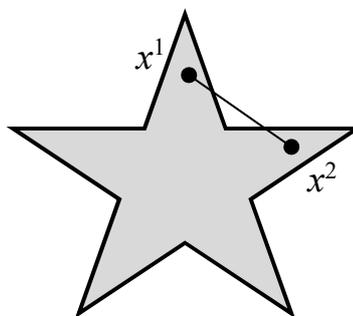
$$x = \mu \cdot x^1 + (1 - \mu) \cdot x^2,$$

т. е. выпуклое множество должно содержать отрезок, соединяющий любые две точки такого множества [5–6].

Примеры выпуклых множеств – куб, квадрат, круг, шар и др. (рис. 1.1).



Выпуклое множество



Невыпуклое множество

Рис. 1.1 – Выпуклость множеств



Ограничения задачи

$$g_j(x) \geq 0, \quad j = 1, 2, \dots, J;$$

$$h_k(x) = 0, \quad k = 1, 2, \dots, K;$$

$$x \in D \subseteq R^n$$

образуют **область допустимых решений (ОДР)** $S \subseteq R^n$. Точка $x \in S$ тогда и только тогда, когда она удовлетворяет всем ограничениям задачи.

Решением задачи (1.1) называют любой вектор x , удовлетворяющий ограничениям, т. е. $x \in S$.

Локальным экстремумом ЦФ $f(x)$ называют вектор $\bar{x} \in S$, если имеет место неравенство [6]

$$f(\bar{x}) \leq f(x) \quad \{x : |x - \bar{x}| < \varepsilon, \varepsilon > 0\}$$

или

$$f(\bar{x}) \geq f(x) \{x: |x - \bar{x}| < \varepsilon, \varepsilon > 0\}.$$

В первом случае имеем локальный минимум, во втором – локальный максимум.

Оптимальным решением или **глобальным экстремумом** задачи (1.1) называют вектор x^* , минимизирующий значение $f(x)$ на множестве всех решений (глобальный минимум):

$$f(x^*) \leq f(x),$$

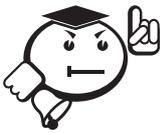
или максимизирующий значение $f(x)$ на множестве всех решений (глобальный максимум):

$$f(x^*) \geq f(x)$$

для всех $x \in S$.

.....

В дальнейшем, в силу (1.2), говоря об экстремуме или оптимуме, будем подразумевать минимум, если не оговаривается иное.



.....

Всякая точка глобального минимума ЦФ $f(x)$ является одновременно и точкой локального минимума этой функции, а точка глобального максимума ЦФ является одновременно и точкой локального максимума. Обратное утверждение, вообще говоря, неверно.

.....

Точность. Характеристикой точности полученного решения может служить величина абсолютного отклонения значения минимизируемой функции, достигнутого в точке $x^k \in S$, от точного значения ее минимума на множестве S :

$$\delta(x) = |f(x^k) - f(x^*)|, \quad x \in S, \quad (1.3)$$

где $f(x^*)$ – либо точное значение минимума ЦФ ($f(x^*) = \min f(x)$), либо полученное «точным» алгоритмом.

Ясно, что чем меньше неотрицательная величина δ , тем точнее полученное решение. Недостатком использования абсолютной погрешности является то

обстоятельство, что она меняется при умножении ЦФ на положительную константу α : $f(x) \rightarrow \alpha \cdot f(x)$.

Целесообразнее использовать следующую оценку точности:

$$c = \left| \frac{f(x^k) - f(x^*)}{f(x^*)} \right| \cdot 100\%. \quad (1.4)$$

Алгоритм называется ε -приближенным, если выполняется неравенство $c \leq \varepsilon$.

Унимодальные функции. Многие методы оптимизации применимы только тогда, когда целевая функция $f(x)$ является унимодальной, т. е. любой локальный минимум ЦФ одновременно является и глобальным [6].

Множество функций, унимодальных на отрезке $[a, b]$, обозначим $Q[a, b]$.

Существование локальных минимумов $f(x)$, отличных от абсолютного, почти всегда затрудняет поиск точки экстремума $x^* \in S$, поэтому многие приближенные методы применимы тогда, когда любой локальный минимум функции $f(x)$ является одновременно и глобальным. Одним из классов функций, удовлетворяющих этому условию, составляют унимодальные функции.



.....
 Функция $f(x)$ называется **унимодальной** на отрезке $[a, b]$, если она непрерывна на $[a, b]$ и существуют числа x_1 и x_2 , причем $a \leq x_1 \leq x_2 \leq b$, такие, что:

- если $a \leq x_1$, то на отрезке $[a, x_1]$ функция $f(x)$ монотонно убывает;
 - если $x_2 \leq b$, то на отрезке $[x_2, b]$ функция $f(x)$ монотонно возрастает;
 - при $x \in [x_1, x_2]$ имеем $f(x) = f^* = \min_{x \in [a, b]} f(x)$.
-

Отметим, что возможно вырождение в точку одного или двух из отрезков $[a, x_1]$, $[x_1, x_2]$ и $[x_2, b]$. Некоторые варианты расположения и вырождения в точку отрезков монотонности и постоянства унимодальных функций представлены на рисунке 1.2.

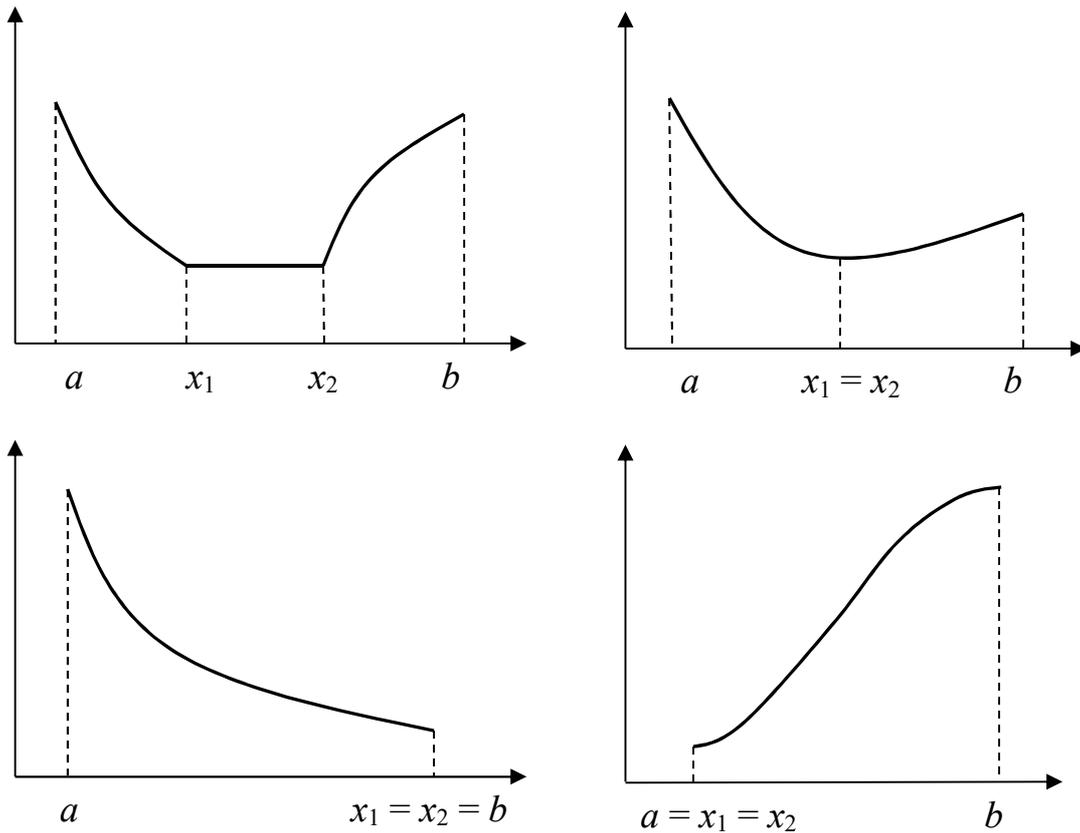


Рис. 1.2 – Примеры унимодальных функций

Очевидно, что в некоторых вырожденных случаях ЦФ $f(x)$ не является дифференцируемой на отрезке $[a, b]$.

Для проверки унимодальности функции $f(x)$ на практике обычно используют следующие критерии:

- если функция $f(x)$ дифференцируема на отрезке $[a, b]$ и производная $f'(x)$ не убывает на этом отрезке, то $f(x) \in Q[a, b]$;
- если функция $f(x)$ дважды дифференцируема на отрезке $[a, b]$ и вторая производная $f''(x) \geq 0$ при $x \in [a, b]$, то $f(x) \in Q[a, b]$.

Если эти критерии не выполняются, то функция $f(x)$ является мульти-модальной или многоэкстремальной.

Выпуклые и вогнутые функции. Функция $f(x)$ выпукла («выпукла вниз») на $[a, b]$, если справедливо неравенство Йенссена

$$f[\mu x^1 + (1 - \mu)x^2] \leq \mu f(x^1) + (1 - \mu)f(x^2) \quad (1.5)$$

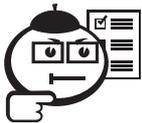
для произвольных $x^1, x^2 \in [a, b]$ и $\mu \in [0, 1]$ (рис. 1.3, а).

Геометрически это означает, что все точки функции $f(x)$ для всех $x \in [a, b]$ лежат ниже прямой $\mu f(x^1) + (1 - \mu) f(x^2)$ для любых точек $x^1, x^2 \in [a, b]$.

На практике обычно используют следующие критерии:

- 1) *одномерный случай*: дважды дифференцируемая функция $f(x)$ выпукла на $[a, b]$, если вторая производная $f''(x) \geq 0$ при всех $x \in [a, b]$;
- 2) *многомерный случай* (критерий Сильвестра): если функция $f(x)$ дважды дифференцируема на выпуклом множестве $D \subseteq R^n$ и все угловые миноры матрицы Гессе – матрицы вторых производных $H_f(x) = \partial^2 f(x) / (\partial x_i \partial x_j)$ – положительны при $x \in D$, то функция $f(x)$ выпукла на множестве D .

Если $F(x) = -f(x)$, и при этом функция $f(x)$ является выпуклой, то функция $F(x)$ будет вогнутой («выпуклой вверх») (рис. 1.3, б). Обратное утверждение также верно. Соответственно, для одномерного случая вторая производная вогнутой функции $f''(x) \leq 0$. Для многомерного случая учитываем, что при изменении знака функции меняются также знаки всех элементов матрицы Гессе: $H_F(x) = -H_f(x)$.



.....
 Какими будут знаки угловых миноров вогнутой функции?

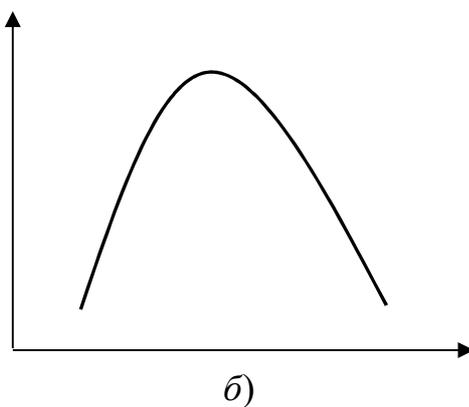
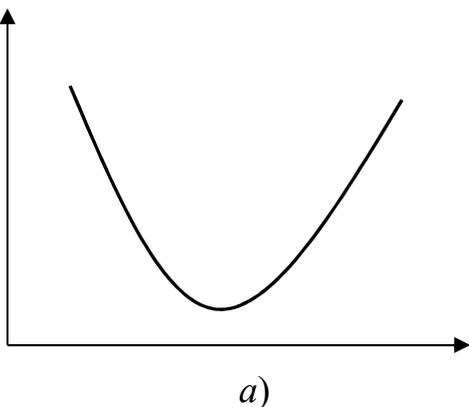


Рис. 1.3 – Пример (a) выпуклой и (б) вогнутой функции



.....
 Функция называется **строго выпуклой**, если неравенство (1.5) является строгим для всех $\mu \in (0,1)$ и $x^1 \neq x^2$.

Квадратичные функции. Во многих задачах оптимизации рассматриваются функции вида

$$f(x) = \sum_{i=1}^n \sum_{j=1}^n x_i a_{ij} x_j + \sum_{j=1}^n b_j x_j + c,$$

или, в матричном виде,

$$f(x) = \frac{1}{2} x^T A x + b^T x + c, \quad (1.6)$$

где x и b есть вектор-столбцы размерности n ;

A – симметричная матрица ($n \times n$);

c – константа.

Градиент и матрица Гессе функции (1.6) равны, соответственно,

$$\text{grad } f(x) = Ax + b,$$

$$H_f(x) = A.$$

Таким образом, для того чтобы функция (1.6) была выпуклой в R^n , достаточно, чтобы матрица A была положительно определена.



.....
Критерии определенности матриц (теорема Сильвестра) [9]. **Положительная определенность:**

- все диагональные элементы матрицы должны быть положительными;
- все ведущие главные определители должны быть положительными.

Положительная полуопределенность:

- все диагональные элементы неотрицательны;
- все главные определители неотрицательны.

Главный определитель – это определитель главного минора.

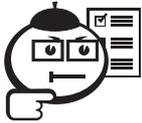
Для положительной определенности квадратичной формы – матрицы A – необходимо и достаточно, чтобы все угловые миноры матрицы A были положительными, т. е.

$$\Delta_1 = a_{11} > 0, \Delta_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0, \dots, \Delta_n = |A| > 0.$$

Соответственно, если у положительно определенной матрицы изменить знаки всех элементов ($A' = -A$), то она станет отрицательно определенной. Если это сделать для положительно полуопределенной матрицы, то она станет отрицательно полуопределенной.

Для отрицательной определенности матрицы A необходимо и достаточно, чтобы знаки угловых миноров матрицы A чередовались, начиная с отрицательного, т. е.

$$\Delta_1 = a_{11} < 0, \Delta_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0, \Delta_3 = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} < 0, \dots, \Delta_n = (-1)^n |A| > 0.$$



.....
 Каким будет критерий отрицательной полуопределенности матрицы?



.....
Стационарной точкой функции $f(x)$, где $x \in R^n$, называется такая точка \bar{x} , в которой функция непрерывна, а ее градиент обращается в ноль:

$$\text{grad } f(x)|_{x=\bar{x}} = 0.$$

Критической точкой функции $f(x)$, или **подозрительной на экстремум точкой**, называется точка \bar{x} , которая либо является стационарной, либо градиент в ней

$$\text{grad } f(x)|_{x=\bar{x}}$$

не существует (т. е. имеет разрыв). Другими словами, это точка, в которой функция непрерывна, а градиент равен нулю или имеет разрыв.

Для функций скалярного аргумента, соответственно, исследуют производную, а не градиент.



.....
 Если критическая точка не соответствует локальному экстремуму (минимуму или максимуму), то она является **седловой точкой**.



..... **Пример 1.1**

Дана функция $f(x) = -x_1^2 + x_2^2 + x_1x_2 + x_1 + x_2$. Классифицировать точку $x = (0, 2; -0, 6)^T$.

Решение.

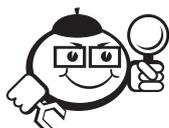
Вычислим градиент функции в точке $x = (0, 2; -0, 6)^T$:

$$\text{grad } f(x) = \begin{pmatrix} -2x_1 + x_2 + 1 \\ x_1 + 2x_2 + 1 \end{pmatrix} \Big|_{x = \begin{pmatrix} 0, 2 \\ -0, 6 \end{pmatrix}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Таким образом, точка $x = (0, 2; -0, 6)^T$ является подозрительной на экстремум. Для того чтобы идентифицировать эту точку, т. е. определить, является ли она точкой максимума (минимума) или седловой точкой, вычислим матрицу Гессе:

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ 1 & 2 \end{pmatrix}.$$

Минор первого порядка $\Delta_1 = -2 < 0$, минор второго порядка $\Delta_2 = -2 \cdot 2 - 1 \cdot 1 = -5 < 0$. Следовательно, матрица Гессе не является ни положительно определенной, ни отрицательно определенной. Поэтому точка $x = (0, 2; -0, 6)^T$ является седловой точкой.



..... **Пример 1.2**

Дана функция $f(x) = x_1^2 + x_2^2 + x_1x_2 + x_1 + x_2$. Классифицировать точку $x = (-1/3; -1/3)^T$.

Решение.

Вычислим градиент функции в точке $x = (-1/3; -1/3)^T$:

$$\text{grad } f(x) = \begin{pmatrix} 2x_1 + x_2 + 1 \\ x_1 + 2x_2 + 1 \end{pmatrix} \Big|_{x = \begin{pmatrix} -1/3 \\ -1/3 \end{pmatrix}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Таким образом, точка $x = (-1/3; -1/3)^T$ является стационарной. Для того чтобы идентифицировать эту точку, вычислим матрицу Гессе:

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}.$$

Минор первого порядка $\Delta_1 = 2 > 0$, минор второго порядка $\Delta_2 = 2 \cdot 2 - 1 \cdot 1 = 3 > 0$. Следовательно, матрица Гессе является положительно определенной. Поэтому точка $x = (-1/3; -1/3)^T$ является точкой минимума.

**Пример 1.3**

Дана функция $f(x) = -x_1^2 - x_2^2 + x_1x_2 + x_1 + x_2$. Классифицировать точку $x = (1; 1)^T$.

Решение.

Градиент функции в точке $x = (1; 1)^T$ равен

$$\text{grad } f(x) = \begin{pmatrix} 2x_1 + x_2 + 1 \\ x_1 + 2x_2 + 1 \end{pmatrix} \Big|_{x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

т. е. точка $x = (1; 1)^T$ является стационарной. Идентифицируем эту точку:

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}.$$

Имеем $\Delta_1 = -2 < 0$, $\Delta_2 = (-2) \cdot (-2) - 1 \cdot 1 = 3 > 0$. Следовательно, матрица Гессе является отрицательно определенной. Поэтому точка $x = (1; 1)^T$ является точкой максимума.

.....

1.2 Постановка и классификация задач оптимизации

Для того чтобы использовать математические результаты и численные методы теории оптимизации для решения конкретных инженерных задач, необходимо:

- 1) установить свойства границы подлежащей оптимизации инженерной системы или объекта;
- 2) построить математическую модель системы;
- 3) составить целевую функцию. Иногда удается подставить в целевую функцию математическую модель и получить явную зависимость ЦФ от управляющих воздействий, т. е. возможных стратегий управления системой. В остальных случаях математическая модель выступает в роли ограничений на управление;
- 4) определить критерий оптимальности, как правило, требование экстремума ЦФ по управляющим воздействиям при наличии ограничений;
- 5) выбрать или построить оптимизационный алгоритм и решить экстремальную задачу.

Корректная постановка задачи служит ключом к успеху оптимизационного исследования. Искусство постановки задачи постигается в практической деятельности на примерах успешно реализованных алгоритмов и основывается на четком представлении преимуществ, недостатков и специфических особенностей различных методов оптимизации.

Значительный интерес представляет деление задач оптимизации по виду целевой функции и ограничений, поскольку специфические особенности тех или иных задач играют важную роль при разработке методов и их решений [1, 14].

Задача одномерной оптимизации без ограничений (безусловная оптимизация):

$$f(x) \rightarrow \min, \quad x \in R^1, \quad (1.7)$$

ограничения отсутствуют, т. е. в задаче (1.1)

$$K = J = 0,$$

$$x \in D \equiv R^1 \text{ или } x \in (-\infty, +\infty).$$

Задача многомерной оптимизации без ограничений (безусловная оптимизация):

$$\begin{cases} f(x) \rightarrow \min, x \in R^n, \\ K = J = 0, \\ x \in D \equiv R^n \text{ или } x_i \in (-\infty, +\infty), i = 1, 2, \dots, n. \end{cases} \quad (1.8)$$

Задача оптимизации с ограничениями (условная оптимизация):

$$\begin{cases} f(x) \rightarrow \min, \\ g_j(x) \geq 0, j = 1, 2, \dots, J, \\ h_k(x) = 0, k = 1, 2, \dots, K, \\ a_i \leq x_i \leq b_i, i = 1, 2, \dots, n, \end{cases} \quad (1.9)$$

т. е. компоненты вектора x ограничены сверху и снизу.

Задача условной оптимизации с линейными ограничениями. Функции-ограничения $g_j(x)$ и $h_k(x)$ в задаче (1.9) являются линейными, а целевая функция может быть либо линейной, либо нелинейной.

Задача линейного программирования (ЗЛП):

$$\begin{cases} f(x) = \sum_{j=1}^n c_j x_j \rightarrow \min, \\ \sum_{j=1}^n a_{ij} x_j \leq b_i, i = 1, 2, \dots, m, \\ x_j \geq 0, j = 1, 2, \dots, n, \end{cases} \quad (1.10)$$

т. е. и целевая функция, и ограничения являются линейными функциями переменных x_j .

Задача целочисленного программирования (ЗЦП). В задаче ЛП (1.10) компоненты вектора x принимают только целые значения:

$$x_j \in Z, j = 1, 2, \dots, n.$$

Задача нелинейного программирования с линейными ограничениями. Целевая функция $f(x)$ – нелинейная, а все ограничения – линейные функции.

Оптимизационные задачи такого рода можно классифицировать на основе структурных особенностей нелинейных целевых функций.

Задача квадратичного программирования. Найти минимум $f(x)$, где целевая функция $f(x)$ является квадратичной:

$$\begin{cases} f(x) = \frac{1}{2}x^T Qx + c^T x + d \rightarrow \min, \\ \sum_{j=1}^n a_{ij}x_j \leq b_i, \quad i = 1, 2, \dots, m, \\ x_j \geq 0, \quad j = 1, 2, \dots, n. \end{cases} \quad (1.11)$$

Задача дробно-линейного программирования. Целевая функция есть отношение линейных функций:

$$f(x) = \frac{\sum_{j=1}^n c_j x_j + c_0}{\sum_{j=1}^n d_j x_j + d_0} \rightarrow \min. \quad (1.12)$$

Здесь приведены лишь некоторые классы ЗО. Остальные описаны в соответствующих разделах учебного пособия.

1.3 Необходимые и достаточные условия существования экстремума

1. Скалярный случай ($x \in R^1$).

Сформулируем *необходимое условие экстремума* функции скалярного аргумента. Если в точке x^* имеется экстремум, то эта точка является критической, т. е. $f'(x^*) = 0$ или $f'(x^*)$ не существует.

Необходимое условие перегиба функции скалярного аргумента в точке \bar{x} – вторая производная $f''(\bar{x}) = 0$ или $f''(\bar{x})$ не существует.

Первое достаточное условие экстремума. Пусть функция непрерывна в некоторой ε -окрестности критической точки x^* . Тогда:

- если $\left\{ f(x) \leq f(x^*) \mid |x - x^*| \leq \varepsilon \right\}$, то в точке x^* имеем локальный максимум;
- если $\left\{ f(x) \geq f(x^*) \mid |x - x^*| \leq \varepsilon \right\}$, то в точке x^* имеем локальный минимум.

Второе достаточное условие экстремума. Пусть функция дифференцируема в некоторой ε -окрестности критической точки x^* . Тогда:

- если $\{f'(x) > 0 \mid x \in [x^* - \varepsilon, x^*]\}$ и $\{f'(x) < 0 \mid x \in (x^*, x^* + \varepsilon]\}$ (т. е. при переходе через точку x^* производная меняет знак с «плюса» на «минус»), то в данной точке функция достигает максимума;
- если $\{f'(x) < 0 \mid x \in [x^* - \varepsilon, x^*]\}$ и $\{f'(x) > 0 \mid x \in (x^*, x^* + \varepsilon]\}$ (т. е. при переходе через точку x^* производная меняет знак с «минуса» на «плюс»), то в данной точке функция достигает минимума.

Обратите внимание, что, согласно достаточным условиям, если функция не существует слева или справа от точки x^* , то такая точка не является экстремумом.

Достаточное условие перегиба. Пусть функция $f(x)$ дважды дифференцируема в некоторой ε -окрестности точки \bar{x} , для которой выполняется необходимое условие перегиба. Тогда, если вторая производная $f''(x)$ при переходе через точку \bar{x} меняет знак, то в данной точке существует перегиб функции $f(x)$.

Приведенные выше критерии универсальны, т. е. их можно применять даже тогда, когда производные функции терпят разрыв. Если же в точке x^* функция $f(x)$ и ее производные непрерывны, то данные критерии значительно упрощаются.

Сформулируем *необходимое и достаточное условие* экстремума и перегиба непрерывной функции скалярного аргумента $f(x)$, имеющей непрерывные производные. Если в точке x^* первые $(n-1)$ производные функции $f(x)$ обращаются в нуль, а производная порядка n отлична от нуля, т. е.

$$f^{(k)} = d^{(k)} f(x)/dx^k \Big|_{x=x^*} = 0, \quad k = 1, 2, \dots, n-1,$$

$$f^{(n)} = d^{(n)} f(x)/dx^n \Big|_{x=x^*} \neq 0,$$

то

- 1) если значение n – нечетное, то x^* – точка перегиба;
- 2) если значение n – четное, то x^* – локальный экстремум.

Причем, если

- $f^{(n)}(x) > 0$, то x^* – точка локального минимума;

- $f^{(n)}(x) < 0$, то x^* – точка локального максимума.

Для выпуклых и унимодальных функций найденный локальный минимум будет являться одновременно и глобальным.



При поиске глобального минимума функции $f(x)$, заданной на отрезке $[a, b]$, необходимо найти все точки локальных минимумов, вычислить в них значения функции, а также вычислить значения функции на границах отрезка $f(a)$ и $f(b)$, затем выбрать среди этих значений наименьшее.

2. Векторный случай ($x \in R^n$).

Сформулируем *необходимое условие экстремума* функции векторного аргумента. Если в точке x^* имеется экстремум, то эта точка является критической, т. е. $\text{grad } f(x^*) = 0$ или $\text{grad } f(x^*)$ не существует.

Первое достаточное условие экстремума. Пусть функция непрерывна в некоторой ε -окрестности критической точки x^* . Тогда:

- если $\{f(x) \leq f(x^*) \mid \|x - x^*\| \leq \varepsilon\}$, то в точке x^* имеем локальный максимум;
- если $\{f(x) \geq f(x^*) \mid \|x - x^*\| \leq \varepsilon\}$, то в точке x^* имеем локальный минимум.

Второе достаточное условие экстремума. Пусть функция дважды дифференцируема в точке x^* . Тогда:

- если матрица Гессе $H_f(x)$ в точке x^* отрицательно определена, то в данной точке функция достигает максимума;
- если матрица Гессе $H_f(x)$ в точке x^* положительно определена, то в данной точке функция достигает минимума.

Если матрица Гессе $H_f(x)$ в точке x^* является неопределенной, то имеем седловую точку. Если матрица $H_f(x)$ в точке x^* положительно или отрицательно полуопределена, то сразу нельзя сделать определенных выводов – это

может быть как экстремум, так и седловая точка. Нужны дополнительные исследования (например, по первому достаточному условию).

Для выпуклых функций найденный локальный минимум будет являться одновременно и глобальным.

3. Минимизация при ограничениях.

Рассматривается задача:

$$\begin{cases} f(x) \rightarrow \min, \\ g_k(x) \leq 0, \quad k = 1, 2, \dots, m, \\ x \in D \equiv R^n. \end{cases} \quad (1.13)$$

Обозначим $L(x, \lambda)$ функцию Лагранжа

$$L(x, \lambda) = f(x) + \sum_{k=1}^m \lambda_k g_k(x), \quad (1.14)$$

где λ_k – неопределенные множители Лагранжа.

Здесь все функции $f(x)$, $g_k(x)$, $L(x, \lambda)$ предполагаются непрерывно дифференцируемыми.



.....
Точка x^ будет глобальным экстремумом задачи (1.14), если в ней выполняются условия Куна – Таккера: существует такое $\lambda^* \geq 0$, что*

$$\begin{aligned} \text{grad } L(x, \lambda^*) \Big|_{x=x^*} &= 0; \\ \lambda_k^* g_k(x^*) &= 0, \quad k = 1, 2, \dots, m. \end{aligned}$$

.....

Пара векторов (x^*, λ^*) образует седловую точку функции Лагранжа, если при всех $x \in D$ и $\lambda \geq 0$ выполняется неравенство

$$L(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L(x, \lambda^*).$$

Таким образом, точка x^* является точкой глобального минимума задачи (1.14), если пара векторов (x^*, λ^*) является седловой точкой функции $L(x, \lambda)$.

1.4 Характеристики алгоритмов оптимизации

Прежде чем приступить к решению конкретной задачи оптимизации, следует выяснить вопрос, какому из методов оптимизации отдать предпочтение

в данном случае. Для этого необходимо уметь сравнивать методы, например, на определенных классах задач.

Очень важна роль *априорных характеристик* методов: трудоемкость вычислений, скорость сходимости, устойчивость метода к ошибкам в вычислениях, чувствительность метода к значениям параметров алгоритма и ряд других [2].

Сходимость алгоритма является наиболее важным вопросом оптимизации. Большинство методов решения ЗО имеют итерационную природу: $x_{i+1} = A(x_i)$, т. е., исходя из некоторой начальной точки x_0 , они порождают последовательность $\{x_0, x_1, \dots, x_i, \dots\}$ на основе алгоритма A , сходящуюся к точке экстремума x^* .



.....

Алгоритм A обладает свойством глобальной сходимости, если для любой начальной точки x_0 последовательность $\{x_k\}$, определяемая выражением $x_k = A(x_{k-1})$, сходится к точке, удовлетворяющей необходимым условиям оптимальности. Данное свойство отражает надежность работы алгоритма.

.....

С практической точки зрения эффективность алгоритма зависит от числа итераций, необходимых для получения приближенного решения x^* с заданной точностью ε .



.....

*Для получения критерия с некоторым абсолютным значением необходимо прибегнуть к другому типу анализа, взяв за объект исследования **асимптотическую сходимость** – поведение последовательности точек $\{x_k\}$ в окрестности предельной точки x^* .*

.....

Это значит, что каждому алгоритму приписывается *индекс эффективности* – *скорость сходимости*. Предположим, что последовательность $\{x_k\}$ сходится к точке x^* .

Линейная сходимость. Выполняется неравенство:

$$\limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \alpha < 1, \quad (1.15)$$

где α – коэффициент сходимости, $\|\cdot\|$ – знак евклидовой нормы в пространстве R^n .

Суперлинейная сходимость:

$$\limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \rightarrow 0. \quad (1.16)$$

Сходимость порядка p . Если существует такое число $p > 1 (2, 3, \dots)$, что

$$\limsup_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} = c < \infty, \quad (1.17)$$

где c – константа, значит, имеем сходимость порядка p :

- $p = 2$ – квадратичная сходимость;
- $p = 3$ – кубическая сходимость;
- $p = n$ – сходимость порядка n .

Исследование скорости сходимости алгоритма позволяет оценить его эффективность и осуществить его сравнение с другими алгоритмами.



.....

Иногда к выражению скорости сходимости последовательности $\{x_k\}$ приходят и при исследовании способа сходимости последовательности $\{f(x_k)\}$ к $f(x^*)$.

Возможны ситуации, когда последовательность $\{x_k\}$, для которой норма $\|x_k - x^*\|$ линейно сходится к нулю, а последовательность $\{f(x_k)\}$ даже не является монотонно убывающей. И наоборот, значения $|f(x_k) - f(x^*)|$ могут линейно сходиться к нулю, когда расстояние $\|x_k - x^*\|$ не является монотонным. В большинстве же случаев оценка скорости сходимости по норме $\|x_k - x^*\|$ *равносильна* оценке разности $|f(x_k) - f(x^*)|$.

.....

Для оценки эффективности выбранных методов можно рекомендовать три характеристики:

- 1) *время*, затраченное на получение решения;
- 2) *точность* решения;
- 3) *чувствительность* к изменению параметра сходимости.

Для одномерных методов оптимизации первые две характеристики можно исследовать на типовых тестовых функциях, например,

$$f(x) = \sin^k(x), \quad \pi \leq x \leq 2\pi,$$

путем варьирования значений показателя степени k на множестве нечетных чисел от 1 до 79. Отметим, что для всех k $\min f(x)$ достигается в точке $x^* = 3\pi/2 \approx 4,71239$, при этом $f(x^*) = -1$. Однако с увеличением k степень гладкости функции, которая обладает узкими впадинами в окрестности x^* , уменьшается. Проверку методов на чувствительность можно осуществить путем варьирования значений параметра сходимости k и параметров алгоритма.

Точность решения можно измерить как относительную (в процентах) ошибку оценивания координаты истинного минимума [4].

1.5 Критерии останова

Приведем несколько наиболее распространенных критериев останова оптимизационных методов [1].

1. Критерии останова по аргументу (здесь ε_x – заданная абсолютная точность, а δ_x – заданная относительная точность по аргументу):

- $|\Delta x| = |x^k - x^{k-1}| \leq \varepsilon_x$ или $|x^k - x^{k-1}| / |x^{k-1}| \leq \delta_x$;
- $\|\Delta x\| = \|x^k - x^{k-1}\| \leq \varepsilon_x$ или $\|x^k - x^{k-1}\| / \|x^{k-1}\| \leq \delta_x$;
- $\|x^k - x^{k/2}\| \leq \varepsilon_x$ или $\|x^k - x^{k/2}\| / \|x^{k/2}\| \leq \delta_x$, где k – четное число.

Здесь первый критерий используется для функций скалярного аргумента (функций с одной переменной), второй – для функций векторного аргумента (функций со многими переменными). Третий критерий используется в методах штрафных и барьерных функций.

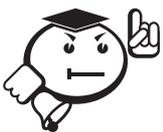
2. Критерии останова по функции (здесь ε_y – заданная абсолютная точность, а δ_y – заданная относительная точность по функции):

- $|f(x^k) - f(x^{k-1})| \leq \varepsilon_y$ или $|f(x^k) - f(x^{k-1})| / |f(x^{k-1})| \leq \delta_y$;
- $|f'(x^k)| \leq \varepsilon_y$;

- $\max_{i=1,2,\dots,n} \left(\left\| \frac{\partial f}{\partial x_i} \right\|_{x=x^k} \right) \leq \varepsilon_y;$
- $\left\| \text{grad } f(x^k) \right\|^2 = \sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} \Big|_{x=x^k} \right)^2 \leq \varepsilon_y;$
- $\left\| \text{grad } f(x^k) \right\| = \sqrt{\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} \Big|_{x=x^k} \right)^2} \leq \varepsilon_y.$

Здесь первый критерий чаще всего используется, когда производная неизвестна, или не используется (в методах прямого поиска). Вторым критерием используется в градиентных методах оптимизации функции одной переменной (скалярного аргумента). Три оставшихся критерия используют в градиентных методах, а также методах второго порядка оптимизации функции нескольких переменных (векторного аргумента). Все критерии останова по функции, кроме первого, используются только в безусловной оптимизации, т. к. в экстремальной точке задачи с ограничениями, в общем случае, условие равенства нулю производной или градиента не выполняется. В различных алгоритмах условной оптимизации предусмотрены специальные критерии останова по функции, о них будет сказано в описании данных алгоритмов.

При выполнении данных условий полагают $x^* \approx x^k$ и $f^* \approx f(x^k)$.



.....
 Если относительная точность δ_x или δ_y задана в процентах, то полученное по данным формулам значение погрешности нужно еще умножить на 100%.

1.6 Численная аппроксимация градиентов

В методах оптимизации с использованием производных аналитическое вычисление производных может быть затруднительно или невозможно. В этом случае производные (компоненты вектора градиента) можно определять на основе численной аппроксимации [4]:

- «конечная разность вперед»:

$$\left. \frac{\partial f(x)}{\partial x_i} \right|_{x=\bar{x}} \approx \frac{f(\bar{x} + \delta \cdot e_i) - f(\bar{x})}{\delta}, \quad \delta \rightarrow 0. \quad (1.18)$$

- «центральная конечная разность»:

$$\left. \frac{\partial f(x)}{\partial x_i} \right|_{x=\bar{x}} \approx \frac{f(\bar{x} + \delta \cdot e_i) - f(\bar{x} - \delta \cdot e_i)}{2\delta}, \quad \delta \rightarrow 0. \quad (1.19)$$

Выбор δ осуществляется в зависимости от вида функции $f(x)$, координат точки \bar{x} и точности ЭВМ. Аппроксимация (1.19) более точна, однако здесь используется дополнительное значение функции. Здесь e_i – единичный вектор для i -й координаты вектора \bar{x} .

1.7 Классы алгоритмов оптимизации

Приведем список классов алгоритмов оптимизации и научной литературы, рекомендуемых для более глубокого изучения материала по дисциплине «Методы оптимизации».

1. Методы одномерной оптимизации:
 - без использования производных [1–6, 9–11, 14, 17];
 - с использованием производных [1–6, 8–12, 17];
 - мультимодальных функций [6, 13, 15].
2. Методы многомерной оптимизации без ограничений:
 - прямого поиска без использования производных [1–6, 8–12, 17];
 - градиентные методы [1–4, 6, 8–12, 15, 17].
3. Линейная оптимизация:
 - методы линейного программирования [1–3, 5–6, 17];
 - методы целочисленного программирования [1–3, 6].
4. Методы нелинейного программирования с ограничениями:
 - методы штрафных и барьерных функций [1, 2, 6, 15, 17];
 - методы линеаризации [1–2, 4, 6, 8];
 - сепарабельное программирование [1];
 - методы условного градиента и проекции градиента [1, 2, 17];
 - квадратичное программирование [1, 2, 10, 11];
 - динамическое программирование [1–2, 6, 15, 17];
 - вариационное исчисление [2, 6, 17];
 - задачи оптимального управления [1–4, 6, 7, 15].



Контрольные вопросы по главе 1

1. Что является решением ЗО? Дайте определение локального и глобального экстремума ЗО.
2. Запишите ЗО общего вида. Приведите классификацию ЗО по виду ЦФ и ограничений.
3. Какие функции называются унимодальными? Приведите критерии для проверки унимодальности функций.
4. Какие функции и множества называются выпуклыми? Приведите примеры.
5. Что такое квадратичная форма? Приведите критерии определенности квадратичных форм (теорема Сильвестра).
6. Приведите необходимые и достаточные условия существования экстремума для скалярного случая. Что такое «точка перегиба» функции и как ее идентифицировать?
7. Пусть точка удовлетворяет ДУ существования локального минимума. Как установить, является ли этот минимум глобальным?
8. Перечислите основные характеристики алгоритмов оптимизации. Дайте определения сходимости и точности решения ЗО.
9. Приведите примеры методов поиска нулей функции.
10. Назовите основные критерии останова работы оптимизационных алгоритмов.

2 Методы минимизации функции одной переменной

Начнем изучение методов решения ЗО с задач одномерной оптимизации без ограничений. Такие задачи относятся к классу задач *безусловной оптимизации* для функции одной переменной (функции скалярного аргумента), или задач одномерного поиска. В данной главе рассмотрим методы решения таких задач.

2.1 Классификация методов

Задача оптимизации функций одной переменной относится к наиболее простому классу оптимизационных задач. Тем не менее, анализ задач такого типа занимает центральное место в оптимизационных исследованиях. Такие задачи обычно решаются в инженерной практике. Вместе с тем, одномерные методы оптимизации часто используются для анализа подзадач, которые возникают при решении многомерных задач оптимизации. Существует большое число алгоритмов решения задач с одной переменной. Классификация методов таких задач основывается на различных предположениях и допущениях относительно природы и свойств функции $f(x)$.

Методы, ориентированные на решение одномерных задач безусловной оптимизации, можно условно разделить на три больших класса [14]:

- 1) *методы прямого поиска* (нулевого порядка), основанные на вычислении только значений ЦФ. Это методы исключения интервалов, а также методы, использующие квадратичную аппроксимацию;
- 2) *методы 1-го порядка*, в которых используются точные значения первых производных. Это метод средней точки и другие интервальные методы решения уравнения $f'(x) = 0$, метод кубической аппроксимации и т. д.;
- 3) *методы 2-го порядка*, в которых используются также вторые производные целевой функции $f(x)$. Это метод Ньютона – Рафсона и другие итерационные методы решения уравнения $f'(x) = 0$.

2.2 Методы исключения интервалов

К данному классу относятся такие методы, как метод равномерного поиска, метод дихотомии, метод золотого сечения, метод Фибоначчи [1–2, 5–6, 9, 11].

Рассмотрим следующую задачу условной оптимизации: найти минимум унимодальной функции скалярного аргумента $f(x)$, определенной в замкнутой области допустимых значений $D = [a, b]$:

$$x^* : f(x) \rightarrow \min_{x \in [a, b]}, \quad (2.1)$$

или

$$f(x^*) = \min_{x \in [a, b]} f(x). \quad (2.2)$$

Решение следует найти с требуемой точностью по аргументу (ε_x или δ_x) и требуемой точностью по функции (ε_y или δ_y). Это означает, что должны выполняться условия:

$$|x^* - \bar{x}| \leq \varepsilon_x \quad \text{или} \quad \left| \frac{x^* - \bar{x}}{\bar{x}} \right| \leq \delta_x \quad (2.3)$$

и

$$|f(x^*) - f(\bar{x})| \leq \varepsilon_y \quad \text{или} \quad \left| \frac{f(x^*) - f(\bar{x})}{f(\bar{x})} \right| \leq \delta_y, \quad (2.4)$$

где x^* – точные координаты точки экстремума, а \bar{x} – приближение точки экстремума, найденное численным алгоритмом. Предполагается, что производные ЦФ не используются (возможно, неизвестны или не существуют).

Идея алгоритмов, относящихся к методу сокращения текущего интервала неопределенности, состоит в исключении в процессе поиска из рассмотрения тех подынтервалов, в которых в силу унимодальности функции $f(x)$ точка x^* отсутствует.

Текущий интервал неопределенности будем обозначать как $[a, b]$ (т. е. все интервалы в данном разделе являются закрытыми интервалами, или отрезками), а его длину как Δ . Таким образом, $\Delta = b - a$.

2.2.1 Метод равномерного поиска

В методе равномерного поиска (МРП) испытания проводятся в точках, которые определяются путем равномерного деления интервала $[a, b]$ на N одинаковых подынтервалов. Из вычисленных значений функции $f(x)$ выбирается наименьшее. Пусть это значение достигается в точке x_j . Тогда, в связи с унимодальностью функции $f(x)$, подынтервалы $[a, x_{j-1}]$ и $[x_{j+1}, b]$ можно ис-

ключить из рассмотрения, т. е. сделать очередным интервалом неопределенности интервал $[x_{j-1}, x_{j+1}]$. Очевидно, что если $N \leq 2$, то уменьшение интервала происходить не будет, поэтому значение N нужно выбирать из условия $N \geq 3$. Алгоритм относится к классу пассивных методов поиска.

Более строго описанную схему алгоритма можно записать в следующем виде:

1. Выполняем присваивания $k = 0$, $a^0 = a$, $b^0 = b$, $\Delta^0 = b^0 - a^0$,

$$\bar{x}^0 = \frac{a^0 + b^0}{2}.$$

2. На интервале $[a^k, b^k]$ строим равномерную сетку с $N + 1$ узлом (см. рис. 2.1):

$$x_i^k = a^k + \frac{\Delta^k}{N} \cdot i, \quad i = 0, 1, \dots, N,$$

$$\Delta^k = b^k - a^k.$$

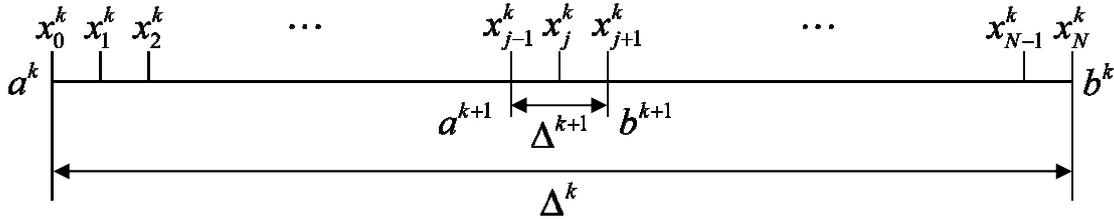


Рис. 2.1 – Построение сетки на текущем интервале неопределенности

3. Вычисляем значения функции $f(x)$ в узлах построенной сетки, кроме первого и последнего: $f(x_1^k)$, $f(x_2^k)$, ..., $f(x_{N-1}^k)$.

4. Находим минимальное из этих значений:

$$j: f(x_j^k) = \min(f(x_1^k), f(x_2^k), \dots, f(x_{N-1}^k)).$$

5. Выполняем присваивания $a^{k+1} = x_{j-1}^k$, $b^{k+1} = x_{j+1}^k$,

$$\bar{x}^{k+1} = \frac{a^{k+1} + b^{k+1}}{2} = x_j^k, \quad \Delta^{k+1} = \left(\frac{2}{N}\right)^{k+1} \cdot \Delta^0 = b^{k+1} - a^{k+1}.$$

6. Если $\Delta^{k+1}/2 \leq \varepsilon_x$ и $|f(\bar{x}^{k+1}) - f(\bar{x}^k)| \leq \varepsilon_y$, то полагаем $x^* \approx \bar{x}^{k+1}$ и заканчиваем вычисления. Иначе выполняем присваивание $k = k + 1$ и переходим на шаг 2.

Легко видеть, что после одной итерации алгоритма равномерного поиска Δ уменьшается в $N/2$ раз. Поэтому

$$\Delta^k = \left(\frac{2}{N}\right)^k (b-a), \quad k = 0, 1, \dots \quad (2.5)$$

За приближение точки экстремума мы принимаем середину интервала $[a^k, b^k]$, следовательно, количество итераций k , необходимых для нахождения минимума функции с точностью по аргументу ε_x , может быть найдено из условия:

$$\left(\frac{2}{N}\right)^k \frac{b-a}{2} \leq \varepsilon_x. \quad (2.6)$$

Сколько потребуется итераций для достижения заданной точности по функции ε_y , заранее неизвестно, т. к. зависит от вида ЦФ $f(x)$. Это утверждение верно для всех методов исключения интервалов.

2.2.2 Метод деления отрезка пополам (метод дихотомии)

В методе деления отрезка пополам, или методе равномерного дихотомического поиска (сокращенно – метод дихотомии, МД), испытания проводятся парами. Координаты каждой последующей пары испытаний разнесены между собой на величину $\delta \in (0, \varepsilon_x)$. Испытания производятся в середине интервала. По значениям $f(x)$, полученным в этих точках, одна половина интервала в силу унимодальности функции $f(x)$ исключается из дальнейшего рассмотрения. Алгоритм относится к классу методов последовательного поиска.

Более строго описанную схему алгоритма можно записать в следующем виде:

1. Выполняем присваивания: $k = 0, a^0 = a, b^0 = b,$

$$\bar{x}^0 = \frac{a^0 + b^0}{2}.$$

2. Вычисляем величины:

$$x_1^k = \bar{x}^k - \frac{\delta}{2}, \quad x_2^k = \bar{x}^k + \frac{\delta}{2}. \quad (2.7)$$

3. Вычисляем значения ЦФ $f(x_1^k)$ и $f(x_2^k)$.

4. Если $f(x_1^k) < f(x_2^k)$ (рис. 2.2), то выполняем присваивания $a^{k+1} = a^k$, $b^{k+1} = x_2^k$.

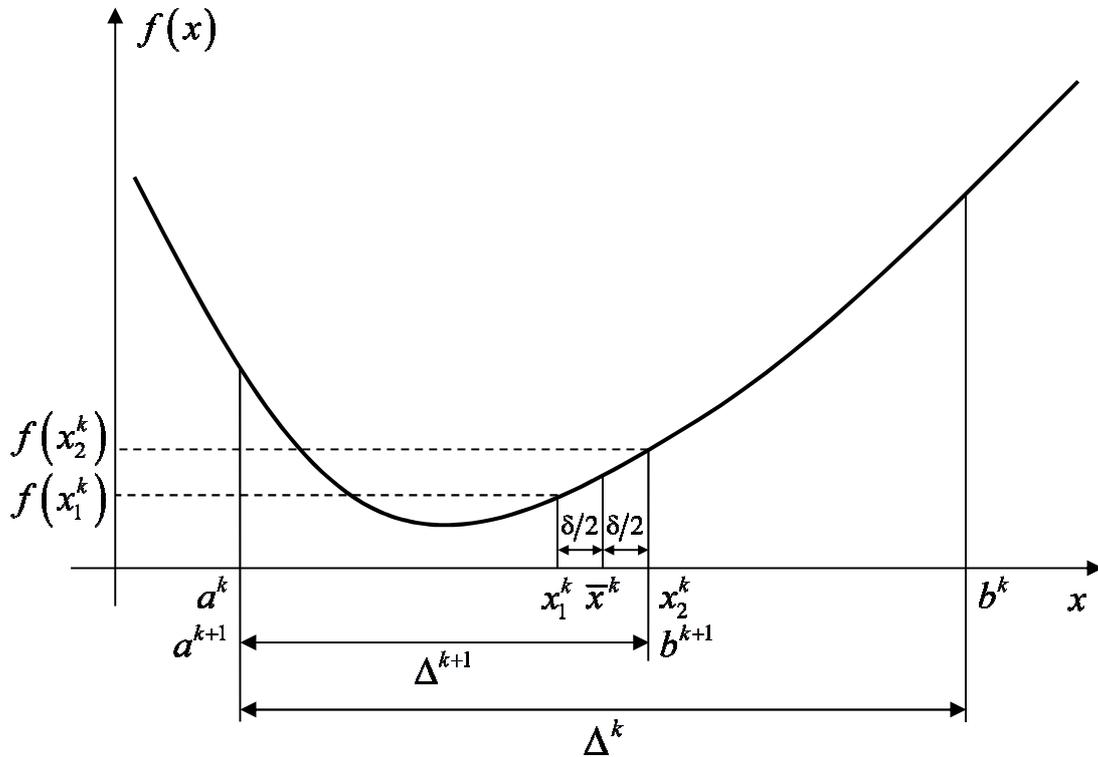


Рис. 2.2 – Итерация метода дихотомии при $f(x_1^k) < f(x_2^k)$

Иначе (рис. 2.3) выполняем присваивания: $a^{k+1} = x_1^k$, $b^{k+1} = b^k$.

Находим следующее приближение точки экстремума:

$$\bar{x}^{k+1} = \frac{a^{k+1} + b^{k+1}}{2}.$$

5. Если $\Delta^{k+1}/2 \leq \varepsilon_x$ и $|f(\bar{x}^{k+1}) - f(\bar{x}^k)| \leq \varepsilon_y$, то полагаем $x^* \approx \bar{x}^{k+1}$ и заканчиваем вычисления. Иначе выполняем присваивание $k = k + 1$ и переходим на шаг 2.

Легко видеть, что после одной итерации алгоритма деления пополам интервал уменьшается примерно в 2 раза. Поэтому

$$\Delta^k \approx \left(\frac{1}{2}\right)^k (b - a), \quad k = 0, 1, \dots \quad (2.8)$$

За приближение точки экстремума мы принимаем середину интервала $[a^k, b^k]$, следовательно, количество итераций k , необходимых для нахождения минимума функции с точностью по аргументу ε_x , может быть найдено из условия:

$$\left(\frac{1}{2}\right)^k \frac{b-a}{2} = \frac{b-a}{2^{k+1}} \leq \varepsilon_x. \quad (2.9)$$

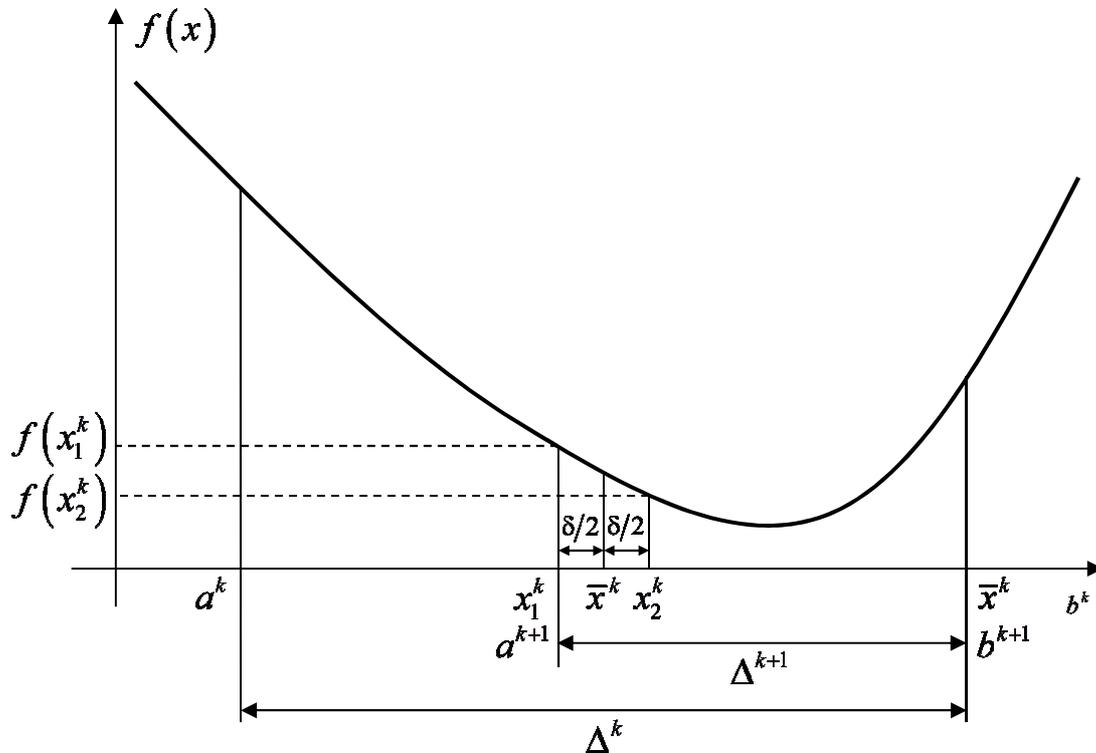


Рис. 2.3 – Итерация метода дихотомии при $f(x_1^k) > f(x_2^k)$

2.2.3 Метод Фибоначчи

Числа Фибоначчи (псевдоним Леонарда Пизанского, первым изучившего эту последовательность в 1202 г.) задаются следующим рекуррентным уравнением:

$$F_0 = 1, F_1 = 1, F_N = F_{N-1} + F_{N-2}, N = 2, 3, \dots \quad (2.10)$$

Некоторые числа Фибоначчи приведены в таблице 2.1.

Таблица 2.1 – Несколько первых чисел Фибоначчи

N	0	1	2	3	4	5	6	7	8	9	10	...
F_N	1	1	2	3	5	8	13	21	34	55	89	...

Общее выражение для N -го числа Фибоначчи можно получить из решения уравнения (2.10):

$$F_N = \frac{\gamma^{N+1} - (-\gamma)^{-(N+1)}}{\sqrt{5}} \quad \text{или} \quad F_N = \frac{\gamma^{N+1} - (1-\gamma)^{N+1}}{\sqrt{5}}, \quad (2.11)$$

где $\gamma = \frac{1+\sqrt{5}}{2} \approx 1,618$.

При больших значениях N членом $(1-\gamma)^{N+1}$ можно пренебречь. При этом

$$F_N \approx \frac{\gamma^{N+1}}{\sqrt{5}}. \quad (2.12)$$

Отсюда следует, что $F_{N+1}/F_N \approx \gamma$, т. е. отношение двух соседних чисел Фибоначчи примерно постоянно и равно γ . Это соотношение тем точнее, чем больше индекс N :

$$\lim_{N \rightarrow \infty} \frac{F_{N+1}}{F_N} = \gamma. \quad (2.13)$$

Метод Фибоначчи (МФ) относится к классу поисковых методов оптимизации и включает в себя два этапа.

I. Первый этап состоит из $N-1$ итерации для $k=0,1,\dots,N-2$. Рассмотрим схему k -й итерации на интервале $[a^k, b^k]$, когда $\Delta^k = b^k - a^k$:

1. Вычисляем величины:

$$x_1^k = b^k - \Delta^k \cdot \frac{F_{N-1-k}}{F_{N-k}}, \quad x_2^k = a^k + \Delta^k \cdot \frac{F_{N-1-k}}{F_{N-k}}. \quad (2.14)$$

2. Вычисляем значения ЦФ $f(x_1^k)$ и $f(x_2^k)$.

3. Если $f(x_1^k) < f(x_2^k)$ (рис. 2.4), то выполняем присваивания $a^{k+1} = a^k$, $b^{k+1} = x_2^k$.

Иначе (рис. 2.5) выполняем присваивания $a^{k+1} = x_1^k$, $b^{k+1} = b^k$.

Алгоритм Фибоначчи обладает следующим свойством: после выполнения итерации $k = N-2$ имеет место ситуация: $x_1^{N-2} = x_2^{N-2} = \bar{x}^{N-2}$. В результате $N-1$ итерации сужения текущего интервала неопределенности не происходит:

$$\Delta^{N-2} = \Delta^{N-1}, \quad [a^{N-2}, b^{N-2}] = [a^{N-1}, a^{N-1}].$$

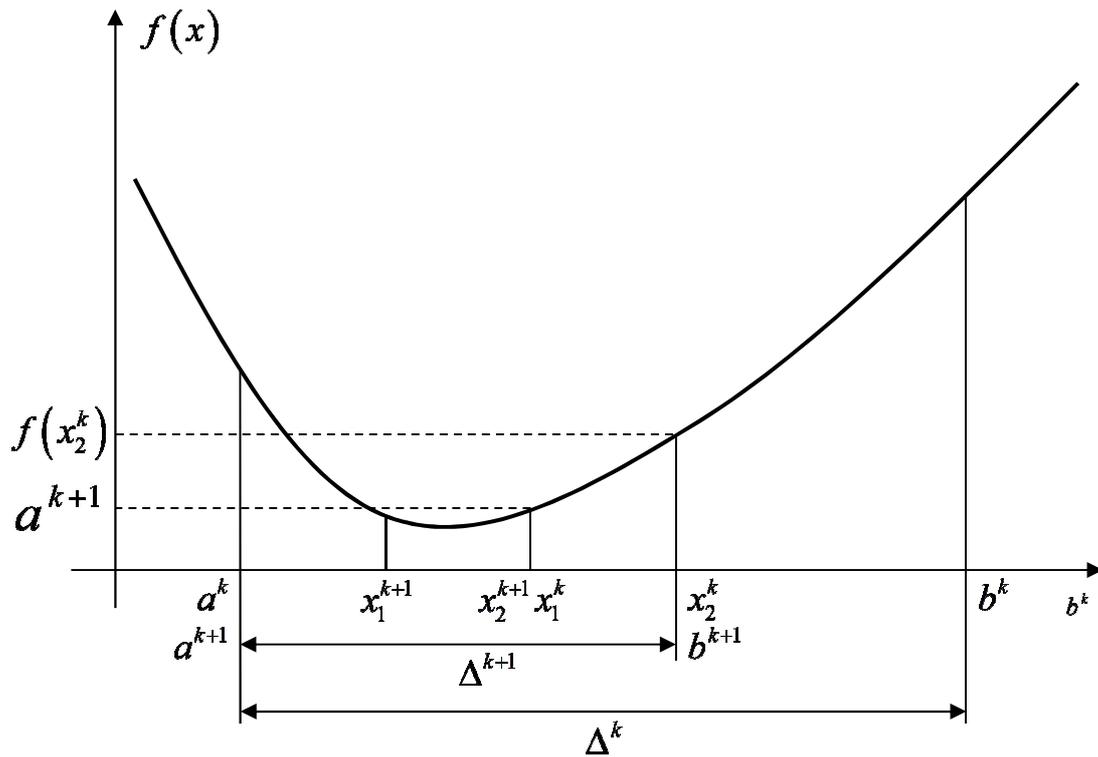


Рис. 2.4 – Итерация метода Фибоначчи и золотого сечения при $f(x_1^k) < f(x_2^k)$

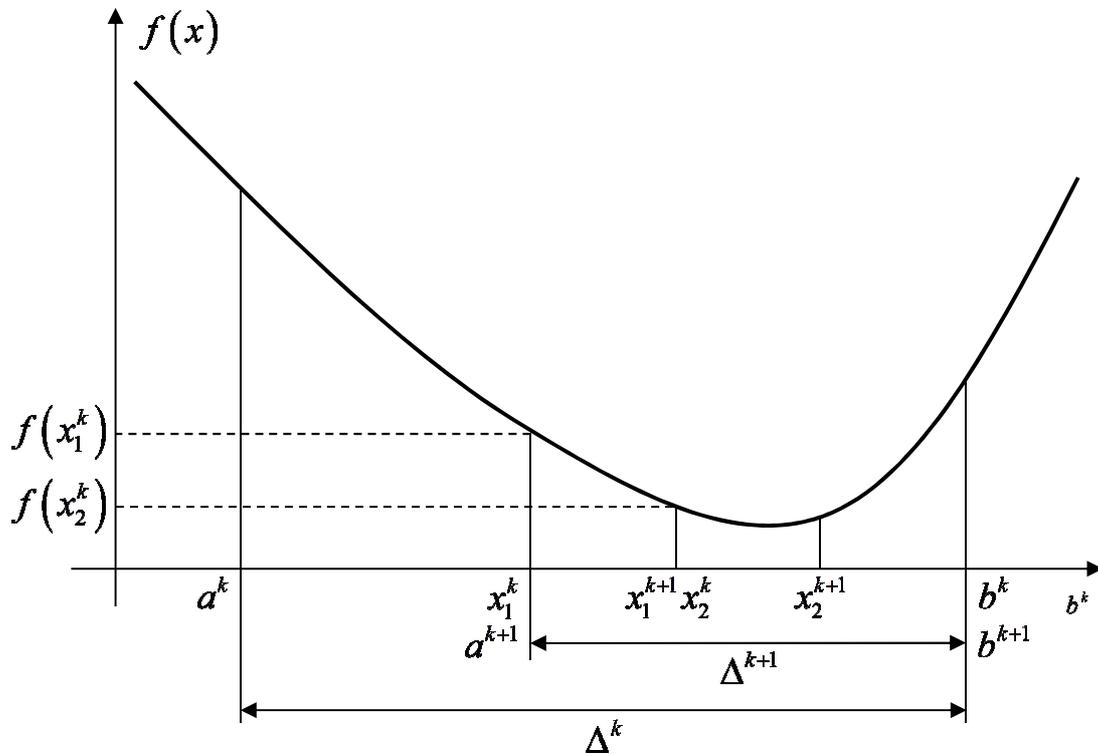


Рис. 2.5 – Итерация метода Фибоначчи и золотого сечения при $f(x_1^k) > f(x_2^k)$

II. Второй этап призван решить, по какую сторону от точки x^{N-2} лежит точка минимума функции $f(x)$. Второй этап выполняется по следующей схеме:

1. Находим точки $x_1^{N-1} = \bar{x}^{N-2}$, $x_2^{N-1} = \bar{x}^{N-2} + \delta$, где $\delta \in [\varepsilon_x/2, \varepsilon_x]$ – свободный параметр алгоритма.
2. Вычисляем значение ЦФ $f(x_2^{N-1})$ (значение ЦФ $f(x_1^{N-1}) = f(x^{N-2})$ нам уже известно).
3. Если $f(x_1^{N-1}) < f(x_2^{N-1})$, то выполняем присваивания $a^N = a^{N-1}$, $b^N = x_1^{N-1}$. Иначе выполняем присваивания $a^N = x_1^{N-1}$, $b^N = b^{N-1}$. Точка x_2^{N-1} – пробная, она на этом шаге решения не используется.
4. В качестве приближенного значения точки минимума принимаем

$$x^* \approx \frac{a^N + b^N}{2}.$$

Рассмотрим без доказательства некоторые свойства алгоритма Фибоначчи.



.....
Утверждение 1. Для любого $k \in [0, N-3]$ алгоритм Фибоначчи обладает следующим свойством: одна из точек x_1^{k+1} или x_2^{k+1} совпадает с одной из точек x_1^k или x_2^k (см. рис. 2.4, 2.5).

Указанное свойство алгоритма Фибоначчи позволяет на каждой итерации (кроме первой) производить испытания только в одной точке, т. е. если $f(x_1^k) < f(x_2^k)$ (рис. 2.4), то $x_2^{k+1} = x_1^k$. Иначе (рис. 2.5) $x_1^{k+1} = x_2^k$.



.....
Утверждение 2. Точки x_1^k , x_2^k расположены симметрично относительно концов текущего интервала неопределенности $[a^k, b^k]$, т. е. расстояние точки x_1^k до точки a^k равно расстоянию точки x_2^k до точки b^k или, что то же самое, $x_2^k - a^k = b^k - x_1^k$ либо $x_1^k - a^k = b^k - x_2^k$.

Соответственно,

$$x_1^k = a^k + \Delta^k \cdot \frac{F_{N-2-k}}{F_{N-k}}, \quad x_2^k = b^k - \Delta^k \cdot \frac{F_{N-2-k}}{F_{N-k}}. \quad (2.15)$$

Для упрощения вычислений можно вычислить расстояние от границ интервала $[a^k, b^k]$ до точек x_1^k и x_2^k :

$$L^k = \Delta^k \cdot \frac{F_{N-2-k}}{F_{N-k}}, \quad (2.16)$$

и затем

$$x_1^k = a^k + L^k, \quad x_2^k = b^k - L^k. \quad (2.17)$$



.....
Утверждение 3. В результате любой итерации $k \in [0, N-3]$ алгоритма Фибоначчи длина текущего интервала неопределенности уменьшается в F_{N-1-k}/F_{N-k} раз.

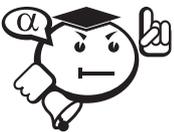
Соответственно,

$$\Delta^{k+1} = \frac{F_{N-1-k}}{F_{N-k}} \cdot \Delta^k. \quad (2.18)$$



.....
Утверждение 4. При достаточно больших N в результате одной итерации алгоритма Фибоначчи длина текущего интервала неопределенности уменьшается примерно в γ раз, где $\gamma \approx 1,618$ – золотое сечение.

Из утверждения 4 следует, что при достаточно больших N алгоритм Фибоначчи практически идентичен алгоритму золотого сечения.



.....
Утверждение 5. В результате N итерации алгоритма Фибоначчи длина интервала неопределенности уменьшается в F_N раз.

Соответственно,

$$\Delta^N = \frac{1}{F_N}(b-a). \quad (2.19)$$

За приближение точки экстремума мы принимаем середину интервала Δ^N , следовательно, количество итераций N , необходимых для нахождения минимума функции с точностью по аргументу ε_x , может быть найдено из условия

$$\frac{1}{F_N} \cdot \frac{b-a}{2} = \frac{b-a}{2F_N} \leq \varepsilon_x. \quad (2.20)$$

2.2.4 Метод золотого сечения

Метод золотого сечения (МЗС) похож на метод Фибоначчи, меняется только способ вычисления точек x_1^k и x_2^k , чтобы отношение длин интервалов было постоянным, т. е.

$$\frac{\Delta^0}{\Delta^1} = \frac{\Delta^1}{\Delta^2} = \dots = \frac{\Delta^{k+1}}{\Delta^k} = \gamma. \quad (2.21)$$

При этом также должно выполняться соотношение

$$\Delta^k = \Delta^{k-1} + \Delta^{k+2}. \quad (2.22)$$

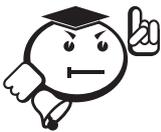
Соответственно, рисунки 2.4 и 2.5 также иллюстрируют и метод золотого сечения.

Из соотношений (2.21) и (2.22) следует, что

$$\gamma = \frac{1 + \sqrt{5}}{2} \approx 1,618, \quad (2.23)$$

а также что

$$\frac{1}{\gamma} = \gamma - 1. \quad (2.24)$$



.....

Число γ называют «золотым сечением» (золотой пропорцией, гармоническим делением и т. д.). В дошедшей до нас литературе впервые встречается в «Началах» Евклида (хотя сам термин был введен значительно позже, возможно, Леонардо да Винчи).

.....

Алгоритм золотого сечения относится к классу последовательных методов поиска. В целом, он аналогичен первой части алгоритма Фибоначчи, кроме способа вычисления точек x_1^k и x_2^k . Но, в отличие от метода Фибоначчи, длина интервала Δ^k может уменьшаться до сколь угодно малой величины, поэтому можно использовать обычный критерий окончания итерационного процесса, как, например, в методе дихотомии:

1. Выполняем присваивания: $k = 0$, $a^0 = a$, $b^0 = b$,

$$\bar{x}^0 = \frac{a^0 + b^0}{2}.$$

2. Вычисляем величины (см. рис. 2.4, 2.5):

$$x_1^k = a^k + \frac{\Delta^k}{\gamma^2}, \quad x_2^k = a^k + \frac{\Delta^k}{\gamma}. \quad (2.25)$$

3. Вычисляем значения ЦФ $f(x_1^k)$ и $f(x_2^k)$.

4. Если $f(x_1^k) < f(x_2^k)$ (рис. 2.4), то выполняем присваивания $a^{k+1} = a^k$, $b^{k+1} = x_2^k$. Иначе (рис. 2.5) выполняем присваивания $a^{k+1} = x_1^k$, $b^{k+1} = b^k$. Находим следующее приближение точки экстремума:

$$\bar{x}^{k+1} = \frac{a^{k+1} + b^{k+1}}{2}.$$

5. Если $\Delta^{k+1}/2 \leq \varepsilon_x$ и $|f(\bar{x}^{k+1}) - f(\bar{x}^k)| \leq \varepsilon_y$, то полагаем $x^* \approx \bar{x}^{k+1}$ и заканчиваем вычисления. Иначе выполняем присваивание $k = k + 1$ и переходим на шаг 2.

Приведем без доказательства некоторые свойства алгоритма золотого сечения. Они аналогичны соответствующим свойствам алгоритма Фибоначчи, с поправкой на изменившиеся пропорции отрезков.



.....
Утверждение 1. Для любого k алгоритм золотого сечения обладает следующим свойством: одна из точек x_1^{k+1} или x_2^{k+1} совпадает с одной из точек x_1^k или x_2^k (см. рис. 2.4, 2.5).

Указанное свойство позволяет на каждой итерации (кроме первой) производить испытания только в одной точке, т. е. если $f(x_1^k) < f(x_2^k)$ (рис. 2.4), то $x_2^{k+1} = x_1^k$. Иначе (рис. 2.5) $x_1^{k+1} = x_2^k$.



.....
Утверждение 2. Точки x_1^k , x_2^k расположены симметрично относительно концов текущего интервала неопределенности $[a^k, b^k]$, т. е. расстояние точки x_1^k до точки a^k равно расстоянию точки x_2^k до точки b^k или, что то же самое, $x_2^k - a^k = b^k - x_1^k$ либо $x_1^k - a^k = b^k - x_2^k$.

Соответственно,

$$x_1^k = a^k + \frac{\Delta^k}{\gamma^2} = b^k - \frac{\Delta^k}{\gamma}, \quad x_2^k = a^k + \frac{\Delta^k}{\gamma} = b^k - \frac{\Delta^k}{\gamma^2}. \quad (2.26)$$

Для упрощения вычислений можно вычислить расстояние от границ интервала $[a^k, b^k]$ до точек x_1^k и x_2^k :

$$L_1^k = \frac{\Delta^k}{\gamma}, \quad (2.27)$$

и затем

$$x_1^k = b^k - L_1^k, \quad x_2^k = a^k + L_1^k, \quad (2.28)$$

или

$$L_2^k = \frac{\Delta^k}{\gamma^2}, \quad (2.29)$$

и затем

$$x_1^k = a^k + L_2^k, \quad x_2^k = b^k - L_2^k. \quad (2.30)$$

Также можно вычислить по формулам (2.25), (2.28) или (2.30) только одну точку, а вторую – из соотношения

$$x_1^k = a^k + b^k - x_2^k \quad \text{или} \quad x_2^k = a^k + b^k - x_1^k. \quad (2.31)$$



.....
Утверждение 3. В результате любой итерации алгоритма золотого сечения длина текущего интервала неопределенности уменьшается в γ раз.

Соответственно, в результате k итерации алгоритма золотого сечения длина текущего интервала неопределенности становится равной

$$\Delta^k = \frac{1}{\gamma^k}(b - a). \quad (2.32)$$

За приближение точки экстремума мы принимаем середину интервала $[a^k, b^k]$, следовательно, количество итераций k , необходимых для нахождения минимума функции с точностью по аргументу ε_x , может быть найдено из условия:

$$\frac{1}{\gamma^k} \cdot \frac{b - a}{2} = \frac{b - a}{2\gamma^k} \leq \varepsilon_x. \quad (2.33)$$

2.3 Полиномиальная аппроксимация и методы точечного оценивания

Применение методов исключения интервалов требует унимодальности целевой функции и основано на простом сравнении значений функции в двух пробных точках. Рассмотрим более эффективные методы, чем методы исключения интервалов [1, 9]. Но выигрыш в эффективности достигается ценой введения дополнительного требования – исследуемые функции должны быть гладкими.

Основная идея рассматриваемых методов связана с возможностью аппроксимации гладкой функции полиномом и последующего использования этого полинома для оценивания координаты точки экстремума.

Необходимые условия эффективной реализации такого подхода – унимодальность и непрерывность исследуемой функции. Согласно теореме Вейерштрасса об аппроксимации, если функция непрерывна на некотором интервале, то ее с любой степенью точности можно аппроксимировать полиномом высокого порядка. Координату точки экстремума функции можно оценить путем вычисления координаты точки экстремума полинома.

Точность оценки точки экстремума можно повысить двумя способами:

- 1) использование полинома более высокого порядка;
- 2) уменьшение интервала аппроксимации.

Второй способ более прост и экономичен, если функция унимодальная.

2.3.1 Квадратичная аппроксимация

Простейший вариант полиномиальной аппроксимации основан на факте, что функция $f(x)$ имеет экстремум во внутренней точке интервала и должна быть, по крайней мере, квадратичной. Линейная функция достигает экстремума в двух граничных точках интервала.

Если задана последовательность точек x_1, x_2, x_3 и известны соответствующие этим последовательностям значения функции f_1, f_2, f_3 , то можно определить константы a_0, a_1, a_2 таким образом, что значения квадратичной функции

$$q(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2) \quad (2.34)$$

совпадут со значениями функции в трех указанных точках. Перейдем к вычислению параметров a_0, a_1, a_2 .

Так как $f_1 = f(x_1) = q(x_1) = a_0$, имеем

$$a_0 = f_1. \quad (2.35)$$

Далее, поскольку $f_2 = f(x_2) = q(x_2) = f_1 + a_1(x_2 - x_1)$, получаем

$$a_1 = \frac{f_2 - f_1}{x_2 - x_1}. \quad (2.36)$$

Наконец, при $x = x_3$ имеем:

$$f_3 = f(x_3) = q(x_3) = f_1 + \frac{f_2 - f_1}{x_2 - x_1}(x_3 - x_1) + a_2(x_3 - x_1)(x_3 - x_2).$$

Разрешая последнее уравнение относительно a_2 , получим:

$$a_2 = \frac{1}{x_3 - x_2} \left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right). \quad (2.37)$$

Таким образом, по трем заданным точкам и соответствующим значениям функции можно оценить параметры a_0 , a_1 , a_2 аппроксимирующего квадратичного полинома.

Если точность аппроксимации исследуемой функции высока на интервале от x_1 до x_3 , то построенный полином можно использовать для оценивания координаты точки экстремума. В данном случае из уравнения

$$\frac{dq}{dx} = a_1 + a_2(x - x_2) + a_2(x - x_1) = 0$$

можно получить стационарную точку:

$$\bar{x} = \frac{x_2 + x_1}{2} - \frac{a_1}{2a_2}. \quad (2.38)$$

Так как ЦФ $f(x)$ и аппроксимирующий полином являются унимодальными на рассматриваемом интервале, то \bar{x} является приемлемой оценкой координаты точки истинного экстремума x^* . Для более точного определения точки экстремума данную процедуру нужно повторять последовательно, как это делается, например, в методе Пауэлла.

2.3.2 Метод Пауэлла

Метод Пауэлла основан на последовательном применении процедуры оценивания с использованием квадратичной аппроксимации.

Схема алгоритма Пауэлла: пусть x^0 – начальная точка; Δx – выбранная величина шага по оси x ; ε_x , ε_y – требуемая точность решения.

Шаг 1. Положить $k = 0$, $x_1^0 = x^0$. Вычислить $x_2^0 = x_1^0 + \Delta x$.

Шаг 2. Вычислить $f_1^0 = f(x_1^0)$ и $f_2^0 = f(x_2^0)$.

Шаг 3. Если $f_1^0 > f_2^0$, положить $x_3^0 = x_2^0 + \Delta x = x_1^0 + 2\Delta x$, иначе $x_3^0 = x_1^0 - \Delta x$.

Вычислить $f_3^0 = f(x_3^0)$. Если $x_3^0 < x_1^0$, то перенумеровать точки в естественном порядке: $x_1^0 = x_3^0$, $x_2^0 = x_1^0$, $x_3^0 = x_2^0$, $f_1^0 = f_3^0$, $f_2^0 = f_1^0$, $f_3^0 = f_2^0$.

Шаг 4. Найти

$$f_{\min} = \min \{f_1^k, f_2^k, f_3^k\}.$$

Значение x_{\min} равно точке x_i^k , которая соответствует f_{\min} .

Шаг 5. По трем точкам x_1^k, x_2^k, x_3^k вычислить \bar{x} , используя формулу (2.38), т. е. используя квадратичную аппроксимацию.

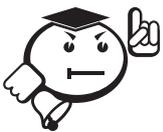
Шаг 6. Проверка на окончание поиска:

а) является ли разность $|f_{\min} - f(\bar{x})| \leq \varepsilon_y$;

б) является ли разность $|x_{\min} - \bar{x}| \leq \varepsilon_x$.

Если условия (а) и (б) выполняются одновременно, то закончить поиск, за оптимальную точку x^* принять наилучшую из точек x_{\min} и \bar{x} (т. е. ту из них, которой соответствует меньшее значение ЦФ). Иначе переход на шаг 7.

Шаг 7. Выбрать наилучшую точку (x_{\min} или \bar{x}) и две точки по обе стороны от нее. Обозначить эти точки в естественном порядке как $x_1^{k+1}, x_2^{k+1}, x_3^{k+1}$, положить $k = k + 1$ и перейти на шаг 4.



.....
После пятого шага необходимо провести дополнительную проверку, т. к. точка \bar{x} может находиться вне интервала $[x_1^k, x_3^k]$.

В этом случае точка \bar{x} заменяется точкой, координата которой вычисляется с учетом заранее установленной длины шага. Если $\bar{x} < x_1^k - \Delta x$, то заменяем ее точкой $\bar{x} = x_1^k - \Delta x$, а если $\bar{x} > x_3^k + \Delta x$, то точкой $\bar{x} = x_3^k + \Delta x$.

.....

Достоинства алгоритма – обеспечивает поиск экстремума даже для функций, не являющихся унимодальными, в отличие от интервальных методов. Недостатки – сильно зависит от выбора начальной точки x_1 и величины шага Δx . Главное требование – интервал $[x_1^0, x_3^0]$, полученный на первой итерации, дол-

жен включать в себя точку экстремума – $x^* \in [x_1^0, x_3^0]$. Если это условие не обеспечено, то сходимость алгоритма не гарантирована.

2.4 Методы с использованием производных

Рассмотренные выше методы поиска основаны на предположениях об унимодальности и, в ряде случаев, о непрерывности исследуемой целевой функции. Если в дополнение к условию непрерывности ввести требование дифференцируемости, то эффективность поисковых процедур можно существенно повысить [1–2].

Напомним, что необходимым условием существования локального минимума функции, не имеющей разрывов в области определения аргумента x , в некоторой точке x^* является требование стационарности точки x^* , т. е.

$$f'(x^*) = \left. \frac{df}{dx} \right|_{x=x^*} = 0.$$

Если $f(x)$ содержит члены, содержащие x в третьей и более высоких степенях, то непосредственное получение аналитического решения уравнения $f'(x) = 0$ может оказаться затруднительным. В таких случаях используются приближенные методы последовательного поиска стационарной точки $f(x)$.

2.4.1 Метод Ньютона – Рафсона

Классическая поисковая схема разработана Ньютоном и позднее уточнена Рафсоном (МНР) [9].

Пусть имеется текущее приближение точки экстремума x^k . Тогда следующее приближение x^{k+1} будем искать в виде

$$x^{k+1} = x^k + h_k. \quad (2.39)$$

Параметр h_k попробуем определить таким образом, чтобы точка x^{k+1} совпадала с точкой экстремума, т. е.

$$f'(x^{k+1}) = f'(x^k + h_k) = 0. \quad (2.40)$$

Разложим $f'(x^k + h_k)$ в окрестности точки x^k в ряд Тейлора:

$$f'(x^k + h_k) = f'(x^k) + f''(x^k)h_k + \frac{1}{2}f'''(x^k)h_k^2 + \dots = 0.$$

Решить данное уравнение в общем виде невозможно. Оставим в уравнении только два первых члена, а остальные отбросим:

$$f'(x^k + h_k) \approx f'(x^k) + f''(x^k)h_k = 0,$$

$$\Rightarrow h_k = -\frac{f'(x^k)}{f''(x^k)}.$$

Окончательная итерационная формула метода:

$$x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)}. \quad (2.41)$$

Отбрасывая члены ряда Тейлора, мы сознательно вносим в решение погрешность, поэтому, в общем случае, условие (2.40) выполняться не будет. Исключение – если ЦФ имеет нулевые производные, от третьего порядка включительно. Например, если функция квадратичная.



.....
 Для квадратичных функций точка экстремума методом Ньютона – Рафсона будет найдена за одну итерацию.

Также следует помнить, что ряд Тейлора является сходящимся, т. е.

$$\lim_{n \rightarrow \infty} \frac{1}{n!} f^{(n+1)}(x^k) h_k^n = 0,$$

поэтому отбрасывание его старших членов хотя и вносит погрешность, но не такую существенную.

Геометрическое пояснение. В точке x^k к графику производной $f'(x)$ строится касательная. Точка пересечения касательной с осью x будет следующим приближением точки экстремума x^{k+1} . В этой точке снова строится касательная, получим точку x^{k+2} и т. д. Если при этом происходит постепенное приближение к точке экстремума x^* , то процесс является сходящимся (рис. 2.6). Сходимость метода квадратичная.

Если же происходит удаление от точки экстремума, процесс расходится (рис. 2.7).

Чтобы процесс сошелся, важно удачно выбрать начальную точку x^0 . Вспомним критерий выбора начальной точки в классическом методе Ньютона поиска нулей функции:

$$f(x^0) \cdot f''(x^0) > 0.$$

Соответственно, если искать нули не функции, а производной, то

$$f'(x^0) \cdot f'''(x^0) > 0. \quad (2.42)$$

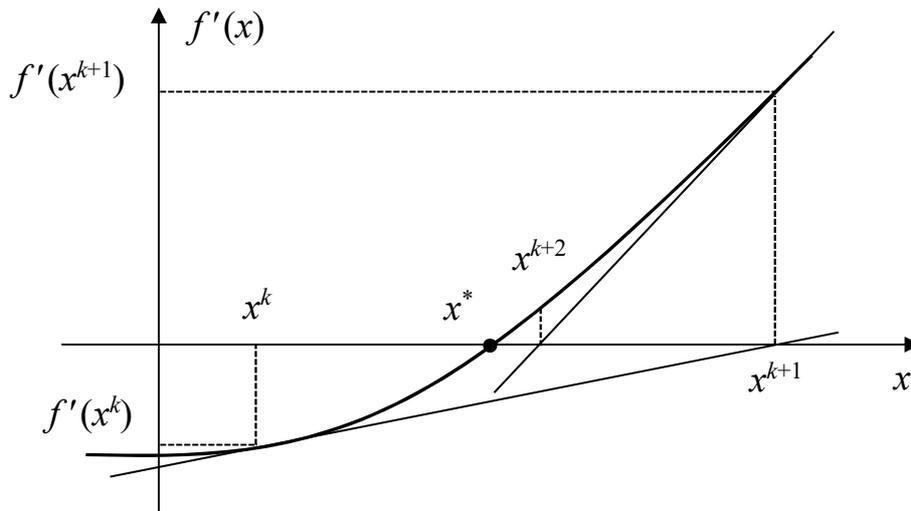


Рис. 2.6 – Сходимость МНР

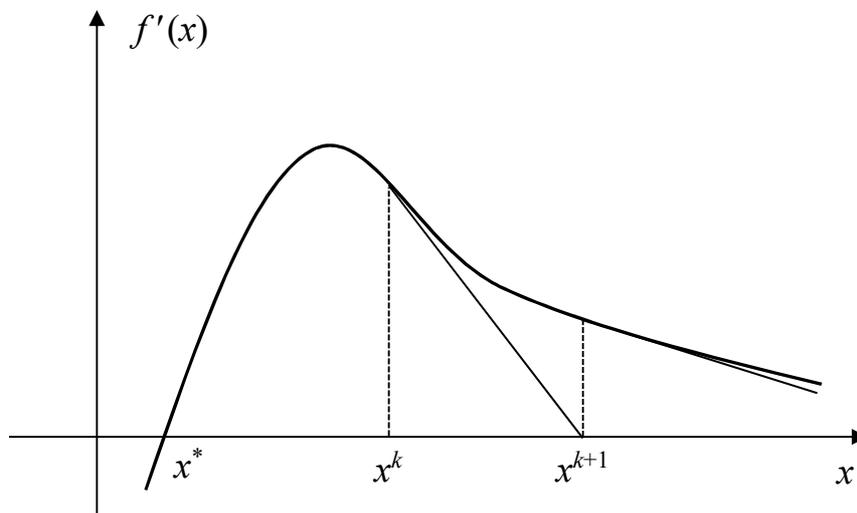


Рис. 2.7 – Отсутствие сходимости

Если точка удовлетворяет этому условию, то она подходит для начала итерационного процесса. В первую очередь проверяют границы отрезка, если обе границы не подходят – другие точки, например середину отрезка и т. п.

Метод Ньютона является итерационным методом поиска нулей функции. Соответственно, МНР – итерационный метод поиска экстремумов. Данный метод может найти как минимум ЦФ, так и максимум, или точку перегиба. По окончании итерационного процесса найденную точку нужно классифицировать, используя соответствующие критерии.

В целом, для решения уравнения $f'(x) = 0$ можно использовать любые численные методы решения уравнений с одной переменной, как интервальные – метод дихотомии, золотого сечения и т. д., так и итерационные – Ньюто-

на (касательных), хорд, итераций и т. д. [1–6]. Пример решения задачи методом Ньютона – Рафсона приведен далее.

2.4.2 Другие итерационные методы поиска нулей функции

Поиск нулей функции $f(x)$ широко используется в методах оптимизации [4, 6]. Например, такая необходимость возникает при определении стационарной точки $f'(x) = 0$ или решении системы нелинейных уравнений $\partial f / \partial x_i = 0$, $i = 1, 2, \dots, n$, как в методе Ньютона для минимизации функций векторного аргумента.

Как уже отмечалось, рассмотренный МНР – это обычный метод Ньютона, только предназначенный для решения уравнения $f'(x) = 0$, а не $f(x) = 0$.

Во всех итерационных методах процесс приближения к корню строится по одинаковой схеме (2.39), где $h_k = \lambda_k \cdot f(x^k)$, т. е.

$$x^{k+1} = x^k + \lambda_k \cdot f(x^k). \quad (2.43)$$

Критерий останова итерационного процесса выглядит следующим образом:

$$|x^{k+1} - x^k| \leq \varepsilon_x, \quad |f(x^{k+1})| \leq \varepsilon_y, \quad (2.44)$$

где ε_x и ε_y – требуемая точность решения по аргументу и по функции соответственно.

Кроме широко известных методов поиска нулей функции, перечисленных выше, можно предложить следующие методы [7]:

1. *Упрощенный метод Ньютона* (сходимость линейная):

$$x^{k+1} = x^k - \frac{1}{\alpha} \cdot f(x^k), \quad (2.45)$$

где α – некоторая константа, например, $\alpha = f'(x^0)$ или $\alpha = \Lambda$.

2. *Метод секущих* (сходимость суперлинейная):

$$x^{k+1} = x^k - \frac{x^k - x^{k-1}}{f(x^k) - f(x^{k-1})} \cdot f(x^k). \quad (2.46)$$

Для начала итерационного процесса необходимы две начальные точки – x^0 и x^1 .

3. *Метод Стефенсена* (квадратичная сходимость):

$$x^{k+1} = x^k - \frac{f(x^k)}{f(x^k + f(x^k)) - f(x^k)} \cdot f(x^k). \quad (2.47)$$

4. *Метод Уолла* (кубическая сходимость):

$$x^{k+1} = x^k - \frac{1}{f'(x^k) - \frac{f(x^k) \cdot f''(x^k)}{2f'(x^k)}} \cdot f(x^k). \quad (2.48)$$

Чтобы использовать эти методы для поиска экстремумов функции, достаточно произвести замену ЦФ в формулах (2.45)–(2.48) на ее производную $f(x) \rightarrow f'(x)$, производную на вторую производную $f'(x) \rightarrow f''(x)$ и т. д. Соответственно, критерий останова итерационного процесса (2.44) переписывается в виде:

$$|x^{k+1} - x^k| \leq \varepsilon_x, \quad |f'(x^{k+1})| \leq \varepsilon_y. \quad (2.49)$$

В отличие от методов исключения интервалов, МНР (и другие методы, использующие производные) находят на отрезке $[a, b]$ любую стационарную точку, т. е. это может быть минимум, максимум или точка перегиба. Полученную в результате решения точку обязательно нужно классифицировать, используя соответствующие критерии.

Обобщенный алгоритм методов с использованием производных:

1. Выполняем присваивание $k = 0$, а также выбираем на отрезке $[a, b]$ начальную точку x^0 , используя какой-либо критерий, например (2.42). При этом проверяют обычно границы отрезка, а если они не подходят – его середину, и т. д.

2. Вычисляем следующее приближение точки экстремума, используя формулу (2.41), либо одну из формул (2.45), (2.46), (2.47), (2.48), но не для функции, а для ее производной, т. е. одну из формул

$$x^{k+1} = x^k - \frac{f'(x^k)}{f''(x^k)},$$

$$x^{k+1} = x^k - \frac{1}{\alpha} \cdot f'(x^k),$$

где, например, $\alpha = f''(x^0)$,

$$x^{k+1} = x^k - \frac{x^k - x^{k-1}}{f'(x^k) - f'(x^{k-1})} \cdot f'(x^k),$$

$$x^{k+1} = x^k - \frac{f'(x^k)}{f'(x^k + f'(x^k)) - f'(x^k)} \cdot f'(x^k),$$

$$x^{k+1} = x^k - \frac{1}{f''(x^k) - \frac{f'(x^k) \cdot f'''(x^k)}{2f''(x^k)}} \cdot f'(x^k).$$

3. Вычисляем значения ЦФ и ее производных в точке x^{k+1} .

4. Если $|x^{k+1} - x^k| \leq \varepsilon_x$ и $|f'(x^{k+1})| \leq \varepsilon_y$, то полагаем $\bar{x} \approx x^{k+1}$ и переходим на шаг 5. Иначе выполняем присваивание $k = k + 1$ и переходим на шаг 2.

5. Если точка \bar{x} удовлетворяет критерию оптимальности, то принимаем ее за решение задачи: $x^* = \bar{x}$. В противном случае при заданных начальных условиях найти оптимум задачи невозможно.

2.4.3 Метод средней точки (поиск Больцано)

Пусть $f(x) \in Q[a, b]$, т. е. функция унимодальна и дифференцируема на интервале $[a, b]$. Для нахождения корня уравнения $f'(x) = 0$ можно воспользоваться эффективным алгоритмом исключения интервалов, на каждой итерации которого рассматривается лишь одна пробная точка.

Если в точке z значение функции $f'(z) < 0$, то точка x^* не может быть левее точки z . В этом случае исключается интервал $[a, z]$ (рис. 2.8).

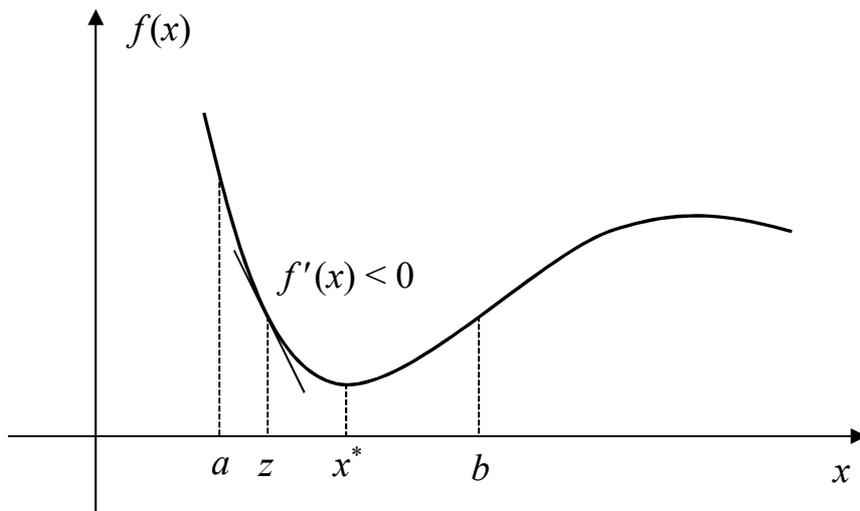


Рис. 2.8 – Поиск Больцано. Исключается интервал $[a, z]$

Если в точке z $f'(z) > 0$, то минимум x^* не может быть расположен правее z . Исключается интервал $[z, b]$ (рис. 2.9).

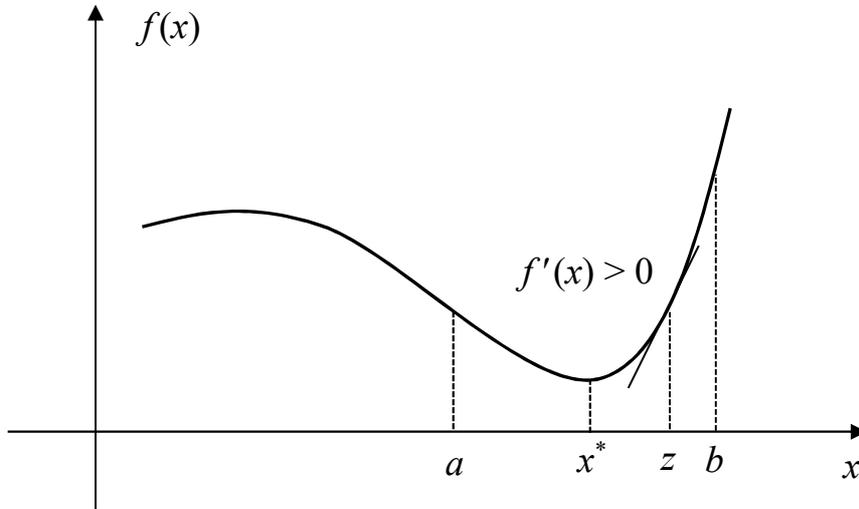


Рис. 2.9 – Поиск Больцано. Исключается интервал $[z, b]$

Исходя из определения унимодальной функции, если на интервале $[a, b]$ имеется точка минимума, то $f'(a) < 0$ и $f'(b) > 0$. Вычислим координату средней точки:

$$z = \frac{a+b}{2}$$

и найдем $f'(z)$:

$$\begin{cases} \text{если } f'(z) > 0, \text{ то исключим } [z, b]; \\ \text{если } f'(z) < 0, \text{ то исключим } [a, z]. \end{cases}$$

Формализованное описание алгоритма. Пусть задана функция $f(x)$, $x \in [a, b]$ и задана точность по аргументу ε_x и функции ε_y .

1. Положить $k = 0$, $a^0 = a$, $b^0 = b$, при этом $f'(a) < 0$, $f'(b) > 0$.

2. Вычислить $\bar{x}^k = \frac{a^k + b^k}{2}$ и значение $f'(\bar{x}^k)$.

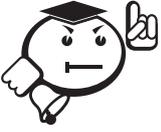
3. Если заданная точность достигнута, т. е.

$$\frac{b^k - a^k}{2} \leq \varepsilon_x \text{ и } |f'(\bar{x}^k)| \leq \varepsilon_y,$$

то закончить поиск и принять $x^* \approx \bar{x}^k$. Иначе:

$$\begin{cases} \text{если } f'(\bar{x}^k) > 0, \text{ положить } a^{k+1} = a^k, b^{k+1} = \bar{x}^k, \\ \text{иначе положить } a^{k+1} = \bar{x}^k, b^{k+1} = b^k, \end{cases}$$

положить $k = k + 1$ и перейти на шаг 2.



.....
Здесь исследуется лишь знак производной независимо от ее значения.
.....

Как можно видеть, МСТ – это обычный метод дихотомии решения уравнений с одной переменной, только предназначенный для решения уравнения $f'(x) = 0$, а не $f(x) = 0$. Аналогично можно модифицировать другие интервальные методы решения уравнений с одной неизвестной – Фибоначчи, золотого сечения и т. д.

2.5 Метод поиска с использованием кубической аппроксимации

Функция аппроксимируется полиномом третьего порядка. Схема метода аналогична схеме методов с использованием квадратичной аппроксимации, но построение полинома проводится на основе меньшего числа точек, т. к. в каждой точке можно вычислять значения и функции, и ее производной [5].

Аппроксимирующий кубический полином записывается в виде:

$$F(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2) + a_3(x - x_1)^2(x - x_2). \quad (2.50)$$

Параметры уравнения (2.50) подбираются таким образом, чтобы значения $F(x)$ и ее производной в точках x_1, x_2 совпадали со значениями $f(x), f'(x)$ в этих точках. Первая производная функции $F(x)$ равна:

$$\frac{dF(x)}{dx} = a_1 + a_2(x - x_1) + a_2(x - x_2) + a_3(x - x_1)^2 + 2a_3(x - x_1)(x - x_2). \quad (2.51)$$

Коэффициенты a_0, a_1, a_2, a_3 уравнения (2.50) определяются по известным значениям $f(x_1), f(x_2), f'(x_1), f'(x_2)$ путем решения следующей системы уравнений:

$$\begin{aligned} f_1 &= f(x_1) = a_0; \\ f_2 &= f(x_2) = a_0 + a_1(x_2 - x_1); \\ f'_1 &= f'(x_1) = a_1 + a_2(x_1 - x_2); \\ f'_2 &= f'(x_2) = a_1 + a_2(x_2 - x_1) + a_3(x_2 - x_1)^2. \end{aligned}$$

Данная система легко решается рекурсивным методом. После того как коэффициенты найдены, действуя по аналогии со случаем квадратичной аппроксимации, можно оценить координату стационарной точки $f(x)$ с помощью полинома (2.50). При этом приравнивание к нулю производной (2.51) приводит к квадратному уравнению. Используя формулу для вычисления корней квадратного уравнения, запишем решение, определяющее стационарную точку кубического полинома, в виде:

$$\bar{x} = \begin{cases} x_2, & \text{если } \mu < 0; \\ x_2 - \mu(x_2 - x_1), & \text{если } 0 \leq \mu \leq 1; \\ x_1, & \text{если } \mu > 1, \end{cases} \quad (2.52)$$

где

$$\mu = \frac{f_2' + w - z}{f_2' - f_1' + 2w}, \quad z = \left[\frac{3 \cdot (f_1 - f_2)}{x_2 - x_1} \right] + f_1' + f_2',$$

$$w = \begin{cases} \sqrt{z^2 - f_1' f_2'}, & \text{если } x_1 < x_2; \\ -\sqrt{z^2 - f_1' f_2'}, & \text{если } x_1 > x_2. \end{cases}$$

Формула для вычисления значения w обеспечивает надлежащий выбор одного из двух корней квадратного уравнения.

Для значений $\mu \in [0, 1]$ формула (2.52) гарантирует, что получаемая точка $\bar{x} \in [x_1, x_2]$.

Схема метода:

1. Выполняем присваивание $k = 0$. Если задано начальное приближение x^0 и шаг Δ , переходим на шаг 2. Если задан интервал $[a, b]$, содержащий искомую точку экстремума, переходим на шаг 3.

2. Полагаем $\bar{x}^0 = x^0$. Если $f'(\bar{x}^0) < 0$, то выполняем присваивание $x_1^0 = \bar{x}^0$. Далее ищем минимальное значение m , такое, что в точке $x_2^0 = \bar{x}^0 + 2^m \cdot \Delta$ значение производной $f'(x_2^0) > 0$ ($m = 0, 1, 2, \dots$). Затем переходим на шаг 4. Иначе (т. е. если условие $f'(\bar{x}^0) < 0$ не выполняется) выполняем присваивание $x_2^0 = \bar{x}^0$. Далее ищем минимальное значение m , такое, что в точке $x_1^0 = \bar{x}^0 - 2^m \cdot \Delta$ значение производной $f'(x_1^0) < 0$ ($m = 0, 1, 2, \dots$). Затем переходим на шаг 4. Цель этого шага – определить отрезок $[x_1^0, x_2^0]$, содержащий точку минимума, т. е. в левой границе отрезка производная должна убывать, а в правой – возрастать.

3. Выполняем присваивания: $x_1^0 = a$, $x_2^0 = b$,

$$\bar{x}^0 = \frac{a+b}{2}.$$

4. По формуле (2.52) вычисляем точку минимума \bar{x}^{k+1} полинома $F(x)$, построенного по точкам x_1^k и x_2^k . Причем, согласно схеме алгоритма, неравенство $x_1^k < x_2^k$ выполняется всегда, поэтому для определения параметра w всегда используем формулу

$$w = \sqrt{z^2 - f_1' f_2'}.$$

5. Если $f'(\bar{x}^{k+1}) < 0$, то $x_1^{k+1} = \bar{x}^{k+1}$, $x_2^{k+1} = x_2^k$. Иначе $x_1^{k+1} = x_1^k$, $x_2^{k+1} = \bar{x}^{k+1}$.

6. Если $|\bar{x}^{k+1} - \bar{x}^k| \leq \varepsilon_x$ и $|f'(\bar{x}^{k+1})| \leq \varepsilon_y$, то полагаем $x^* \approx \bar{x}^{k+1}$ и заканчиваем вычисления. Иначе выполняем присваивание $k = k + 1$ и переходим на шаг 4.

2.6 Сравнение методов

Проведем сравнение относительных эффективностей методов исключения интервалов.

Обозначим длину исходного интервала неопределенности как Δ^0 , а Δ^k – длина полученного интервала в результате n вычислений ЦФ. Тогда $FR(n) = \Delta^k / \Delta^0$ – относительное уменьшение первоначального интервала, т. е. показатель эффективности.

1. При использовании МРП на каждой итерации требуется вычислить $N-1$ значение ЦФ, где N – заданный заранее параметр. Поэтому всего за k итераций значение функции будет вычислено $n = k(N-1)$ раз, т. е.

$$k = \frac{n}{N-1}.$$

Из (2.5) имеем

$$\Delta^k = \left(\frac{2}{N}\right)^k \cdot \Delta^0 = \left(\frac{2}{N}\right)^{\frac{n}{N-1}} \cdot \Delta^0,$$

следовательно,

$$FR_{\text{МРП}}(n) = \frac{\Delta^k}{\Delta^0} = \left(\frac{2}{N}\right)^{\frac{n}{N-1}}. \quad (2.53)$$

Как выбрать значение N , чтобы поиск был оптимальным? Для этого нужно решить следующую задачу:

$$FR_{\text{МРП}}(N, n) = \left(\frac{2}{N}\right)^{\frac{n}{N-1}} \rightarrow \min, \\ N \geq 3.$$

Решением этой задачи является $N^* \approx 4,311$, ближайшими целыми значениями будут $N^* = 4$ и $N^* = 5$.

Вообще, если значение N четное, то метод можно оптимизировать. В этом случае на каждой итерации, кроме первой, требуется вычислить только $N - 2$ значения ЦФ, т. к. значение ЦФ в точке, расположенной в середине отрезка Δ^k , уже было вычислено на предыдущей итерации решения, т. е.

$$n = (N - 1) + (k - 1) \cdot (N - 2) \Rightarrow k = \frac{n - 1}{N - 2}, \\ FR_{\text{МРПЧ}}(N, n) = \frac{\Delta^k}{\Delta^0} = \left(\frac{2}{N}\right)^{\frac{n-1}{N-2}}. \quad (2.54)$$

Данная функция для четных значений N и при условии $N \geq 3$ имеет минимум в точке $N^* = 4$.

2. При использовании метода дихотомии на каждой итерации вычисляется 2 значения ЦФ (в точках x_1^k и x_2^k), поэтому

$$n = 2 \cdot k \Rightarrow k = \frac{n}{2}.$$

Из (2.8) имеем

$$\Delta^k \approx \left(\frac{1}{2}\right)^k \cdot \Delta^0 = \left(\frac{1}{2}\right)^{n/2} \cdot \Delta^0,$$

следовательно,

$$FR_{\text{МД}}(n) = \frac{\Delta^k}{\Delta^0} = \left(\frac{1}{2}\right)^{n/2}. \quad (2.55)$$

3. При использовании метода Фибоначчи на первой итерации вычисляется 2 значения ЦФ, а на каждой последующей – только одно (в силу того, что одна из точек x_1^{k+1} или x_2^{k+1} совпадает с одной из точек x_1^k или x_2^k , полученных на предыдущей итерации). Таким образом,

$$n = 2 + (k - 1) \cdot 1 \Rightarrow k = n - 1.$$

Из (2.18) имеем

$$\begin{aligned}\Delta^k &= \frac{F_{N-k}}{F_{N-k+1}} \cdot \Delta^{k-1} = \frac{F_{N-k}}{F_{N-k+1}} \cdot \frac{F_{N-k+1}}{F_{N-k+2}} \cdot \Delta^{k-2} = \dots = \\ &= \frac{F_{N-k}}{F_{N-k+1}} \cdot \frac{F_{N-k+1}}{F_{N-k+2}} \cdot \dots \cdot \frac{F_{N-1}}{F_N} \cdot \Delta^0 = \frac{F_{N-k}}{F_N} \cdot \Delta^0 = \frac{F_{N-n+1}}{F_N} \cdot \Delta^0,\end{aligned}$$

следовательно,

$$FR_{\text{МФ}}(N, n) = \frac{\Delta^k}{\Delta^0} = \frac{F_{N-n+1}}{F_N}. \quad (2.56)$$

4. При использовании метода золотого сечения на первой итерации вычисляется 2 значения ЦФ, а на каждой последующей – только одно (в силу того, что по свойствам золотого сечения одна из точек x_1^{k+1} или x_2^{k+1} совпадает с одной из точек x_1^k или x_2^k , полученных на предыдущей итерации). Таким образом,

$$n = 2 + (k - 1) \cdot 1 \Rightarrow k = n - 1.$$

Из (2.32) имеем

$$\Delta^k = \frac{1}{\gamma^k} \cdot \Delta^0 = \frac{1}{\gamma^{n-1}} \cdot \Delta^0,$$

следовательно,

$$FR_{\text{МЗС}}(n) = \frac{\Delta^k}{\Delta^0} = \frac{1}{\gamma^{n-1}}. \quad (2.57)$$

Сравним величины относительного уменьшения интервала для рассмотренных методов (табл. 2.2). Первые две строки – результаты МРП для значений параметра $N = 4$ и $N = 5$. Столбец для $n = 2$ не заполнен, т. к. при двух вычислениях значения ЦФ уменьшить интервал невозможно. Третья и четвертая строки – оптимизированный МРП для четных значений параметра N при $N = 4$ и $N = 6$. Пятая строка – результаты МД. Шестая, седьмая и восьмая строки – МФ для различных значений параметра N . При этом, если $n \geq N$, столбец не заполняется, т. к. метод уже закончит свою работу. Последняя, девятая, строка – результаты МЗС.

Видно, что МЗС наиболее эффективен в смысле точности. МФ с увеличением параметра N приближается к нему. Но здесь еще не учитывается, что нужно вычислять также сами числа Фибоначчи. МД имеет преимущество только на первой итерации, т. к. при двух вычислениях функции позволяет сократить интервал неопределенности в 2 раза, а МЗС – только в $\gamma \approx 1,618$ раз. Но в дальнейшем МД становится менее эффективен, чем МЗС и МФ. МРП – самый

неэффективный. При его оптимизации для четных значений параметра N , и выборе значения $N=4$, он немного менее эффективен, чем МД. Во всех остальных случаях его эффективность заметно ниже.

Таблица 2.2 – Величины относительного уменьшения интервала

Метод поиска	Количество вычислений значений функции				
	$n=2$	$n=5$	$n=10$	$n=15$	$n=20$
МРП, $N=5$	–	0,318	0,1012	0,0322	0,01024
МРП, $N=4$	–	0,315	0,0992	0,0313	0,00984
МРП*, $N=6$	–	0,333	0,0844	0,0214	0,00542
МРП*, $N=4$	–	0,250	0,0442	0,0078	0,00138
МД	0,500	0,177	0,0313	0,0055	0,00098
МФ, $N=6$	0,615	0,154	–	–	–
МФ, $N=11$	0,618	0,146	0,0139	–	–
МФ, $N=21$	0,618	0,146	0,0132	0,0012	0,00011
МЗС	0,618	0,146	0,0132	0,0012	0,00011

С другой стороны, можно также сравнивать количество вычислений значений функций, требуемых для достижения заданной точности ε_x . При этом нормируем длину интервала, т. е. полагаем $\Delta^0 = 1$.

1. В МРП количество итераций, требуемых для достижения заданной точности по аргументу, можно найти из (2.6):

$$\left(\frac{2}{N}\right)^k \frac{\Delta^0}{2} \leq \varepsilon_x \Rightarrow k = \left\lceil \log_{2/N} 2\varepsilon_x \right\rceil = \left\lceil \frac{\ln 2\varepsilon_x}{\ln 2/N} \right\rceil,$$

$$k = \frac{n}{N-1} \Rightarrow n = (N-1) \cdot \left\lceil \frac{\ln 2\varepsilon_x}{\ln 2/N} \right\rceil. \quad (2.58)$$

При оптимизации для четных значений N получим

$$k = \frac{n-1}{N-2} \Rightarrow n = 1 + (N-2) \cdot \left\lceil \frac{\ln 2\varepsilon_x}{\ln 2/N} \right\rceil. \quad (2.59)$$

Здесь $\lceil \cdot \rceil$ – операция округления до ближайшего целого в большую сторону.

2. В МД количество итераций, требуемых для достижения заданной точности по аргументу, можно найти из (2.9):

$$\frac{\Delta^0}{2^{k+1}} \leq \varepsilon_x \Rightarrow k = \left\lceil \log_2 \frac{1}{\varepsilon_x} - 1 \right\rceil = \left\lceil \log_2 \frac{1}{2\varepsilon_x} \right\rceil = \left\lceil \frac{\ln 2\varepsilon_x}{\ln 1/2} \right\rceil,$$

$$k = \frac{n}{2} \Rightarrow n = 2 \cdot \left\lceil \frac{\ln 2\varepsilon_x}{\ln 1/2} \right\rceil. \quad (2.60)$$

3. В МФ количество итераций, требуемых для достижения заданной точности по аргументу, можно найти из (2.20):

$$\frac{\Delta^0}{2F_N} \leq \varepsilon_x \Rightarrow F_N \geq \frac{1}{2\varepsilon_x}.$$

Находим минимальное значение F_N , удовлетворяющее этому условию, тогда $k = N$, затем

$$k = n - 1 \Rightarrow n = N + 1. \quad (2.61)$$

4. В МЗС количество итераций, требуемых для достижения заданной точности по аргументу, можно найти из (2.33):

$$\frac{\Delta^0}{2\gamma^k} \leq \varepsilon_x \Rightarrow k = \left\lceil \log_\gamma \frac{1}{2\varepsilon_x} \right\rceil = \left\lceil \frac{\ln 2\varepsilon_x}{\ln 1/\gamma} \right\rceil.$$

$$k = n - 1 \Rightarrow n = \left\lceil \frac{\ln 2\varepsilon_x}{\ln 1/\gamma} \right\rceil + 1. \quad (2.62)$$

Таблица 2.3 – Требуемые количества вычислений функции

Метод поиска	Заданная точность				
	$\varepsilon = 0,1$	$\varepsilon = 0,05$	$\varepsilon = 0,01$	$\varepsilon = 0,005$	$\varepsilon = 0,001$
МРП, $N = 5$	8	12	20	24	28
МРП, $N = 4$	9	12	18	21	27
МРП*, $N = 6$	9	13	17	21	25
МРП*, $N = 4$	7	9	13	15	19
МД	6	8	12	14	18
МФ	5	7	10	12	15
МЗС	5	6	10	11	14

Наиболее эффективны МЗС и МФ. В целом, методы в таблице расположены в порядке возрастания эффективности.

С помощью теоретических выкладок можно показать, что такие методы точного оценивания, как метод Пауэлла, метод поиска с использованием ку-

бичной аппроксимации и производных, существенно эффективнее методов исключения интервалов. Данный вывод справедлив лишь в предположении, что интервалы сходимости сравнимы между собой, а исследуемая функция является достаточно гладкой и унимодальной.

1. Если необходимо получить решение с очень высокой степенью точности, то лучшими оказываются методы на основе полиномиальной аппроксимации.
2. Если важно добиться надежной работы алгоритма, то целесообразно выбрать метод золотого сечения.
3. Поисквые методы типа метода Пауэлла следует использовать совместно с МЗС, когда возникают затруднения с реализацией соответствующих итераций на ЭВМ.

Для оценки эффективности методов обычно используются три характеристики:

- время, затраченное на получение решения;
- точность решения;
- чувствительность к изменениям параметра сходимости.

Например, метод средней точки, метод Пауэлла и поиск с использованием кубической аппроксимации могут быть исследованы при поиске минимума функции $f(x) = \sin^k x$ для различных нечетных значений k :

$$k \in [1, 79], \quad x^* = 4,71239, \quad f(x^*) = -1,0$$

для любого k .

При возрастании значения k увеличивается время решения и уменьшается точность. Однако МЗС не чувствителен к росту k .

В заключение отметим, что МЗС обладает высокой вычислительной эффективностью и простотой реализации. Методы точечного оценивания позволяют определить точки экстремума с помощью квадратичной и кубической аппроксимации целевой функции. Если интервалы сходимости сравнимы между собой, а целевая функция гладкая и унимодальная, то методы точечного оценивания сходятся значительно быстрее, чем методы исключения интервалов. Однако при исследовании мультимодальных или быстроизменяющихся функций (типа $\min f(x) = \sin^k x$, которая обладает узкими впадинами вблизи точки минимума) наиболее надежным оказывается МЗС.



Контрольные вопросы по главе 2

1. В чем заключается основная идея работы оптимизационных алгоритмов интервальной оценки: дихотомии, золотого сечения, Фибоначчи и других?
2. Почему методы полиномиальной аппроксимации являются более эффективными по сравнению с методами интервальной оценки?
3. На каком предположении основан метод квадратичной аппроксимации?
4. Какого типа должна быть целевая функция и аппроксимирующий полином в алгоритме Пауэлла?
5. Поясните графически схему работы алгоритма Пауэлла.
6. В чем заключается сущность метода оптимизации на основе кубичной аппроксимации целевой функции?
7. Какие требования предъявляются к целевой функции в методах с использованием производных (методах точечного оценивания)?
8. Поясните графически схему работы метода средней точки (Больцано).
9. Проведите сравнение эффективности одномерных методов оптимизации.
10. Являются ли методы интервальной оценки в целом более эффективными, чем методы точечного оценивания? Почему?

3 Методы поиска экстремума функции многих переменных

В предыдущей главе рассматривались задачи, где ЦФ зависела только от одного аргумента. В данной главе рассмотрим ЗО, в которой ЦФ имеет несколько оптимизируемых параметров. Такие задачи относятся к классу задач безусловной оптимизации для функции многих переменных (функции векторного аргумента), или задач многомерного поиска.

3.1 Классификация методов

Задачи, которые рассматриваются в данной главе, имеют следующий вид: найти

$$f(x) \rightarrow \min, x \in R^n, \quad (3.1)$$

т. е.

$$f(x) = f(x_1, x_2, \dots, x_n), x_i \in (-\infty, +\infty).$$

Далее будет показано, что при определенных условиях многие задачи с ограничениями типа

$$f(x) \rightarrow \min, g_i(x) \leq 0, i = 1, 2, \dots, m$$

можно свести к ЗО вида (3.1), т. е. к задаче безусловной оптимизации. Для этого используются условия Куна – Таккера, метод множителей Лагранжа, методы штрафов и т. д.

Методы, ориентированные на решение задач безусловной оптимизации, можно условно разделить на три больших класса [14]:

- 1) *методы прямого поиска* (нулевого порядка), основанные на вычислении только значений ЦФ;
- 2) *градиентные методы* (методы 1-го порядка), в которых используются точные значения градиента ЦФ;
- 3) *методы 2-го порядка*, в которых используется также гессиан целевой функции.

К отдельному классу можно отнести *методы с переменной метрикой*, где точное значение матрицы Гессе не используется, но используется ее приближение (аппроксимация).

Ни один метод или класс методов не отличается высокой эффективностью при решении оптимизационных задач различных типов. В некоторых си-

туациях вычисление значений ЦФ требует чрезмерных затрат времени. Иногда невозможно или очень трудно найти аналитическое выражение для производных целевой функции. Поэтому при использовании градиентных методов следует применять процедуру разностной аппроксимации производных. Таким образом, в каждом конкретном случае необходимо приспособлять применяемый метод к конкретным характеристикам решаемой ЗО.

3.2 Методы прямого поиска

Рассмотрим три метода прямого поиска [1]:

- 1) поиск по симплексу или s^2 -метод;
- 2) метод Хука – Дживса;
- 3) метод сопряженных направлений Пауэлла.

Достоинства данных методов: относительная простота вычислительных процедур, легкость реализации и корректировки.

Недостатки: значительные затраты времени по сравнению с градиентными методами.

Здесь не рассматриваются методы, основанные на идее исключения интервалов (Бокса, Дэвиса, Свенна, Кролака, Купера и др.), т. к. они в значительной степени уступают другим известным методам.

3.2.1 Симплексный метод

Метод поиска по симплексу был предложен в 1962 г. Спендли (W. Spendley), Хекстом (G. R. Hex) и Химсвортом (F. R. Himsworth). Этот метод называют *последовательным симплекс-методом* (ПСМ). Следует отметить, что указанный метод и другие подобные методы не имеют отношения к симплекс-методу линейного программирования, а сходство названий носит чисто случайный характер.



.....

В n -мерном евклидовом пространстве n -мерный симплекс представляет собой фигуру, образованную $n+1$ точками (вершинами), не принадлежащими одновременно ни одному пространству меньшей размерности.

.....

В одномерном пространстве симплекс есть отрезок прямой, в двумерном пространстве – треугольник; в трехмерном пространстве – треугольная пирамида (тетраэдр) и т. д.



.....

Симплекс называется **регулярным**, если расстояния между вершинами равны. В ПСМ используются регулярные симплексы. Расстояние между вершинами Δ называется **ребром симплекса**.

.....

Из любого симплекса, отбросив одну его вершину, можно получить новый симплекс, если к оставшимся вершинам добавить всего лишь одну точку.

Для оценки направления движения во всех вершинах симплекса V_i , $i=1,2,\dots,n+1$, где n – размерность вектора x , необходимо оценить значение ЦФ $F_i = f(V_i)$.

При поиске минимума наиболее целесообразно будет движение от вершины V_p с наибольшим значением F_p к противоположной грани симплекса. Шаг поиска выполняется переходом из некоторого симплекса V^k в новый симплекс V^{k+1} путем исключения вершины V_p^k и построения ее зеркального отображения \tilde{V}_p^k относительно общей грани.

Например, на рисунке 3.1 изображены линии уровня некоторой ЦФ. При этом худшее значение ЦФ в симплексе V^k – в вершине V_2^k . Поэтому зеркально отражаем эту вершину относительно грани (V_1^k, V_3^k) в точку \tilde{V}_2^k . Получим новый симплекс V^{k+1} , в котором две вершины общие с симплексом V^k ($V_1^{k+1} = V_1^k$, $V_3^{k+1} = V_3^k$), а третья вершина $V_2^{k+1} = \tilde{V}_2^k$ – зеркальное отражение V_2^k .

Многократное отражение худших вершин приводит к шаговому движению центра симплекса к цели по траектории некоторой ломаной линии (см. рис. 3.2).

Если в зеркальной точке \tilde{V}_p^k значение ЦФ оказывается больше, чем во всех вершинах симплекса V^k , то осуществляется *сжатие симплекса* – уменьшение его ребер на некоторый коэффициент пропорциональности $\alpha \in (0,1)$ относительно лучшей вершины (с наименьшим значением ЦФ).

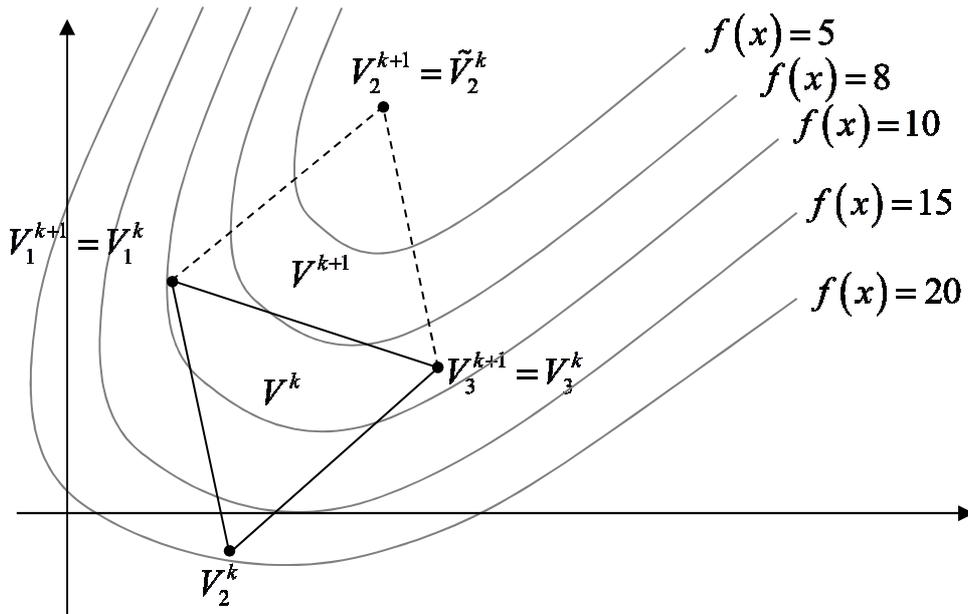


Рис. 3.1 – Зеркальное отражение симплекса

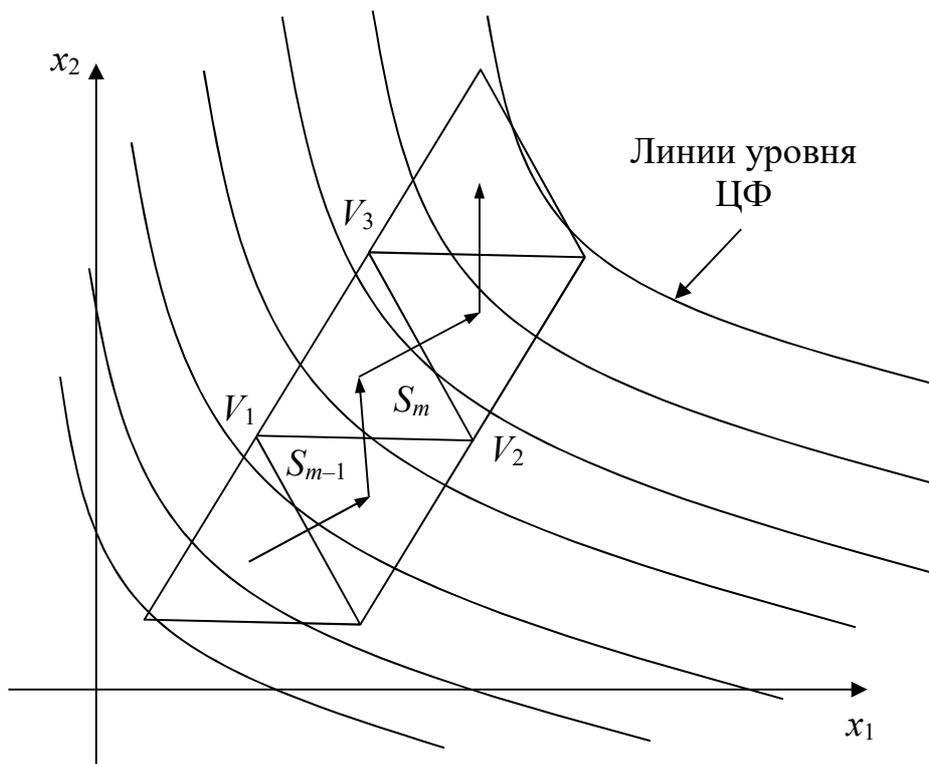


Рис. 3.2 – Траектория движения по симплексу

Например, такая ситуация рассмотрена на рисунке 3.3. Худшее значение ЦФ в симплексе V^k – в вершине V_2^k . Однако в ее зеркальном отражении \tilde{V}_2^k значение ЦФ больше, чем во всех вершинах симплекса V^k . Поэтому выбирается вершина с наименьшим значением ЦФ (в данном случае это V_3^k), и симплекс V^{k+1} получается сжатием симплекса V^k относительно этой вершины. На рисун-

ке изображено сжатие с коэффициентом $\alpha = 0,5$, т. е. длина ребер симплекса Δ была уменьшена в 2 раза.

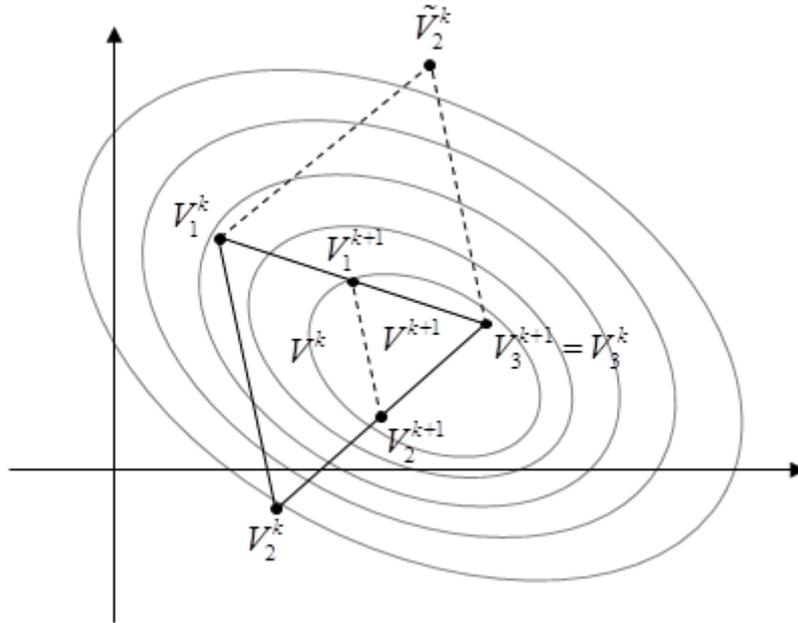


Рис. 3.3 – Сжатие симплекса

Алгоритм ПСМ. Исходные данные – ЦФ, начальная точка x^0 , длина ребра симплекса Δ^0 , коэффициент сжатия $\alpha \in (0,1)$, точность по аргументу ε_x , точность по функции ε_y .

1. Положить $k = 0$ и вычислить вершины симплекса V^0 :

$$V^0 = \begin{pmatrix} x_1^0 & x_2^0 & x_3^0 & \dots & x_n^0 \\ x_1^0 + p_n & x_2^0 + g_n & x_3^0 + g_n & \dots & x_n^0 + g_n \\ x_1^0 + g_n & x_2^0 + p_n & x_3^0 + g_n & \dots & x_n^0 + g_n \\ x_1^0 + g_n & x_2^0 + g_n & x_3^0 + p_n & \dots & x_n^0 + g_n \\ \dots & \dots & \dots & \dots & \dots \\ x_1^0 + g_n & x_2^0 + g_n & x_3^0 + g_n & \dots & x_n^0 + p_n \end{pmatrix}, \quad (3.2)$$

где n – размерность пространства аргументов R^n ,

$$p_n = \Delta^0 \cdot \frac{\sqrt{n+1} + n - 1}{n \cdot \sqrt{2}}, \quad g_n = p_n - \Delta^0 \cdot \frac{\sqrt{2}}{2} = \Delta^0 \cdot \frac{\sqrt{n+1} - 1}{n \cdot \sqrt{2}}. \quad (3.3)$$

За начальное приближение точки минимума \bar{x}^0 принять геометрический центр симплекса:

$$\bar{x}^0 = \frac{1}{n+1} \sum_{i=1}^{n+1} V_i^0. \quad (3.4)$$

Здесь V_i^0 – координаты i -й вершины симплекса V^0 , т. е. i -я строка матрицы V^0 . Матрица V^0 имеет размеры $(n+1) \times n$.

2. Вычислить значения ЦФ во всех вершинах симплекса:

$$F_i = f(V_i^k), \quad i = 1, 2, \dots, n+1.$$

3. Определить индекс p вершины с максимальным значением ЦФ:

$$p: F_p = \max_{i \in [1, n+1]} F_i.$$

4. Найти зеркальное отражение вершины V_p^k :

$$\tilde{V}_p^k = \frac{2}{n} \left[\left(\sum_{i=1}^{n+1} V_i^k \right) - V_p^k \right] - V_p^k \quad \text{или} \quad \tilde{V}_p^k = \frac{2}{n} \left(\sum_{\substack{i=1 \\ i \neq p}}^{n+1} V_i^k \right) - V_p^k. \quad (3.5)$$

5. Если $f(\tilde{V}_p^k) \leq F_p$, положить $\Delta^{k+1} = \Delta^k$ (т. е. размер симплекса не менять)

и построить новый зеркальный симплекс V^{k+1} :

$$V^{k+1} = \begin{pmatrix} V_1^k \\ \dots \\ V_{p-1}^k \\ \tilde{V}_p^k \\ V_{p+1}^k \\ \dots \\ V_{n+1}^k \end{pmatrix} = \begin{cases} V_i^k, & i \neq p, \\ \tilde{V}_p^k, & i = p. \end{cases} \quad (3.6)$$

Иначе (если $f(\tilde{V}_p^k) > F_p$) выполнить сжатие симплекса. Вычислить новую длину ребра $\Delta^{k+1} = \alpha \cdot \Delta^k$, далее определить индекс вершины m с минимальным значением ЦФ:

$$m: F_m = \min_{i \in [1, n+1]} F_i.$$

Затем построить новый симплекс:

$$V_i^{k+1} = \alpha V_i^k + (1 - \alpha) V_m^k. \quad (3.7)$$

Проверить правильность построения вершин симплекса можно следующим образом: длина всех его ребер должна быть равна Δ^{k+1} , т. е.

$$\|V_i^{k+1} - V_j^{k+1}\| = \Delta^{k+1}, \quad i = 1, 2, \dots, n+1, \quad j = 1, 2, \dots, n+1, \quad i \neq j.$$

6. Вычислить новое приближение точки оптимума:

$$\bar{x}^{k+1} = \frac{1}{n+1} \sum_{i=1}^{n+1} V_i^{k+1}. \quad (3.8)$$

В принципе, за приближение точки оптимума можно принять вершину симплекса V^{k+1} с наименьшим значением ЦФ, однако формула (3.8) в среднем обеспечивает более высокую точность.

7. Если $\|\bar{x}^{k+1} - \bar{x}^k\| \leq \varepsilon_x$ и $|f(\bar{x}^{k+1}) - f(\bar{x}^k)| \leq \varepsilon_y$, то положить $x^* \approx \bar{x}^{k+1}$ и закончить вычисления. Иначе выполнить присваивание $k = k + 1$ и вернуться на шаг 2.

3.2.2 Метод поиска Хука – Дживса

Симплекс-метод и метод Хука – Дживса относятся к категории эвристических методов.

В методе поиска по симплексу основное внимание было уделено геометрическому расположению пробных точек. Основная цель построения множества таких точек – определение направления, в котором должен вестись поиск. Расположение пробных точек влияет лишь на чувствительность направления поиска к изменениям топологических свойств целевой функции.

Стратегию поиска по симплексу можно усовершенствовать путем введения множества векторов, задающих направление поиска.



.....
Ограничение. Все рассматриваемые методы прямого поиска используют не менее n независимых направлений, где n – размерность вектора.

Например, легко убедиться, что нельзя вести поиск минимума ЦФ трех переменных с использованием двух направлений поиска.

Элементарным примером метода, реализующего процедуру рекурсивного перебора на множестве направлений поиска, является метод циклического изменения переменных – каждый раз меняется только одна переменная. Затем вдоль каждого из координатных направлений последовательно проводится поиск точки экстремума на основе методов одномерной минимизации. Однако если линии уровня ЦФ имеют овражный характер, то процедура поиска становится неэффективной и даже может привести к отсутствию сходимости к точке

локального экстремума, если изменение координатных направлений поиска осуществляется в циклическом порядке (показано Пауэллом).

Модифицированный метод Хука – Дживса позволяет существенно ускорить сходимость за счет того обстоятельства, что поиск периодически проводится в направлении $d^i = x^i - x^{i-1}$.

По существу процедура Хука – Дживса представляет собой комбинацию двух поисков:

- а) *исследующий поиск* (для выявления характера локального поведения ЦФ и определения направления движения вдоль «оврагов») с циклическим изменением переменных;
- б) *ускоряющий поиск по образцу* с использованием определенных эвристических правил.

Исследующий поиск. Выбирается некоторая исходная точка x^0 . Задается величина шага Δ_i , которая может быть различной для разных координатных направлений и изменяться в процессе поиска.

Если значение ЦФ в пробной точке меньше значения ЦФ в исходной точке, то шаг поиска успешный. В противном случае из исходной точки делается шаг в противоположном направлении. После перебора всех n координат исследующий поиск завершается. Полученная точка x_0^k называется *базовой*.

В конечном итоге возникает ситуация, когда такой поиск не приводит к успеху. В этом случае уменьшается шаг путем введения коэффициента α и возобновляется исследующий поиск.

Поиск по образцу. Осуществляется шаг из полученной базовой точки вдоль прямой, соединяющей эту точку с предыдущей базовой точкой. Новая точка образца определяется по формуле:

$$x_0^k = x_0^k + (x_0^k - x^k).$$

Как только движение по образцу не приводит к уменьшению ЦФ, точка x_0^k фиксируется. Если в результате получается точка с меньшим значением ЦФ, чем в точке x^k , то она рассматривается как новая базовая точка x^{k+1} .

Алгоритм Хука – Дживса. Исходные данные – ЦФ, начальная точка x^0 , вектор приращения Δ^0 , коэффициент сжатия $\alpha > 1$, точность по аргументу ε_x , точность по функции ε_y .

1. Положить $k = 0$, $\bar{x}^0 = x^0$, вычислить $f(\bar{x}^0)$.

2. Положить $x_n^0 = \bar{x}^k$. Провести исследующий поиск, т. е. для каждой координаты точки x_n^{i-1} , $i = 1, 2, \dots, n$ сначала выполнить шаг в сторону приращения Δ_i^k :

$$x_n^i = x_n^{i-1} + \Delta_i^k \cdot e_i, \quad (3.9)$$

где e_i — i -й единичный вектор размера n . Если шаг был удачным, т. е. $f(x_n^i) < f(x_n^{i-1})$, то положить $x_n^i = x_n^i$. Иначе сделать шаг в обратном направлении:

$$x_n^i = x_n^{i-1} - \Delta_i^k \cdot e_i. \quad (3.10)$$

Если шаг был удачным, то положить $x_n^i = x_n^i$. Если оба шага оказались неудачными, то для данной координаты исследующий поиск оказался неудачным, и $x_n^i = x_n^{i-1}$.

3. Если исследующий поиск был удачным хотя бы для одной координаты, перейти на шаг 5. Иначе перейти на шаг 4.

4. Положить $\bar{x}^{k+1} = \bar{x}^k$, уменьшить приращение $\Delta^{k+1} = \Delta^k / \alpha$, выполнить присваивание $k = k + 1$ и вернуться на шаг 2.

5. Положить $\Delta^{k+1} = \Delta^k$. За базовую точку принять $x_0^k = x_n^k$. Провести поиск по образцу:

$$x_0^k = x_0^k + (x_0^k - \bar{x}^k). \quad (3.11)$$

Если поиск по образцу был удачным, т. е. $f(x_0^k) < f(\bar{x}^k)$, то положить $\bar{x}^{k+1} = x_0^k$, иначе $\bar{x}^{k+1} = \bar{x}^k$.

6. Если $\|\Delta^{k+1}\| \leq \varepsilon_x$ и $|f(\bar{x}^{k+1}) - f(\bar{x}^k)| \leq \varepsilon_y$, то положить $x^* \approx \bar{x}^{k+1}$ и закончить вычисления. Иначе выполнить присваивание $k = k + 1$ и вернуться на шаг 2.

Достоинства метода Хука – Дживса: несложная стратегия поиска, простота вычислений, малый объем требуемой памяти (меньше, чем в симплекс-методе).

Недостатки: при наличии значительных нелинейных эффектов процесс вырождается в последовательность исследующих поисков без перехода к ускоряющему поиску по образцу.

Возможные варианты модификации метода Хука – Дживса:

а) если движение по образцу приводит к успеху, то желательно увеличить длину шага по образцу, чтобы полностью использовать возможность поиска вдоль прямой:

$$x_0^k = x_6^k + j \cdot (x_6^k - \bar{x}^k). \quad (3.12)$$

Параметр $j = 0, 1, 2, \dots$ подбирается таким образом, что

$$f(x_6^k + j \cdot (x_6^k - \bar{x}^k)) < f(x_6^k + (j+1) \cdot (x_6^k - \bar{x}^k)).$$

При дальнейшем увеличении j ЦФ больше не уменьшается. Тогда поиск по образцу будет удачным, если $j \geq 1$;

б) введение дополнительных правил увеличения и уменьшения приращенных переменных и т. д.

3.2.3 Метод сопряженных направлений Пауэлла

Данный метод и его модификации являются наиболее эффективными алгоритмами прямого поиска [2, 8].

Метод ориентирован на решение задач с квадратичными целевыми функциями и основывается на фундаментальных теоретических результатах.

Основная идея алгоритма: если квадратичная функция $f(x)$, $x = (x_1, \dots, x_n)^T$ приведена к виду суммы полных квадратов, то ее экстремум может быть найден в результате реализации n^2 одномерных поисков по преобразованным координатным направлениям.

Рассмотрим общие принципы. Пусть имеем квадратичную функцию (в окрестности точки оптимума любую нелинейную функцию можно аппроксимировать квадратичной функцией) вида

$$f(x) = \frac{1}{2} x^T A x + b^T x + c,$$

где A – положительно определенная симметричная матрица порядка $n \times n$; b – n -мерный вектор $(b_1, \dots, b_n)^T$; c – константа.

Суть методов сопряженных направлений состоит в том, чтобы начиная от точки x^0 , минимизировать последовательно $f(x)$ по n линейно-независимым направлениям d_0, d_1, \dots, d_{n-1} , обладающим свойством быть попарно сопряженными относительно квадратичной формы $f(x)$.

Это условие записывается в виде:

$$\left. \begin{array}{l} \forall i \ (0 < i < n-1), \\ \forall j \ (0 < j < n-1), \\ i \neq j, \end{array} \right\} \Rightarrow d_i^T A d_j = 0. \quad (3.13)$$

Итак, предположим, что точки x^{k+1} , $k = 0, 1, \dots, n-2$ определены, начиная с x^1 , формулой

$$x^{k+1} = x^k + \lambda_k d_k,$$

где λ_k – значение величины λ , минимизирующее значение одномерной функции $f(x^k + \lambda d_k)$. Покажем, что при этих условиях точка

$$x^n = x^0 + \sum_{j=0}^{n-1} \lambda_j d_j, \quad (3.14)$$

полученная на n -м этапе, будет экстремумом задачи, т. е. будет удовлетворять соотношениям

$$\nabla f(x^n) = Ax^n + b = 0. \quad (3.15)$$

А поскольку λ_k минимизирует функцию $f(x)$ в направлении d_k , то должно выполняться соотношение для всех k :

$$d_k^T \cdot \nabla f(x^{k+1}) = d_k^T (Ax^{k+1} + b) = 0.$$

Учитывая, что $x^{k+1} = x^k + \lambda_k d_k$, имеем

$$d_k^T A(x^k + \lambda_k d_k) + d_k^T b = 0, \quad (3.16)$$

откуда получаем

$$\lambda_k = -\frac{d_k^T (Ax^k + b)}{d_k^T A d_k}. \quad (3.17)$$

Так как A – положительно определенная матрица, то знаменатель в (3.17) отличен от 0. Учитывая, что

$$x^k = x^0 + \sum_{j=0}^{k-1} \lambda_j d_j,$$

можем снова записать:

$$d_k^T A x^k = d_k^T A x^0 + \sum_{j=0}^{k-1} \lambda_j d_k^T A d_j = d_k^T A x^0, \quad (3.18)$$

значит, λ_k определяется выражением:

$$\lambda_k = -\frac{d_k^T (Ax^0 + b)}{d_k^T Ad_k}. \quad (3.19)$$



.....
 Для любого k ($1 \leq k \leq n$) точка

$$x^k = x^0 + \sum_{j=0}^{k-1} \lambda_j d_j$$

есть оптимум функции $f(x)$, порожденный направлениями d_0, d_1, \dots, d_{k-1} , проходящими через точку x^0 . В частности, точка

$$x^n = x^0 + \sum_{j=0}^{n-1} \lambda_j d_j$$

есть оптимум функции $f(x)$ в пространстве R^n .



Доказательство. Заметим, что

$$d_i^T Ax^k = d_i^T Ax^0 + \sum_{j=0}^{k-1} \lambda_j d_i^T Ad_j.$$

Учитывая (3.13), получим:

$$d_i^T Ax^k = d_i^T Ax^0 + \lambda_i d_i^T Ad_i, \\ 0 \leq i \leq k-1.$$

Отсюда, используя (3.19), получаем:

$$d_i^T Ax^k = d_i^T Ax^0 - d_i^T (Ax^0 + b),$$

откуда с учетом (3.18)

$$d_i^T (Ax^k + b) = 0, \quad \forall i = 0, 1, \dots, k-1.$$

Получили:

$$d_i^T \cdot \nabla f(x^k) = 0, \quad i = 0, 1, \dots, k-1.$$

Так как в пространстве R^k система векторов $d_i, i = 0, 1, \dots, k-1$ образует базис, то условие ортогональности и вектора $\nabla f(x^k)$ со всеми векторами d_i возможно только в случае $\nabla f(x^k) = 0$ (т. е. градиент есть нулевой вектор).

В частности, при $k = n$ имеем n линейно независимых векторов d_0, d_1, \dots, d_{n-1} , и x^n есть экстремум для функции $f(x)$ на всем пространстве R^n .

.....

В алгоритме Пауэлла $\min f(x)$ отыскивается последовательно вдоль p сопряженных направлений d_0, d_1, \dots, d_p ($p < n$), исходя каждый раз из последней найденной точки.

Таким образом, начиная с точки x^0 , шаг за шагом строится последовательность x^1, x^2, \dots, x^p , определенная соотношениями:

$$\begin{aligned} f(x^1) &= f(x^0 + \lambda_1 d_1) = \min f(x^0 + \lambda_1 d_1); \\ f(x^2) &= f(x^1 + \lambda_2 d_2) = \min f(x^1 + \lambda_2 d_2); \\ &\dots\dots\dots \\ f(x^p) &= f(x^{p-1} + \lambda_p d_p) = \min f(x^{p-1} + \lambda_p d_p). \end{aligned}$$

Предположим, далее, что эта операция повторяется на этот раз с началом в точке $y^0 \neq x^0$, получим другую последовательность: y^1, y^2, \dots, y^p .

Тогда направление $d = y^p - x^p$ будет сопряженным по отношению к p направлениям d_1, \dots, d_p .

.....



Теорема (свойство параллельного подпространства). Пусть $f(x)$ – квадратичная функция:

$$f(x) = \frac{1}{2} x^T A x + b^T x + c,$$

A – положительно определенная матрица Гессе.

Пусть x^p (соответственно, y^p) – точка, полученная одномерной оптимизацией последовательно вдоль p сопряженных направлений d_1, \dots, d_p , исходящих из точки x^0 (соответственно y^0).

Тогда если $y^p \neq x^p$, то направление $d_{p+1} = y^p - x^p$ сопряжено по отношению к направлениям d_1, \dots, d_p .

.....



.....
Доказательство. Точки прямой, исходящей из точки x^{p-1} в направлении d_p , задаются формулой:

$$x = x^{p-1} + \lambda d_p.$$

Минимум $f(x)$ вдоль направления d_p определяется путем нахождения λ_p , при котором $df/d\lambda = 0$.

Вычислим эту производную:

$$df/d\lambda = (x^{p-1} + \lambda d_p) A d_p + b^T d_p = (Ax + b)^T d_p.$$

При $\lambda = \lambda_p$, $x = x_p$ и $df/d\lambda = 0$ можно записать выражение:

$$(Ax^p + b)^T d_p = 0. \quad (3.20)$$

Аналогично для точки y^p , определяемой уравнением:

$$y = y^{p-1} + \lambda d_p$$

при $\lambda = \lambda_p$, $y = y_p$, получим

$$(Ay^p + b)^T d_p = 0. \quad (3.21)$$

Найдем разность выражений (3.20) и (3.21):

$$(y^p - x^p)^T A d_p = 0,$$

что и требовалось доказать.

Геометрически это выглядит следующим образом. Пусть заданы квадратичная функция $f(x)$, две произвольные несовпадающие точки x^1 , x^2 и направление d . Если $x^2 = x^1 + \lambda d$ минимизирует функцию $f(x^1 + \lambda d)$ при $\lambda = \lambda_1$, а $y^2 = y^1 + \lambda d$ минимизирует $f(y^1 + \lambda d)$ при $\lambda = \lambda_2$, то направление $(y^2 - x^2)$ сопряжено с d , т. е. $(y^2 - x^2)^T A d = 0$ (рис. 3.4).

Из рисунка видно, что поиск, проводимый из точки x^2 и y^2 в направлении $(y^2 - x^2)$, обеспечивает получение экстремума функции $f(x)$.

Алгоритм метода сопряженных направлений. Исходные данные – ЦФ, начальная точка x^0 , точность по аргументу ε_x , точность по функции ε_y , а также

система n линейно независимых направлений. На начальном этапе можно положить $d_i = e_i, i = 1, 2, \dots, n$, где e_i – единичные вектора (рис. 3.5).

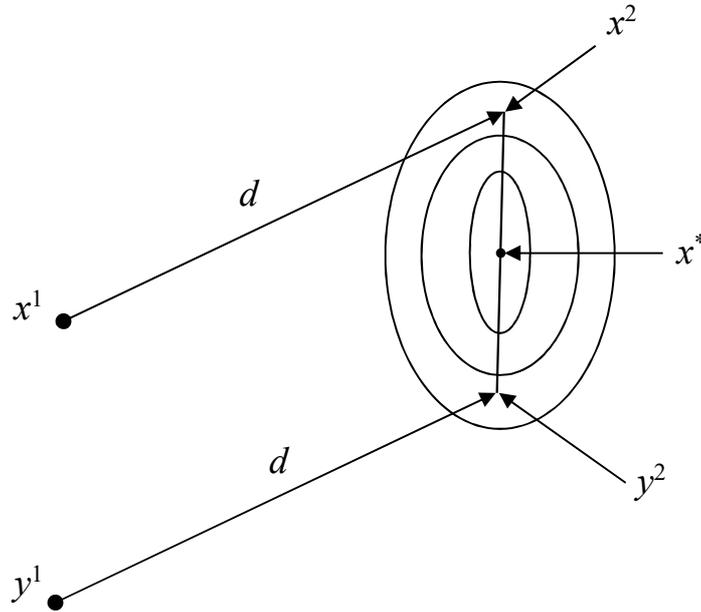


Рис. 3.4 – Система сопряженных направлений

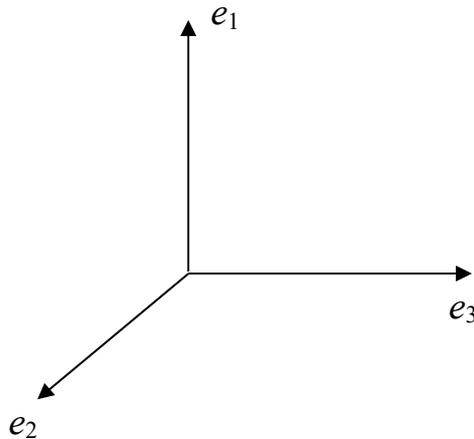


Рис. 3.5 – Система единичных векторов

1. Задать номер итерации $k = 0$, вычислить $f(x^0)$.
2. Минимизировать $f(x)$ при последовательном движении по $n + 1$ направлениям, при этом полученная ранее точка минимума берется в качестве исходной точки, а направление d_n используется как в первом, так и в последнем поиске, т. е.

$$x^{k(n+1)+1} = x^{k(n+1)} + \lambda_0 d_n, \quad \Phi(\lambda_0) = \min_{\lambda} f(x^{k(n+1)} + \lambda d_n),$$

$$x^{k(n+1)+2} = x^{k(n+1)+1} + \lambda_1 d_1, \quad \Phi(\lambda_1) = \min_{\lambda} f(x^{k(n+1)+1} + \lambda d_1),$$

$$x^{k(n+1)+3} = x^{k(n+1)+2} + \lambda_2 d_2, \quad \Phi(\lambda_2) = \min_{\lambda} f(x^{k(n+1)+2} + \lambda d_2),$$

$$\dots$$

$$x^{k(n+1)+n+1} = x^{k(n+1)+n} + \lambda_n d_n, \quad \Phi(\lambda_n) = \min_{\lambda} f(x^{k(n+1)+n} + \lambda d_n).$$

При этом, если после вычисления некоторой точки x^p выполняется критерий останова – положить $x^* \approx x^p$ и закончить вычисления. Если ЦФ квадратичная, то минимум будет найден не более чем за n^2 одномерных поисков. Поэтому, независимо от других критериев, точка x^{n^2} будет оптимальной. Однако оптимальная точка может быть найдена и быстрее. Если же ЦФ не является квадратичной, то данный критерий использовать нельзя. Поэтому следует после нахождения каждой точки x^p использовать рассмотренные ранее критерии останова: $\|x^p - x^{p-1}\| \leq \varepsilon_x$, $|f(x^p) - f(x^{p-1})| \leq \varepsilon_y$.

3. Определить новое сопряженное направление с помощью рассмотренной теоремы:

$$d_{n+1} = x^{k(n+1)+n+1} - x^{k(n+1)+1}. \quad (3.22)$$

4. Произвести смену направлений

$$d_1 = d_2, \quad d_2 = d_3, \quad \dots, \quad d_n = d_{n+1}, \quad (3.23)$$

выполнить присваивание $k = k + 1$ и перейти на шаг 2.

Скорость метода – суперлинейная.

3.3 Градиентные методы и методы второго порядка

Предполагается, что целевая функция $f(x)$ непрерывна и имеет по крайней мере непрерывные первые производные. Необходимым условием существования экстремума является наличие стационарной точки ЦФ. Таким образом, основная идея многих методов оптимизации без ограничений в пространстве R^n заключается в отыскании стационарной точки x^* , в которой градиент ЦФ $\nabla f(x^*) = 0$ [5, 10].

Эта задача эквивалентна решению нелинейной системы уравнений вида

$$\begin{cases} \frac{\partial f(x)}{\partial x_i} = 0, & i = 1, 2, \dots, n. \end{cases} \quad (3.24)$$

Можно отыскивать решение непосредственно этой системы, что приводит к методу Ньютона [1]. При этом предполагается, что функция дважды дифференцируема.

В данном случае речь идет об итерационных процессах, порождающих последовательность точек x^0, x^1, \dots, x^k , сходящихся к локальному экстремуму функции $f(x)$ в точке x^* .

На каждом k -м этапе значение x^{k+1} определяется выражением

$$x^{k+1} = x^k + \lambda_k d_k, \quad (3.25)$$

где d_k – направление перемещения, которое может быть:

- либо градиентом функции в точке x^k , т. е. $d_k = -\nabla f(x^k)$;
- либо вычисленным, исходя из направления градиента $\nabla f(x^k)$;
- либо выбранным произвольно при условии, что это будет направление спуска. В этом случае должно выполняться неравенство вида $\nabla f^T(x^k) \cdot d_k < 0$.

Здесь λ_k – параметр, характеризующий длину шага.

Способ определения d_k и λ_k на каждой итерации связан с особенностями применяемого метода. Обычно выбор λ_k осуществляется путем решения задачи минимизации $f(x)$ в направлении d_k . Поэтому при реализации изучаемых методов необходимо использовать эффективные методы одномерной оптимизации.

Градиентные методы работают следующим образом. Исходя из начальной точки x^0 , вычисляют $\nabla f(x^0)$ в точке x^0 . Поскольку $\nabla f(x^0)$ указывает направление наибольшего возрастания ЦФ, снабжаем направление, противоположное градиенту, величиной λ_0 и находим следующую точку:

$$x^1 = x^0 - \lambda_0 \frac{\nabla f(x^0)}{\|\nabla f(x^0)\|}.$$

Повторение процедуры дает последовательность точек:

$$x^{k+1} = x^k - \lambda_k \frac{\nabla f(x^k)}{\|\nabla f(x^k)\|}, \quad \forall k, \lambda_k > 0.$$

В этом семействе методов следует выделить методы градиента с заданным шагом, в которых заранее задаются значения λ_k .

Доказано, что построенная последовательность сходится к решению, т. е. $x^k \rightarrow x^*$, если выполняются два условия:

- 1) $\lambda_k \rightarrow 0$ при $k \rightarrow \infty$;
- 2) $\sum_{k=0}^{\infty} \lambda_k \rightarrow +\infty$ (например, $\lambda_k = \frac{1}{k}$).

Данная процедура (*метод расходящегося ряда*) может оказаться медленной.

3.3.1 Метод наискорейшего спуска (метод Коши)

Известный французский математик Огюстен Луи Коши первым использовал аналогичный алгоритм для решения системы линейных уравнений [2].

В широко используемом методе Коши (МК) λ_k выбираются так, чтобы минимизировать функцию:

$$\Phi(\lambda) = f(x^k + \lambda d_k), \text{ где } d_k = -\nabla f(x^k) \quad (3.26)$$

на множестве значений $\lambda \geq 0$ (одномерная минимизация).

Алгоритм Коши. Исходные данные – ЦФ и ее градиент, начальная точка x^0 , точность по аргументу ε_x , точность по функции ε_y .

1. Задать номер итерации $k = 0$, вычислить $f(x^0)$.
2. Вычислить направление спуска $d_k = -\nabla f(x^k)$.
3. Решить задачу

$$\Phi(\lambda) = f[x^k - \lambda \nabla f(x^k)] \rightarrow \min_{\lambda \geq 0},$$

т. е. найти такое λ_k , что

$$\Phi(\lambda_k) = \min_{\lambda \geq 0} f(x^k + \lambda d_k).$$

4. Положить $x^{k+1} = x^k + \lambda_k d_k$.

5. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $\|\nabla f(x^{k+1})\| \leq \varepsilon_y$, то положить $x^* \approx x^{k+1}$ и закончить вычисления. Иначе выполнить присваивание $k = k + 1$ и вернуться на шаг 2.



.....

Будем говорить, что алгоритм A глобально сходится, если для любой выбранной исходной точки x^0 последовательность $\{x^k\}$, порожденная точками $x^{k+1} \in A(x^k)$, сходится к точке, удовлетворяющей необходимым условиям оптимальности.

.....



.....

МК обладает глобальной сходимостью, но это не означает обязательного получения глобального экстремума функции $f(x)$. Если функция $f(x)$ дважды дифференцируема в стационарной точке \bar{x} и гессиан $\nabla^2 f(\bar{x})$ положительно определен, то \bar{x} – локальный минимум ЦФ. И только если $f(x)$ является выпуклой функцией, то \bar{x} является точкой глобального минимума $f(x)$.

.....

МК обладает устойчивостью, т. е. при достаточно малом значении λ_k обеспечивается выполнение неравенства

$$f(x^{k+1}) \leq f(x^k).$$

МК позволяет существенно уменьшить значение ЦФ (быстро спуститься на «дно оврага») при движении из точек, расположенных на значительных расстояниях от точки x^* , но в дальнейшем может произойти «зацикливание», поэтому МК часто используются совместно с другими градиентными методами (например, методом Ньютона) в качестве начальной процедуры.

Недостаток: для некоторых типов функций сходимость может оказаться медленной. В самом деле, если λ_k минимизирует функцию

$$\Phi(\lambda) = f(x^k + \lambda d_k),$$

то должно быть

$$\frac{d\Phi(\lambda_k)}{d\lambda} = d_k^T \nabla f(x^k + \lambda_k d_k) = d_k^T \nabla f(x^{k+1}) = 0, \quad (3.27)$$

откуда вытекает равенство:

$$d_k^T d_{k+1} = 0, \quad (3.28)$$

которое доказывает, что последовательные направления ортогональны. В случае ярко выраженной нелинейности («овражности») ЦФ происходит «зацикливание» (см. рис. 3.6).

Несмотря на то что МК не имеет большого практического значения, он реализует важнейшие шаги большинства градиентных методов.

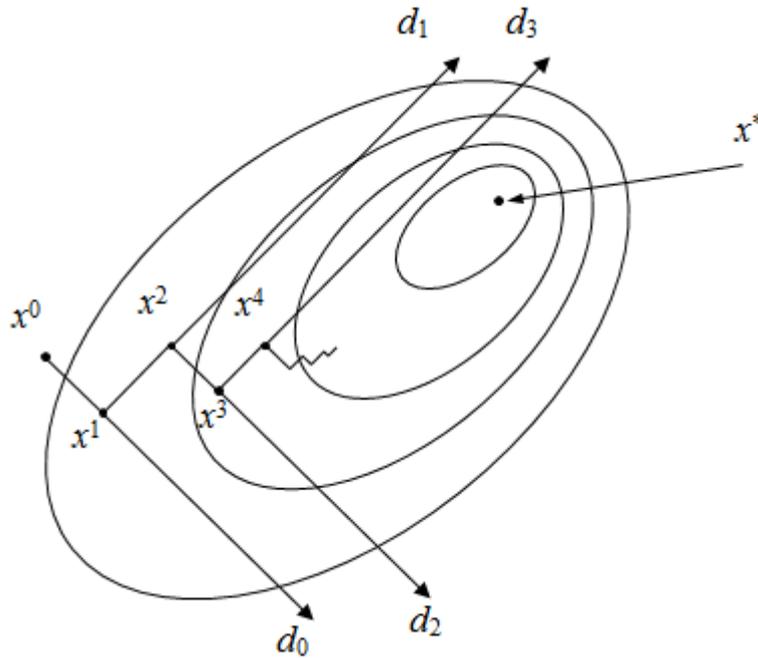
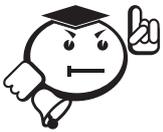


Рис. 3.6 – Метод Коши – «зацикливание» траектории спуска

3.3.2 Метод Ньютона

Итерационная формула метода Ньютона (МН) имеет вид [6, 8]:

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k). \quad (3.29)$$



В МН и направление, и шаг перемещения фиксированы, т. е. в формуле (3.25) полагаем:

$$\lambda_k = 1, \quad d_k = -[\nabla^2 f(x^k)]^{-1} \nabla f(x^k).$$

Эта формула есть не что иное, как МН в применении к решению системы нелинейных уравнений:

$$\frac{\partial f(x)}{\partial x_i} = 0, \quad i = 1, \dots, n.$$

Алгоритм Ньютона. Исходные данные – ЦФ, ее градиент и гессиан, начальная точка x^0 , точность по аргументу ε_x , точность по функции ε_y .

1. Задать номер итерации $k = 0$, вычислить $f(x^0)$.
2. Вычислить направление спуска $d_k = -[\nabla^2 f(x^k)]^{-1} \nabla f(x^k)$.
3. Положить $x^{k+1} = x^k + d_k$.

4. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $\|\nabla f(x^{k+1})\| \leq \varepsilon_y$, то положить $x^* \approx x^{k+1}$ и закончить вычисления. Иначе выполнить присваивание $k = k + 1$ и вернуться на шаг 2.

Если $f(x)$ – строго выпуклая функция, то МН сходится за одну итерацию.

Недостаток. При использовании МН на каждой итерации возникает необходимость вычисления и последующего обращения матрицы Гессе $\nabla^2 f(x^k)$.

3.3.3 Модифицированный метод Ньютона

Если функция $f(x)$ не квадратичная (или не близка по форме к квадратичной функции), то МН не отличается высоким быстродействием и надежностью. В модифицированном МН (ММН) последовательность итераций строится в соответствии с формулой [6]:

$$x^{k+1} = x^k - \lambda_k \left[\nabla^2 f(x^k) \right]^{-1} \nabla f(x^k), \quad (3.30)$$

т. е. от формулы МН (3.29) она отличается учетом длины шага λ_k .

Выбор параметра λ_k осуществляется подобно методу Коши – чтобы минимизировать функцию

$$\Phi(\lambda) = f(x^k + \lambda d_k), \text{ где } d_k = \left[\nabla^2 f(x^k) \right]^{-1} \nabla f(x^k) \quad (3.31)$$

на множестве значений $\lambda \geq 0$ (одномерная минимизация). Это гарантирует выполнение неравенства

$$f(x^{k+1}) \leq f(x^k).$$

Если ЦФ строго выпуклая, то ММН, как и МН, сходится за одну итерацию, причем минимизация функции (3.31) дает $\lambda_0 = 1$.

Модифицированный алгоритм Ньютона. Исходные данные – ЦФ, ее градиент и гессиан, начальная точка x^0 , точность по аргументу ε_x , точность по функции ε_y .

1. Задать номер итерации $k = 0$, вычислить $f(x^0)$.
2. Вычислить направление спуска $d_k = -\left[\nabla^2 f(x^k) \right]^{-1} \nabla f(x^k)$.
3. Решить задачу

$$\Phi(\lambda) = f \left[x^k - \lambda \left[\nabla^2 f(x^k) \right]^{-1} \nabla f(x^k) \right] \rightarrow \min_{\lambda \geq 0},$$

т. е. найти такое λ_k , что

$$\Phi(\lambda_k) = \min_{\lambda \geq 0} f(x^k + \lambda d_k).$$

4. Положить $x^{k+1} = x^k + \lambda_k d_k$.

5. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $\|\nabla f(x^{k+1})\| \leq \varepsilon_y$, то положить $x^* \approx x^{k+1}$ и закончить вычисления. Иначе выполнить присваивание $k = k + 1$ и вернуться на шаг 2.

Недостаток: как и в МН, на каждой итерации возникает необходимость вычисления и последующего обращения матрицы Гессе $\nabla^2 f(x^k)$.

3.3.4 Метод Марквардта

Метод Марквардта (ММ) является комбинацией методов Коши и Ньютона. В нем сочетаются положительные свойства обоих методов. Направление поиска в ММ определяется равенством

$$d_k = -[H_f(x^k) + \lambda_k E]^{-1} \nabla f(x^k), \quad (3.32)$$

где E – единичная матрица.

На начальной стадии λ_0 присваивается большое значение (например, 10^4), так что

$$-[H_f(x^0) + \lambda_0 E]^{-1} \approx -(\lambda_0 E)^{-1} = -\frac{1}{\lambda_0} E. \quad (3.33)$$

Таким образом, большим значениям λ_0 соответствует направление поиска $d_0 = -\nabla f(x^0)$, т. е. направление поиска совпадает с направлением антиградиента, как в методе Коши. Из формулы (3.33) можно заключить, что при уменьшении λ до нуля направление d_k изменяется от $-\nabla f(x^k)$ до $-H_f^{-1}(x^k)$, т. е.

$$\lim_{\lambda \rightarrow 0} [H_f(x^k) + \lambda_k E]^{-1} = H_f^{-1}(x^k), \quad (3.34)$$

т. е. получаем направление поиска $d_k = -[\nabla^2 f(x^k)]^{-1} \nabla f(x^k)$, как в методе Ньютона.

Если после первого шага получена точка с меньшим значением ЦФ (т. е. $f(x^1) < f(x^0)$), следует выбрать $\lambda_1 < \lambda_0$ (например, $\lambda_1 = \lambda_0/\beta$, где $\beta > 1$) и реализовать еще один шаг. В противном случае следует выбрать $\lambda_1 > \lambda_0$ (например, $\lambda_1 = \lambda_0\beta$, где $\beta > 1$), и вновь реализовать предыдущий шаг.

Алгоритм Марквардта. Исходные данные – ЦФ, ее градиент и гессиан, максимально допустимое количество итераций M , начальная точка x^0 , начальное значение параметра λ_0 , величина изменения шага β , точность по аргументу ε_x , точность по функции ε_y .

1. Задать номер итерации $k = 0$, вычислить $f(x^0)$.
2. Вычислить направление поиска $d_k = -[H_f(x^k) + \lambda_k E]^{-1} \nabla f(x^k)$.
3. Положить $x^{k+1} = x^k + d_k$.
4. Проверить выполнение неравенства: $f(x^{k+1}) < f(x^k)$. Если оно выполняется, перейти на шаг 5, иначе на шаг 6.
5. Положить $\lambda_{k+1} = \lambda_k / \beta$ и перейти на шаг 7.
6. Положить $\lambda_{k+1} = \lambda_k \beta$ и перейти на шаг 2.
7. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $\|\nabla f(x^{k+1})\| \leq \varepsilon_y$, или превышено допустимое количество итераций ($k \geq M$), то положить $x^* \approx x^{k+1}$ и закончить вычисления. Иначе выполнить присваивание $k = k + 1$ и вернуться на шаг 2.

Достоинства ММ. Относительная простота, ЦФ убывает от итерации к итерации, высокая скорость сходимости в окрестности точки минимума x^* , отсутствует процедура поиска вдоль прямой.

Недостатки. На каждой итерации возникает необходимость вычисления матрицы Гессе $H_f(x^k)$ и последующего обращения (иногда неоднократного) матрицы $H_f(x^k) + \lambda_k E$.

Этот метод широко используется при решении задач (например, в регрессионном анализе), в которых функция $f(x)$ может быть записана в виде суммы квадратов, т. е.

$$f(x) = f_1^2(x) + f_2^2(x) + \dots + f_m^2(x) = \sum_{i=1}^m f_i^2(x). \quad (3.35)$$

3.3.5 Методы сопряженных градиентов

Ранее рассматривались методы Коши и Ньютона. Отмечалось, что МК эффективен при поиске на значительных расстояниях от точки минимума x^* и плохо работает в окрестности x^* , тогда как МН не отличается высокой надежностью при поиске x^* из удаленной точки, однако оказывается весьма эффек-

тивными в тех случаях, когда x^k находится вблизи точки минимума. Далее будут рассмотрены алгоритмы, которые обладают положительными свойствами МК и МН и основаны на вычислении только первых производных [2, 6].

Достоинства методов сопряженных градиентов (МСГ): высокая надежность при поиске x^* из удаленной точки x^0 и быстрая сходимость в окрестности x^* .

1 МСГ для квадратичных функций

Предполагается, что квадратичная функция имеет вид:

$$f(x) = \frac{1}{2} x^T A x + b^T x + c.$$

Идея метода состоит в последовательном построении направлений d_0, d_1, \dots, d_{n-1} , взаимно сопряженных относительно матрицы A (A -сопряженных). На каждом шаге k направление d_k получается как линейная комбинация градиентов $-\nabla f(x^k)$ в точке x^k и предшествующих направлений $(d_0, d_1, \dots, d_{k-1})$, причем коэффициенты линейной комбинации выбираются так, чтобы d_k было сопряженным ко всем предшествующим направлениям.

Для удобства введем обозначения:

$$g(x) = \nabla f(x) = Ax + b, \quad d_k = -g(x^k) = -g_k. \quad (3.36)$$

Таким образом,

$$\begin{cases} g(x^0) = Ax^0 + b, \\ g(x^1) = Ax^1 + b, \end{cases}$$

где x^0 – начальная точка.

Запишем изменение градиента при переходе от точки x^0 к точке x^1 . С учетом обозначений имеем:

$$\Delta g(x) = g(x^1) - g(x^0) = A(x^1 - x^0) \text{ или } \Delta g(x) = A\Delta x, \quad (3.37)$$

что выражает свойство квадратичных функций.

Алгоритм МСГ. Исходные данные – квадратичная ЦФ, ее градиент и гессиан, начальная точка x^0 , точность по функции ε_y . Однако в этом алгоритме точность можно не задавать. Если все вычисления выполняются без погрешностей, то оптимальное решение будет найдено не более чем за n итераций (при этом должно выполняться строгое равенство $\nabla f(x^*) = 0$). Поэтому точность

можно задавать лишь на тот случай, если вычисления производятся с погрешностью (например, на ЭВМ).

1. Задать номер итерации $k = 0$, вычислить $f(x^0)$.

2. Вычислить значение градиента $g_0 = \nabla f(x^0) = Ax^0 + b$ и начальное направление спуска $d_0 = -g_0$.

3. Далее, как и в методе Коши, нужно решить задачу:

$$\Phi(\lambda) = f(x^k + \lambda d_k) \rightarrow \min_{\lambda \geq 0}.$$

Однако, учитывая свойства квадратичных функций, проводить одномерную оптимизацию для этого не требуется. Учитывая, что критерий останова одномерного поиска вдоль каждого из направлений d_k записывается в виде: $\nabla f^T(x^{k+1})d_k = 0$, параметр λ_k можно оценить из выражения:

$$\lambda_k = -\frac{g_k^T d_k}{d_k^T A d_k} = -\frac{\nabla f_k^T d_k}{d_k^T A d_k}. \quad (3.38)$$

Поэтому данный метод позволяет найти точное решение ЗО, если все вычисления выполнять без погрешностей.

4. Положить $x^{k+1} = x^k + \lambda_k d_k$.

5. Если $\|\nabla f(x^{k+1})\| = 0$ или $k + 1 \geq n$ и $\|\nabla f(x^{k+1})\| \leq \varepsilon_y$, то положить $x^* \approx x^{k+1}$ и закончить вычисления. Иначе перейти на шаг 6. Если все вычисления проводятся точно, то метод гарантирует нахождение оптимального решения не более чем за n итераций.

6. Вычислить новое направление спуска по формуле:

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad (3.39)$$

где $g_{k+1} = \nabla f(x^{k+1}) = Ax^{k+1} + b$, а параметр β_k находится из условия $d_{k+1}^T A d_k = 0$ (т. е. чтобы направление d_{k+1} было A -сопряжено со всеми построенными ранее направлениями):

$$\beta_k = \frac{g_{k+1}^T A d_k}{d_k^T A d_k} = \frac{\nabla f_{k+1}^T A d_k}{d_k^T A d_k}. \quad (3.40)$$

7. Выполнить присваивание $k = k + 1$. Вернуться на шаг 3.

В целом, имеем эффективный метод для решения оптимизационных задач с квадратичными ЦФ.

2 Метод Флетчера – Ривза

Флетчер (R. Fletcher) и Ривз (C. M. Reeves) расширили предшествующий метод на случай произвольных функций. В применении к квадратичным функциям он становится равносильным методу сопряженных градиентов.

Достоинство метода Флетчера – Ривза (МФР): скорость сходимости значительно выше классических градиентных методов. Данный метод полезен для решения задач большой размерности.

Алгоритм метода Флетчера – Ривза. Исходные данные – ЦФ и ее градиент, начальная точка x^0 , точность по аргументу ε_x , точность по функции ε_y .

1. Задать номер итерации $k = 0$, вычислить $f(x^0)$.
2. Вычислить начальное направление спуска $d_0 = -\nabla f(x^0)$.
3. Решить задачу:

$$\Phi(\lambda) = f(x^k + \lambda d_k) \rightarrow \min_{\lambda \geq 0},$$

т. е. найти такое λ_k , что

$$\Phi(\lambda_k) = \min_{\lambda \geq 0} f(x^k + \lambda d_k).$$

4. Положить $x^{k+1} = x^k + \lambda_k d_k$.
5. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $\|\nabla f(x^{k+1})\| \leq \varepsilon_y$, то положить $x^* \approx x^{k+1}$ и закончить вычисления. Иначе перейти на шаг 6.
6. Вычислить новое направление спуска по формуле

$$d_{k+1} = -\nabla f(x^{k+1}) + \beta_k d_k, \quad (3.41)$$

где

$$\beta_k = \frac{\|\nabla f(x^{k+1})\|^2}{\|\nabla f(x^k)\|^2}. \quad (3.42)$$

7. Выполнить присваивание $k = k + 1$. Вернуться на шаг 3.

3 Метод Полака – Рибьера

Метод Полака – Рибьера (МПР) основан на точной процедуре проведения поиска вдоль прямой и на более общем предположении об аппроксимации ЦФ.

Алгоритм метода Полака – Рибьера. Исходные данные – ЦФ и ее градиент, начальная точка x^0 , точность по аргументу ε_x , точность по функции ε_y .

1. Задать номер итерации $k = 0$, вычислить $f(x^0)$.
2. Вычислить начальное направление спуска $d_0 = -\nabla f(x^0)$.
3. Решить задачу:

$$\Phi(\lambda) = f(x^k + \lambda d_k) \rightarrow \min_{\lambda \geq 0},$$

т. е. найти такое λ_k , что

$$\Phi(\lambda_k) = \min_{\lambda \geq 0} f(x^k + \lambda d_k).$$

4. Положить $x^{k+1} = x^k + \lambda_k d_k$.
5. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $\|\nabla f(x^{k+1})\| \leq \varepsilon_y$, то положить $x^* \approx x^{k+1}$ и закончить вычисления. Иначе перейти на шаг 6.
6. Вычислить новое направление спуска по формуле:

$$d_{k+1} = -\nabla f_{k+1} + \beta_k d_k, \quad (3.43)$$

где

$$\beta_k = \frac{\nabla f_{k+1}^T \Delta f_k}{\|\nabla f_k\|^2}, \quad (3.44)$$

$$\nabla f_k = \nabla f(x^k), \quad \nabla f_{k+1} = \nabla f(x^{k+1}), \quad \Delta f_k = \nabla f_{k+1} - \nabla f_k.$$

7. Выполнить присваивание $k = k + 1$. Вернуться на шаг 3.

Легко видеть, что единственное различие между МФР и МПР заключается в способе выбора параметра β . Причем отличия начинаются только на второй итерации решения.

4 Метод Миля – Кентрелла

Метод Миля – Кентрелла (ММК) называется *градиентным методом с памятью*, очень эффективен по числу необходимых итераций, но требует большого числа вычислений компонент градиента (или хранения их в памяти ЭВМ).

Алгоритм метода Миля – Кентрелла. Исходные данные – ЦФ и ее градиент, начальная точка x^0 , точность по аргументу ε_x , точность по функции ε_y .

1. Задать номер итерации $k = 0$, вычислить $f(x^0)$.
2. Вычислить начальное направление спуска $d_0 = -\nabla f(x^0)$.
3. Решить задачу:

$$\Phi(\lambda) = f(x^k + \lambda d_k) \rightarrow \min_{\lambda \geq 0},$$

т. е. найти такое λ_k , что

$$\Phi(\lambda_k) = \min_{\lambda \geq 0} f(x^k + \lambda d_k).$$

4. Положить $x^{k+1} = x^k + \lambda_k d_k$.

5. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $\|\nabla f(x^{k+1})\| \leq \varepsilon_y$, то положить $x^* \approx x^{k+1}$ и закончить вычисления. Иначе перейти на шаг 6.

6. Вычислить новое направление спуска по формуле:

$$d_{k+1} = -\nabla f(x^{k+1}) - \sum_{i=0}^k \beta_i \nabla f(x^i). \quad (3.45)$$

где

$$\beta_k = \frac{\|\nabla f(x^{k+1})\|^2}{\|\nabla f(x^k)\|^2}. \quad (3.46)$$

При вычислении нового направления спуска учитываются значения градиента во всех предыдущих точках x^k , поэтому метод и называется методом с памятью.

7. Выполнить присваивание $k = k + 1$. Вернуться на шаг 3.

Единственное различие между МФР и ММК заключается в способе вычисления направления спуска d_k . Причем отличия начинаются только на второй итерации решения. На первой итерации метод аналогичен МФР и МПР.

3.3.6 Квазиньютоновские методы (методы с переменной метрикой)

Квазиньютоновские методы (КМ) также основаны на свойствах квадратичных функций. Данные методы обладают положительными чертами метода Ньютона, однако используют только первые производные [2, 4, 8–9]. С чем это связано?

Итерационный поиск по методу Ньютона осуществлялся по формуле (модифицированный метод с длиной шага):

$$x^{k+1} = x^k - \lambda_k \left[\nabla^2 f(x^k) \right]^{-1} \nabla f(x^k). \quad (3.47)$$

Трудность может возникнуть, когда матрица Гессе $\nabla^2 f(x^k) = H_f(x^k)$ не является положительно определенной. В этом случае направление перемещения

$$d_k = -[\nabla^2 f(x^k)]^{-1} \nabla f(x^k) = -H_f^{-1}(x^k) \nabla f(x^k) \quad (3.48)$$

может не быть направлением спуска и глобальная сходимость метода не будет обеспечена.

В таких ситуациях обратную матрицу Гессе $H_f^{-1}(x^k)$ заменяют положительно определенной матрицей A^k , дающей направление перемещения, исходя из градиента $\nabla f(x^k)$. Отсюда получаем итерационную формулу:

$$x^{k+1} = x^k - \lambda_k A^k \nabla f(x^k), \quad (3.49)$$

где λ_k выбирается так, чтобы минимизировать функцию $f(x^{k+1})$ в направлении $d_k = -A^k \nabla f(x^k)$.

Очевидно, что матрица A^k на каждой итерации модифицируется так, чтобы для каждой квадратичной функции вида:

$$f(x) = \frac{1}{2} x^T \tilde{A} x + b^T x + c$$

(с положительно определенной матрицей \tilde{A}) матрицы A^k сходились к обращению \tilde{A}^{-1} матрицы Гессе функции $f(x)$. Следовательно, на конечном этапе сходимости мы вновь придем к методу Ньютона.

Если метод применяется к произвольной функции, то A^k может рассматриваться на каждом шаге как аппроксимация (положительно определенная) обращения матрицы Гессе функции $f(x)$.

Для аппроксимации матрицы $H_f^{-1}(x^k)$ пользуются следующим рекуррентным соотношением:

$$A^{k+1} = A^k + A_c^k, \quad (3.50)$$

где A_c^k – корректирующая матрица.

Матрица A^k будет использоваться в формулах (3.47)–(3.49). Задача заключается в том, чтобы построить матрицу A^k таким образом, чтобы последовательность A^0, A^1, \dots, A^{k+1} давала приближение к $H_f^{-1}(x^*)$. При этом для получения решения x^* требуется один дополнительный поиск вдоль прямой, если $f(x)$ – квадратичная функция. Данный подход приводит к успеху при решении задач с нелинейными ЦФ общего вида.

Еще раз напомним свойства квадратичных функций:

$$\begin{cases} \nabla f(x) = \tilde{A}x + b = g(x), \\ g(x^0) = \tilde{A}x^0 + b, \\ g(x^1) = \tilde{A}x^1 + b. \end{cases}$$

Изменение градиента при переходе из точки x^0 в точку x^1 выражается соотношением

$$\Delta g(x) = g(x^1) - g(x^0) = \tilde{A}(x^1 - x^0)$$

или

$$\Delta g(x) = \tilde{A}\Delta x \Rightarrow \Delta x = \tilde{A}^{-1}\Delta g. \quad (3.51)$$

Предположим, что матрица \tilde{A}^{-1} аппроксимируется по формуле $\tilde{A}^{-1} = \beta A^k$, где β – скалярная величина. Наиболее предпочтительным является приближение, удовлетворяющее соотношению (3.51), то есть $\Delta x^k = A^k \Delta g^k$. Однако построить такую аппроксимацию невозможно, т. к. для того чтобы найти Δg^k , необходимо знать матрицу A^k . Здесь используются следующие обозначения:

$$\Delta x^k = x^{k+1} - x^k, \quad \Delta g^k = g(x^{k+1}) - g(x^k).$$

С другой стороны, можно потребовать, чтобы новое приближение удовлетворяло также формуле (3.51) с учетом свойств квадратичных функций:

$$\Delta x^k = A^{k+1} \Delta g^k, \quad A^{k+1} = A^k + A_c^k. \quad (3.52)$$

Подставляя (3.50) в (3.52), получим:

$$\Delta x^k = (A^k + A_c^k) \Delta g^k = A^k \Delta g^k + A_c^k \Delta g^k.$$

Выразим отсюда $A_c^k \Delta g^k$:

$$A_c^k \Delta g^k = \Delta x^k - A^k \Delta g^k. \quad (3.53)$$

С помощью непосредственной подстановки можно убедиться, что матрица

$$A_c^k = \left(\frac{\Delta x^k y^T}{y^T \Delta g^k} \right) - \frac{A^k \Delta g^k z^T}{z^T \Delta g^k} \quad (3.54)$$

является решением этого уравнения (3.53) (для этого надо (3.54) умножить справа на Δg^k , и мы получим выражение (3.53)). Здесь y и z – произвольные векторы, т. е. (3.54) определяет некоторое семейство решений.

Другой способ получения матрицы A_c^k . В зависимости от того, имеет матрица A_c^k ранг 1 или ранг 2, будем говорить о коррекции ранга 1 или ранга 2. Рассмотрим коррекцию ранга 1:

$$A_c^k = \beta_k u_k u_k^T, \quad (3.55)$$

где β_k – скаляр, u_k – вектор, выбранные так, чтобы выполнялось соотношение

$$A^{k+1} \Delta g^k = \Delta x^k. \quad (3.56)$$

Обозначим $\delta_k = \Delta x^k = x^{k+1} - x^k$, $\gamma_k = \Delta g^k = \nabla f_{k+1} - \nabla f_k$. Покажем, как определить β_k и u_k , чтобы $A^{k+1} \gamma_k = \delta_k$. Используя ранее введенное выражение (3.50) $A^{k+1} = A^k + A_c^k$, перепишем (3.56) с учетом (3.55):

$$\left[A^k + \beta_k (u_k u_k^T) \right] \gamma_k = \delta_k. \quad (3.57)$$

Умножим скалярно обе части выражения (3.57) на γ_k , получим:

$$\gamma_k^T A^k \gamma_k + \beta_k (\gamma_k^T u_k) (u_k^T \gamma_k) = \gamma_k^T \delta_k. \quad (3.58)$$

Отсюда

$$\beta_k (u_k^T \gamma_k)^2 = \gamma_k^T (\delta_k - A^k \gamma_k). \quad (3.59)$$

Используем тождество

$$\beta_k (u_k u_k^T) = \frac{(\beta_k u_k u_k^T \gamma_k) (\beta_k u_k u_k^T \gamma_k)^T}{\beta_k (u_k^T \gamma_k)^2} \quad (3.60)$$

и, осуществив замену $\beta_k u_k u_k^T \gamma_k$ на $\delta_k - A^k \gamma_k$ (см. (3.57)), $\beta_k (u_k u_k^T)^2$ на $\gamma_k (\delta_k - A^k \gamma_k)$ (см. (3.59)), получим формулу коррекции (ранга 1):

$$A_c^k = \beta_k u_k u_k^T = \frac{(\delta_k - A^k \gamma_k) (\delta_k - A^k \gamma_k)^T}{\gamma_k^T (\delta_k - A^k \gamma_k)}. \quad (3.61)$$

Далее доказывается теорема о том, что для квадратичной функции $f(x)$ с положительно определенной матрицей A последовательность матриц A^k сходится не более чем за n этапов к обращению матрицы A^{-1} гессиана функции $f(x)$.

1 Метод Дэвидона – Флетчера – Пауэлла

Метод Дэвидона – Флетчера – Пауэлла (МДФП) использует матрицу коррекции ранга 2. В формуле (3.54) положим $y = \Delta x^k$ и $z = A^k \Delta g^k$. Тогда получим:

$$\begin{cases} A_c^k = \frac{\Delta x^k (\Delta x^k)^T}{(\Delta x^k)^T \Delta g^k} - \frac{A^k \Delta g^k (\Delta g^k)^T (A^k)^T}{(\Delta g^k)^T (A^k)^T \Delta g^k}, \\ A^{k+1} = A^k + A_c^k. \end{cases} \quad (3.62)$$

Данная рекуррентная формула сохраняет свойства симметрии и положительной определенности матриц, поэтому если A^0 – симметричная положительно определенная матрица, то матрицы A^1, A^2, \dots будут так же симметричными положительно определенными матрицами (см. ниже теорему).

Обычно удобно выбирать $A^0 = E$, где E – единичная матрица. Точка x^{k+1} получается из x^k перемещением в направлении $d_k = -A^k \nabla f(x^k)$. В формуле (3.62)

$$\Delta x^k = x^{k+1} - x^k; \quad \Delta g^k = \nabla f(x^{k+1}) - \nabla f(x^k).$$

Обозначим $\gamma_k = \Delta g^k$, $\delta_k = \Delta x^k$.



.....
Теорема. Пусть матрицы A^k положительно определены. Тогда, при условии $\delta_k^T \gamma_k > 0$, матрица $A^{k+1} = A^k + A_c^k$, полученная по формуле (3.62), положительно определена. Эти условия выполняются, если точка x^{k+1} получается из x^k одномерной минимизацией в направлении $d_k = -A^k \nabla f(x^k)$.



.....
Доказательство. Пусть x – произвольный вектор, $x \neq 0$. Требуется показать, что $x^T A^{k+1} x > 0$. Имеем из (3.62):

$$x^T A^{k+1} x = x^T A^k x + \frac{(x^T \delta_k)^2}{\delta_k^T \gamma_k} - \frac{(x^T A^k \gamma_k)^2}{\gamma_k^T A^k \gamma_k}. \quad (3.63)$$

В (3.62) полагаем, что A^k – симметричная матрица. Далее введем обозначения: $u = (A^k)^{1/2} \times x$, $v = (A^k)^{1/2} \times \gamma_k$. $(A^k)^{1/2}$ существует, т. к. A^k – положительно определенная симметричная матрица. Тогда (3.63) имеет вид

$$x^T A^{k+1} x = \frac{(u^T u)(v^T v) - (u^T v)^2}{(v^T v)} + \frac{(\delta_k^T x)^2}{\delta_k^T \gamma_k}. \quad (3.64)$$

Первое слагаемое справа ≥ 0 , так как, согласно неравенству Коши – Буняковского, $(u^T v)^2 \leq (u^T u)(v^T v)$. Второе слагаемое будет больше нуля, если $\delta_k^T \gamma_k > 0$. С другой стороны, оба слагаемые не могут одновременно обращаться в нуль. В самом деле, если первый член равен нулю, то

$$u = \lambda v, (\lambda \neq 0) \Rightarrow x = \lambda \gamma_k.$$

Но тогда $\delta_k^T x = \lambda \delta_k^T \gamma_k \neq 0$ и второй член отличен от нуля. Первая часть теоремы доказана.

Покажем, в частности, что $\delta_k^T \gamma_k > 0$, если x^{k+1} получена из x^k одномерной минимизацией в направлении $d_k = -A^k \nabla f(x^k)$. Имеем:

$$\delta_k = x^{k+1} - x^k = \Theta d_k = -\Theta A^k \nabla f(x^k), \Theta > 0;$$

$$d_k^T \nabla f(x^{k+1}) = 0 \Rightarrow \delta_k^T \nabla f(x^{k+1}) = 0.$$

При этих условиях

$$\delta_k^T \gamma_k = \delta_k^T (\nabla f(x^{k+1}) - \nabla f(x^k)) = -\delta_k^T \nabla f(x^k) = \Theta \nabla f^T(x^k) A^k \nabla f(x^k) > 0,$$

что и требовалось доказать.

.....

Свойство сохранения положительной определенности матриц существенно, ибо оно гарантирует, что направления d_k , последовательно порождаемые алгоритмом, являются направлениями спуска. Действительно, первая вариация $f(x)$ равна $\Delta f(x) = \nabla f^T(x^k) \Delta x$. Используя формулы (3.47) и (3.48), получаем:

$$\Delta f(x^k) = -\lambda_k \nabla f^T(x^k) A^k \nabla f(x^k) < 0,$$

откуда $f(x^{k+1}) < f(x^k)$ при любых $\lambda_k > 0$, если матрица A^k положительно определена.

Алгоритм метода Дэвидона – Флетчера – Пауэлла. Исходные данные – ЦФ и ее градиент, начальная точка x^0 , точность по аргументу ε_x , точность по функции ε_y .

1. Задать номер итерации $k = 0$, вычислить $f(x^0)$, положить $A^0 = E$.
2. Вычислить направление спуска $d_k = -A^k \nabla f(x^k)$.
3. Решить задачу:

$$\Phi(\lambda) = f\left[x^k - \lambda A^k \nabla f(x^k)\right] \rightarrow \min_{\lambda \geq 0},$$

т. е. найти такое λ_k , что

$$\Phi(\lambda_k) = \min_{\lambda \geq 0} f(x^k + \lambda d_k).$$

4. Положить $x^{k+1} = x^k + \lambda_k d_k$.

5. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $\|\nabla f(x^{k+1})\| \leq \varepsilon_y$, то положить $x^* \approx x^{k+1}$ и закончить вычисления. Иначе перейти на шаг 6.

6. Вычислить новое приближение обращенного гессиана:

$$A^{k+1} = A^k + A_c^k, \quad A_c^k = \frac{\Delta x^k (\Delta x^k)^T}{(\Delta x^k)^T \Delta g^k} - \frac{A^k \Delta g^k (\Delta g^k)^T (A^k)^T}{(\Delta g^k)^T (A^k)^T \Delta g^k}.$$

7. Выполнить присваивание $k = k + 1$ и вернуться на шаг 2.

Достоинства метода. МДФП – наиболее широко используемый градиентный метод. Устойчивость – при решении самых различных задач, возникающих на практике.

Недостаток. Необходимость хранения матрицы A^k порядка $n \times n$.



Теорема. Алгоритм МДФП в применении к квадратичной функции (предполагается, что матрица Гессе A положительно определена): порождает направления $\delta_0, \delta_1, \dots, \delta_k$, которые при любом k удовлетворяют соотношению:

$$\delta_i^T A \delta_j = 0; \quad 0 \leq i < j \leq k; \quad (3.65)$$

$$A^{k+1} A \delta_i = \delta_i, \quad 0 \leq i \leq k. \quad (3.66)$$



Доказательство. Обозначим

$$\delta_k = x^{k+1} - x^k, \quad \gamma_k = \nabla f(x^{k+1}) - \nabla f(x^k).$$

Тогда соотношения (3.62) переписутся в следующем виде:

$$\begin{cases} A_c^k = \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{A^k \gamma_k \gamma_k^T A^k}{\gamma_k^T A^k \gamma_k}, \\ A^{k+1} = A^k + A_c^k. \end{cases} \quad (3.67)$$

Умножим справа на $A \delta_k$, получим для любого k :

$$A^{k+1}A\delta_k = A^{k+1}\gamma_k = A^k\gamma_k + \frac{\delta_k\delta_k^\top\gamma_k}{\delta_k^\top\gamma_k} - \frac{A^k\gamma_k\gamma_k^\top A^k\gamma_k}{\gamma_k^\top A^k\gamma_k} = \delta_k.$$

Значит, соотношения (3.65) и (3.66) верны при $k=0$. Допустив, что они верны и при $k=1$, докажем их справедливость для произвольного значения k . При $0 \leq i < k$ имеем

$$\nabla f(x^k) - \nabla f(x^i) = \gamma_i + \gamma_{i+1} + \dots + \gamma_{k-1}.$$

Гессиан функции $f(x)$ постоянен и равен A , поэтому $A\delta_i = \gamma_i$, $\forall i$, и значит,

$$\nabla f(x^k) - \nabla f(x^i) = A(\delta_i + \delta_{i+1} + \dots + \delta_{k-1}).$$

А поскольку x^i есть оптимум $f(x)$ в направлении δ_{i-1} , то $\delta_{i-1}^\top \nabla f(x^i) = 0$. Следовательно, по предположению индукции:

$$\delta_{i-1}^\top \nabla f(x^k) = \delta_{i-1}^\top A(\delta_i + \delta_{i+1} + \dots + \delta_{k-1}) = 0, \quad i=1, 2, \dots, k-1.$$

В силу (3.66) и предположения индукции $\delta_{i-1} = A_k A \delta_{i-1}$. Отсюда

$$\delta_{i-1}^\top A A^k \nabla f(x^k) = 0, \quad \forall i, \quad 1 \leq i \leq k.$$

А поскольку

$$\delta_k = x^{k+1} - x^k = -\Theta A^k \nabla f(x^k), \quad \Theta > 0,$$

то из этого следует вывод, что $\delta_{i-1}^\top A \delta_k = 0$, $\forall i=1, 2, \dots, k$, и, стало быть, соотношение (3.50) выполняется для произвольных значений k .

Докажем теперь, что $A^{k+1}A\delta_i = \delta_i$, $\forall i=0, 1, \dots, k-1$ (как доказано выше, для $i=k$ это верно). Имеем:

$$A^{k+1}A\delta_i = A^k A\delta_i + \frac{\delta_k\delta_k^\top A\delta_i}{\delta_k^\top\gamma_k} - \frac{A^k\gamma_k\gamma_k^\top A^k A\delta_i}{\gamma_k^\top A^k\gamma_k}.$$

Второй член справа равен нулю, т. к. $\delta_k^\top A\delta_i = 0$, $i \leq k-1$. Третий член – тоже нуль, т. к. по предположению индукции $A^k A\delta_i = \delta_i$ и $\gamma_k^\top = \delta_k^\top A$, и значит

$$\gamma_k^\top A^k A\delta_i = \gamma_k^\top \delta_i = \delta_k^\top A\delta_i = 0, \quad i \leq k-1.$$

Следовательно,

$$A^{k+1}A\delta_i = A^k A\delta_i = \delta_i \quad 0 \leq i \leq k.$$

.....

Теорема показывает, что в квадратичном случае направления $\delta_0, \delta_1, \dots, \delta_k$, порожденные алгоритмом, последовательно сопряжены относительно матрицы A квадратичной формы. В этом случае алгоритм сходится не более чем за n итераций.

Наконец, при $k = n - 1$ соотношение (3.66) дает

$$A^n A\delta_i = \delta_i \quad i = 0, 1, \dots, n - 1,$$

а поскольку δ_i линейно независимы (т. к. они последовательно сопряжены относительно A), то отсюда получим:

$$A^n A = E \Rightarrow A^n = A^{-1}.$$



.....
Квадратичная форма имеет вид:

$$Q(x) = x^T A^{-1} x.$$

.....

Матрица A плохо обусловлена, если малые изменения вектора x порождают относительно большие изменения Q . Это связано с тем, что A является почти вырожденной матрицей: $|A| \rightarrow 0$.

Число обусловленности матрицы:

$$K(A) = \left| \frac{\lambda_{\max}}{\lambda_{\min}} \right| = \text{cond } A,$$

где λ_{\max} , λ_{\min} – максимальные и минимальные собственные числа матрицы A . Для плохо обусловленной матрицы $K(A) \square 1 \rightarrow \infty$. Для хорошо обусловленной матрицы $K(A) \rightarrow 1$.

2 Метод Бройдена – Флетчера – Гольдфарба – Шенно

В (3.67) матрица A^{k+1} удовлетворяет соотношению

$$A^{k+1}\gamma_k = \delta_k. \quad (3.68)$$

Если переставить местами δ_k и γ_k в (3.67) и рассмотреть последовательность матриц

$$G^{k+1} = G^k + \frac{\gamma_k \gamma_k^T}{\gamma_k^T \delta_k} - \frac{G^k \delta_k \delta_k^T G^k}{\delta_k^T G^k \delta_k}, \quad (3.69)$$

то полученные матрицы будут удовлетворять соотношению, обратному (3.67):

$$G^{k+1}\delta_k = \gamma_k. \quad (3.70)$$

Таким образом, выражение (3.69) позволяет построить аппроксимацию самого гессиана (а не его обращения).

Получим формулу для аппроксимации обращения гессиана из (3.69). После несложных выкладок можно получить выражение:

$$(G^{k+1})^{-1} = (G^k)^{-1} + \left[1 + \frac{\gamma_k^T (G^k)^{-1} \gamma_k}{\delta_k^T \gamma_k} \right] \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{\delta_k \gamma_k^T (G^k)^{-1} + (G^k)^{-1} \gamma_k \delta_k^T}{\delta_k^T \gamma_k}. \quad (3.71)$$

Заменяя $(G^k)^{-1}$ на A^k , получим:

$$A^{k+1} = A^k + \left[1 + \frac{\gamma_k^T A^k \gamma_k}{\delta_k^T \gamma_k} \right] \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{\delta_k \gamma_k^T A^k + A^k \gamma_k \delta_k^T}{\delta_k^T \gamma_k} \quad (3.72)$$

или

$$A^{k+1} = \left[E - \frac{\delta_k \gamma_k^T}{\delta_k^T \gamma_k} \right] A^k \left[E - \frac{\delta_k \gamma_k^T}{\delta_k^T \gamma_k} \right]^T + \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k}. \quad (3.73)$$

Алгоритм метода Бroyдена – Флетчера – Гольдфарба – Шенно (МБФГШ). Исходные данные – ЦФ и ее градиент, начальная точка x^0 , точность по аргументу ε_x , точность по функции ε_y .

1. Задать номер итерации $k = 0$, вычислить $f(x^0)$, положить $A^0 = E$.
2. Вычислить направление спуска: $d_k = -A^k \nabla f(x^k)$.
3. Решить задачу:

$$\Phi(\lambda) = f[x^k - \lambda A^k \nabla f(x^k)] \rightarrow \min_{\lambda \geq 0},$$

т. е. найти такое λ_k , что

$$\Phi(\lambda_k) = \min_{\lambda \geq 0} f(x^k + \lambda d_k).$$

4. Положить $x^{k+1} = x^k + \lambda_k d_k$.
5. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $\|\nabla f(x^{k+1})\| \leq \varepsilon_y$, то положить $x^* \approx x^{k+1}$ и закончить вычисления. Иначе перейти на шаг 6.
6. Вычислить новое приближение обращенного гессиана:

$$A^{k+1} = \Gamma^k A^k (\Gamma^k)^T + \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k},$$

где $\Gamma^k = E - \frac{\delta_k \gamma_k^T}{\delta_k^T \gamma_k}$, $\delta_k = x^{k+1} - x^k$, $\gamma_k = \nabla f(x^{k+1}) - \nabla f(x^k)$.

7. Выполнить присваивание $k = k + 1$ и вернуться на шаг 2.

Достоинства. Не всегда обязателен возврат к начальной итерации алгоритма и относительно слабая зависимость от точности вычислений одномерного поиска.

3.3.7 Обобщенный градиентный метод

Сходство методов сопряженных градиентов и квазиньютоновских методов дает основание для разработки обобщенного алгоритма (обобщенный градиентный метод, ОГМ), в основе которого лежит использование ряда рассмотренных выше методов.

Алгоритм ОГМ

1. Задать M – максимально допустимое число итераций; n – количество переменных; x^0 – начальная точка; ε_x или δ_x – требуемая погрешность по аргументу (абсолютная или относительная), ε_y – требуемая погрешность по функции; ε'_x (или δ'_x), ε'_y – точность одномерной минимизации.

2. Положить $k = 0$.

3. Вычислить компоненты $\nabla f(x^k)$.

4. Проверить выполнение критерия останова: $\|\nabla f(x^k)\| \leq \varepsilon_y$. Да: перейти на шаг 13. Нет: перейти на шаг 5.

5. Проверить выполнение критерия останова: $k > M$. Да: перейти на шаг 13. Нет: перейти на шаг 6.

6. Вычислить направление $d_k = d(x^k)$.

7. Проверить выполнение неравенства: $\nabla f^T(x^k)d(x^k) < 0$. Да: перейти на шаг 8. Нет (т. е. было выбрано неудачное направление): положить $d(x^k) = -\nabla f(x^k)$ и перейти на шаг 8.

8. Найти такое значение длины шага λ_k , при котором $\Phi(\lambda) = f(x^k + \lambda d_k) \rightarrow \min$ (если применяется численный метод одномерной оптимизации, искать минимум с точностью по аргументу ε'_x/δ'_x и по функции ε'_y).

9. Положить: $x^{k+1} = x^k + \lambda_k d_k$.

10. Проверить выполнение неравенства: $f(x^{k+1}) < f(x^k)$. Да: перейти на шаг 11. Нет: перейти на шаг 13.

11. Проверить выполнение неравенства: $\|\Delta x\| \leq \varepsilon_x$ или $\|\Delta x\|/\|x^k\| \leq \delta_x$. Да: перейти на шаг 13. Нет: перейти на шаг 12.

12. Положить $k = k + 1$, перейти на шаг 3.

13. Окончание поиска.

В данном алгоритме можно использовать различные методы для определения соответствующих направлений поиска на шаге 6. Отдельные шаги (шаг 7) не соответствуют методу Коши.

Эффективность алгоритма можно повысить за счет включения в шаг 8 дополнительных процедур, предложенных Дэвидоном (W. Davidon), Пауэллом (M. Powell) и Шенно (D. F. Shanno).



.....
 В процессе одномерного поиска следует по возможности избегать точных вычислений, т. к. на выполнение этих операций тратится значительная часть времени общего счета.

3.4 Сравнение методов

Предполагается, что выполняются следующие условия: функция $f(x)$ дважды дифференцируема; гессиан $\nabla^2 f(x^*)$ положительно определен.

Метод наискорейшего спуска (метод Коши). Последовательность $f(x^k)$ удовлетворяет неравенству:

$$\limsup_{k \rightarrow \infty} \frac{f(x^{k+1}) - f(x^*)}{f(x^k) - f(x^*)} = \alpha \leq \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \right)^2, \quad (3.74)$$

где λ_{\max} , λ_{\min} – соответственно наибольшее и наименьшее собственные значения матрицы Гессе $\nabla^2 f(x^*)$, вычисленные в точке x^* .

Таким образом, в худшем случае имеет место *линейная сходимость*, для которой коэффициент асимптотической сходимости α (называемый отношением Канторовича) непосредственно связан с $K(H)$ – обусловленностью матрицы $H_f(x^*)$.

Для плохо обусловленной функции (овражного типа) сходимость может быть очень медленной ($\alpha \approx 1$).

Ускоренные методы наискорейшего спуска. Для ускоренного метода второго порядка имеем

$$\limsup_{k \rightarrow \infty} \frac{f(x^{k+1}) - f(x^*)}{f(x^k) - f(x^*)} = \alpha \leq \left(\frac{\lambda_{m_2} - \lambda_{\min}}{\lambda_{m_2} + \lambda_{\min}} \right)^2, \quad (3.75)$$

где λ_{m_2} – второе наибольшее собственное значение $\nabla^2 f(x^*)$. В двумерном случае имеем $\lambda_{m_2} = \lambda_{\min}$, и, следовательно, ускоренный метод второго порядка имеет сходимость высшего порядка, что очень эффективно. Однако в размерностях выше 2-й сходимость методов с порядком $p = 2$ может практически быть такой же плохой, как и в обычном методе, т. к. 2-е собственное значение может мало отличаться от λ_{\max} .

При $p = n$ (размерность пространства) имеем суперлинейную сходимость. Но при этом на каждой итерации требуется применение $n + 1$ одномерных минимизаций. В таком случае метод становится эквивалентным методу сопряженного градиента.

Метод Ньютона. Обладает квадратичной локальной сходимостью. На практике глобальная сходимость не гарантируется.

Методы сопряженных градиентов (относительно метода Флетчера – Ривза). Доказано, что

$$\frac{\|x^{k+n} - x^*\|}{\|x^k - x^*\|} \rightarrow 0, \quad k \rightarrow \infty, \quad (3.76)$$

т. е. последовательность $x^0, x^n, x^{2n}, \dots, x^{gn}$ сходится к x^* суперлинейно. Если $\nabla^2 f(x)$ удовлетворяет условию Липшица

$$|\nabla^2 f(x)y| \leq \Lambda|y|, \quad \forall y$$

в окрестности точки x^* , то имеем квадратичную сходимость по n шагам:

$$\limsup_{k \rightarrow \infty} \frac{\|x^{k+n} - x^*\|}{\|x^k - x^*\|^2} < \infty. \quad (3.77)$$

Квазиньютоновские методы (применительно к алгоритму МДФП). Сходимость суперлинейная:

$$\frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} \rightarrow 0, k \rightarrow \infty. \quad (3.78)$$

Если, кроме того, $\nabla^2 f(x)$ удовлетворяет условию Липшица в окрестности точки x^* , то сходимость к x^* квадратичная:

$$\limsup_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^2} < \infty. \quad (3.79)$$

Преимущества квазиньютоновских методов по сравнению с методами сопряженных градиентов: требуют в n раз меньше этапов (шагов), т. е. в n раз меньше числа одномерных минимизаций. Сходимость для одного и того же поведения – квадратичная или суперлинейная.

Недостатки – нагрузка памяти пропорциональна n^2 , объем вычислений пропорционален n^2 .

Результаты исследования функции Розенброка:

$$f(x) = 100 \times (x_2 - x_1^2)^2 + (1 - x_1)^2, \quad (3.80)$$

$$x^0 = (-1, 2; 1, 0)^T, \quad x^* = (1, 0; 1, 0)^T -$$

приведены в таблице 3.1.

Таблица 3.1 – Сравнение методов

Метод	Метод одномерной минимизации		
	дихотомии	золотого сечения	кубической аппроксимации
Коши	Значение ЦФ $1,1 \cdot 10^{-10}$ Количество вычислений ЦФ 38424 Время вычисления 7,47 с	$1,25 \cdot 10^{-10}$ 4066 1,382	$6,19 \cdot 10^{-10}$ 10685 2,631
Флетчера – Ривза	$3,24 \cdot 10^{-6}$ 988 0,18	$5,91 \cdot 10^{-6}$ 805 0,57	$2,77 \cdot 10^{-7}$ 273 0,059
Дэвидона – Флетчера – Пауэлла	$2,45 \cdot 10^{-8}$ 977 0,173	$2,39 \cdot 10^{-8}$ 656 0,133	$4,3 \cdot 10^{-8}$ 239 0,063

Метод	Метод одномерной минимизации		
	дихотомии	золотого сечения	кубической аппроксимации
Бройдена –	$5,6 \cdot 10^{-8}$	$3,6 \cdot 10^{-8}$	$3,9 \cdot 10^{-9}$
Флетчера –	932	740	204
Гольдфарба –	0,169	0,161	0,065
Шенно			

Во всех методах использовалась численная аппроксимация градиента.



Выводы

1. Метод Коши + дихотомия – обеспечивает нахождение наиболее точного значения ЦФ, но требует больших затрат машинного времени.
2. Метод БФГШ с использованием кубической аппроксимации – самый эффективный по количеству вычислений функции.



Контрольные вопросы по главе 3

1. В чем заключается суть работы симплекс-метода?
2. Поясните принцип исследующего поиска и поиска по образцу в методе Хука – Дживса.
3. В чем заключается идея метода сопряженных направлений Пауэлла? Какие направления называются сопряженными?
4. Как определяется параметр длины рабочего шага в методе Коши?
5. Почему метод Ньютона неэффективен при оптимизации «овражных» целевых функций?
6. Чем отличается метод Ньютона от модифицированного метода Ньютона?
7. Назовите положительные свойства метода Марквардта.
8. Какие существуют разновидности метода сопряженного градиента? Что у них общего и чем они отличаются?
9. В чем сущность квазиньютоновских методов с переменной метрикой? Что такое матрица коррекции?
10. Каким образом можно сравнивать многомерные методы безусловной оптимизации?

4 Линейное программирование

В предыдущих главах мы рассматривали задачи, где не было никаких ограничений на переменные, входящие в запись ЦФ. Поэтому методы, предназначенные для решения таких задач, назывались *методами безусловной оптимизации*. В реальных же оптимизационных задачах практически всегда присутствуют ограничения на управляемые переменные, связанные с конечностью ресурсов – материально-технических, человеческих, временных и др. Такие ограничения существенно уменьшают размеры области, в которой проводится поиск экстремума.

В дальнейших главах учебника будем рассматривать *методы условной оптимизации*, предназначенные для решения задач с ограничениями.

4.1 Классификация методов



.....

Задачами линейного программирования (задачами ЛП, ЗЛП) называются задачи оптимизации, в которых ограничения, представленные в виде равенств или неравенств, и целевая функция линейны.

.....

Методы линейного программирования широко используются для решения различных военных, экономических, промышленных и организационных задач. Главными причинами столь широкого применения методов линейного программирования является доступность математического обеспечения для решения задач большой размерности и возможность анализа решений задач ЛП при использовании вариации исходных данных (анализа чувствительности).



.....

Линейное программирование представляет собой наиболее часто используемый метод оптимизации (74% от общего числа методов оптимизации). Около четверти машинного времени, затраченного за последние годы на проведение научных исследований, было отведено решению задач ЛП и их многочисленных модификаций [2, 16].

.....

Для решения ЗЛП существуют следующие методы:

1. *Графический метод*. Данным методом можно решать задачи, количество независимых переменных в которых равно 2. Определение базисных и независимых переменных будет дано ниже.
2. *Симплекс-метод*. Этим методом можно решать задачи, представленные в стандартной форме. Приведение ЗЛП к стандартной форме рассмотрим далее.
3. *Метод Гомори*. Модификация симплекс-метода, позволяющая решать ЗЛП с дополнительным условием целочисленности решения.
4. *Методы решения транспортной задачи*. Предназначены для решения ЗЛП особого вида, рассмотренных в главе 5.

4.2 Разработка моделей линейного программирования

Построение моделей линейного программирования практических задач не следует рассматривать как науку, скорее это искусство, которое постигается с опытом.

Разработка модели линейного программирования включает в себя следующие этапы [1]:

1. Определение переменных задачи.
2. Представление ограничений ЗЛП в виде линейных уравнений или неравенств.
3. Задание линейной целевой функции, подлежащей минимизации или максимизации.

Мы будем рассматривать только задачи на минимизацию. Если требуется максимизировать ЦФ $f(x)$, то нужно просто изменить ее знак на обратный $F(x) = -f(x)$ и далее минимизировать ЦФ $F(x)$.

В ЗЛП не используются рассмотренные ранее критерии останова итерационного процесса. На это есть ряд причин.

Во-первых, все методы решения ЗЛП являются точными. При программной реализации алгоритмов решения ЗЛП могут возникать погрешности, связанные с конечностью разрядной сетки ЭВМ, но сами алгоритмы являются точными. Здесь можно провести аналогию с численным решением систем уравнений – существуют точные методы решения систем линейных уравнений (СЛАУ), но все методы решения систем нелинейных уравнений (СНУ) являются приближенными.

$A = \{a_{ij}\}$ – действительная матрица ограничений размерностью $m \times n$;

a_{ij} – коэффициенты при переменных в ограничениях (4.2), (4.5);

$c = (c_1, c_2, \dots, c_n)^T$ – вектор, где c_j – коэффициенты при переменных в целевой функции (4.1), (4.4);

c_0 – свободный коэффициент ЦФ (4.1);

$b = (b_1, b_2, \dots, b_m)^T$ – вектор правой части ограничений;

$f(x) = c_0 + (c, x) = c_0 + c^T x = c_0 + \sum_{j=1}^n c_j x_j$ – целевая функция.

Целевая функция может не иметь свободного коэффициента:

$$f(x) = (c, x) = c^T x = \sum_{j=1}^n c_j x_j,$$

и на положение точки экстремума он влияния не оказывает. От него зависит только значение ЦФ в точке экстремума.

Стандартная форма (СФ) задач линейного программирования. Задача линейного программирования имеет стандартную (каноническую) форму, если [1, 5]:

- ЦФ минимизируется;
- все ограничения имеют форму равенства (кроме ограничений неотрицательности переменных $x_j \geq 0$);
- элементы вектора правой части ограничений неотрицательны:

$$\begin{aligned} f(x) &= c_0 + c^T x \rightarrow \min; \\ Ax &= b; \\ x &\geq 0, \quad b \geq 0. \end{aligned} \tag{4.7}$$

Основные методы решения задач линейного программирования ориентированы на использование стандартной формы записи. Любую задачу линейного программирования можно представить в стандартной форме:

1. Если задача на максимизацию, то умножить ЦФ на (-1) и искать минимум.
2. Если i -е ограничение имеет вид:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i,$$

то необходимо преобразовать его в равенство, добавив к левой части неотрицательную переменную $x_{n+1} \geq 0$:

$$\sum_{j=1}^n a_{ij}x_j + x_{n+1} = b_i, \quad (4.8)$$

а число переменных задачи увеличить до $n + 1$. При этом полагается, что новая переменная входит в ЦФ с коэффициентом 0.

3. Если i -е ограничение имеет вид:

$$\sum_{j=1}^n a_{ij}x_j \geq b_i,$$

то необходимо преобразовать его в равенство, отняв от левой части неотрицательную переменную $x_{n+1} \geq 0$:

$$\sum_{j=1}^n a_{ij}x_j - x_{n+1} = b_i, \quad (4.9)$$

а число переменных задачи увеличить до $n + 1$. При этом полагается, что новая переменная входит в ЦФ с коэффициентом 0.

4. Неотрицательность элементов вектора b можно получить, умножив, если это необходимо, соответствующие неравенства (4.2) или (4.5) на (-1) .

5. Если на переменную x_j не наложены ограничения на знак, т. е. x_j может принимать и положительные, и отрицательные значения, то вхождения неограниченной переменной в ЦФ и ограничения заменяют выражением:

$$x_j = x_{n+1} - x_{n+2}, \text{ где } x_{n+1} \geq 0, x_{n+2} \geq 0, \quad (4.10)$$

а число переменных задачи увеличивают до $n + 1$. Значения x_j могут быть любого знака, в зависимости от соотношения величин x_{n+1} и x_{n+2} . Исходная переменная x_j из задачи исключается. После получения оптимального решения ее значение может быть получено по формуле (4.10):

$$x_j^* = x_{n+1}^* - x_{n+2}^*.$$

6. Если на переменную x_j наложено ограничение на знак вида $x_j \leq 0$, то ее вхождения в ЦФ и ограничения заменяют выражением

$$x_j = -x_{n+1}, \text{ где } x_{n+1} \geq 0, \quad (4.11)$$

а число переменных задачи не изменится. Исходная переменная x_j из задачи исключается. После получения оптимального решения ее значение может быть получено по формуле (4.11):

$$x_j^* = -x_{n+1}^*.$$

стемы, имеющей m уравнений и n неизвестных, к каноническому виду при помощи элементарных операций над строками. При использовании первых m переменных (x_1, x_2, \dots, x_m) такая система примет следующий вид:

$$\begin{aligned} x_1 + \bar{a}_{1m+1}x_{m+1} + \dots + \bar{a}_{1s}x_s + \dots + \bar{a}_{1n}x_n &= \bar{b}_1, \\ &\vdots \\ x_k + \bar{a}_{km+1}x_{m+1} + \dots + \bar{a}_{ks}x_s + \dots + \bar{a}_{kn}x_n &= \bar{b}_k, \\ &\vdots \\ x_m + \bar{a}_{mm+1}x_{m+1} + \dots + \bar{a}_{ms}x_s + \dots + \bar{a}_{mn}x_n &= \bar{b}_m. \end{aligned} \quad (4.14)$$

Или (4.14) можно переписать следующим образом:

$$x_i = \bar{b}_i - \sum_{j=m+1}^n \bar{a}_{ij}x_j, \quad i=1, 2, \dots, m. \quad (4.15)$$



.....

*Переменные x_1, \dots, x_m , входящие с единичными коэффициентами только в одно уравнение системы (4.14) или (4.15) и с нулевыми – в остальные, называются **базисными** или **зависимыми**. В канонической системе каждому уравнению соответствует ровно одна базисная переменная.*

*Остальные $n - m$ переменных (x_{m+1}, \dots, x_n) называются **небазисными, свободными** или **независимыми переменными**. В общем случае любые m переменных задачи можно выбрать в качестве базисных, только остальные $n - m$ переменных останутся свободными.*

.....

При записи системы в каноническом виде все ее решения можно получить присваивая независимым переменным произвольные значения и решая затем получающуюся систему относительно зависимых переменных. Для приведения системы к каноническому виду можно использовать 2 типа элементарных операций над строками:

- 1) умножение любого ограничения системы на положительное или отрицательное число;
- 2) прибавление к любому ограничению другого ограничения системы, умноженного на положительное или отрицательное число.



.....
Базисным решением системы (4.14) называется решение, полученное при нулевых значениях небазисных переменных.

Например, в системе (4.14) базисное решение задается как

$$x_1 = \bar{b}_1, x_2 = \bar{b}_2, \dots, x_m = \bar{b}_m, x_{m+1} = \dots = x_n = 0. \quad (4.16)$$



.....
*Базисное решение называется **допустимым базисным решением**, если значения входящих в него базисных переменных $x_j \geq 0$, $j = 1, 2, \dots, m$, что эквивалентно условию неотрицательности $\bar{b}_j \geq 0$, $j = 1, 2, \dots, m$. Если базисное решение является недопустимым (т. е. некоторые коэффициенты $\bar{b}_j < 0$), то следует выделить базис по другим переменным.*

*Допустимое базисное решение является **угловой точкой** допустимого множества S задачи линейного программирования и называется иногда **опорным планом**.*

*Если среди неотрицательных чисел b_j в (4.14) есть числа, равные нулю, то допустимое базисное решение называется **вырожденным** (вырожденной угловой точкой) и соответствующая задача линейного программирования называется **вырожденной**.*

Предположим, что ЗЛП (4.12) является невырожденной, а базисное решение (4.16) допустимо. Обозначим его:

$$x^0 = (\bar{b}_1, \bar{b}_2, \dots, \bar{b}_m, 0, 0, \dots, 0)^T. \quad (4.17)$$

Используя соотношение (4.15), выразим целевую функцию из (4.12) через свободные переменные x_j , $j = m + 1, \dots, n$:

$$f(x) = p_0 + \sum_{j=m+1}^n p_j x_j, \quad (4.18)$$

где $p_0 = \sum_{i=1}^m c_i \bar{b}_i$, $p_j = c_j - \sum_{i=1}^m c_i \bar{a}_{ij}$, $j = m + 1, \dots, n$.

В ЦФ ненулевые коэффициенты останутся только при независимых переменных, а при всех базисных переменных будут равны 0.

4.5 Поиск начального базиса

Алгоритм выделения базиса и получения опорного плана ЗЛП [9]:

1. Привести задачу к СФ.
2. Если после этого в системе ограничений уже есть m переменных, которые можно принять за базисные, перейти к шагу 4. Иначе – шаг 3.
3. Если базис не выделен, возможны два способа его выделения:
 - 3.1. Выделить базис по существующим переменным, используя указанные выше элементарные операции над строками матрицы ограничений и вектора правых частей ограничений. Перейти на шаг 4.
 - 3.2. Получить базисное решение, используя метод искусственного базиса и закончить работу алгоритма.
4. Получить начальное базисное решение.
5. Избавиться от базисных переменных в записи ЦФ.

В общем случае ЗЛП может иметь несколько допустимых начальных базисных решений, и нет разницы, с каким из них начинать решение задачи.

4.5.1 Метод Жордана – Гаусса

Метод Жордана – Гаусса – это метод симплексного преобразования системы ограничений-равенств. Этот метод является модификацией метода Гаусса. В его основе лежит использование элементарных преобразований системы ограничений, как в классическом методе Гаусса, используемом в численных методах. Цель преобразований – привести m произвольных столбцов матрицы коэффициентов ограничений к единичным столбцам, т. е. чтобы вместе эти столбцы образовывали единичную матрицу размером $m \times m$. Здесь m – количество ограничений задачи.

4.5.2 Метод искусственного базиса

Метод искусственного базиса применяется тогда, когда система ограничений задана в виде, в котором трудно выделить базисные переменные. В этом случае в задачу вводятся m дополнительных (искусственных) переменных $x_{n+i} \geq 0$, $i = 1, 2, \dots, m$, которые добавляются к соответствующему i -му ограничению:

$$x_{n+i} + \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m. \quad (4.19)$$

Так как задача предварительно преобразована к СФ, т. е. все ограничения уже имели форму равенства, то решение полученной задачи будет эквивалентно решению исходной задачи только тогда, когда в оптимальном решении все $x_{n+i}^* = 0$. Чтобы это гарантировать, составляется вспомогательная ЦФ:

$$F(x) = \sum_{i=1}^m x_{n+i}.$$

Так как $x_{n+i} = b_i - \sum_{j=1}^n a_{ij}x_j$, то

$$F(x) = \sum_{i=1}^m x_{n+i} = \sum_{i=1}^m \left(b_i - \sum_{j=1}^n a_{ij}x_j \right) = p_0 + \sum_{j=1}^n p_j x_j, \quad (4.20)$$

где

$$p_0 = \sum_{i=1}^m b_i - \text{свободный коэффициент ЦФ};$$

$$p_j = -\sum_{i=1}^m a_{ij}, \quad j = 1, \dots, n - \text{коэффициент ЦФ при переменной } x_j.$$

Таким образом, получили задачу

$$\begin{aligned} F(x) &= p_0 + \sum_{j=1}^n p_j x_j \rightarrow \min, \\ x_{n+i} + \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = 1, 2, \dots, m, \\ x_k &\geq 0, \quad k = 1, \dots, n + m. \end{aligned} \quad (4.21)$$

Начальным базисным решением полученной задачи будет точка $x^0 = (0, 0, \dots, 0, b_1, b_2, \dots, b_m)^T$. Решая ее одним из предложенных далее методов, получим базисное решение исходной задачи.

4.6 Графическое решение ЗЛП

Графическим методом можно решить ЗЛП, имеющие не более двух независимых переменных, т. е. при условии $n - m \leq 2$, т. к. если $n - m \geq 3$, то задачу придется решать графически в пространстве R^3 и более высоких размерностей, что затруднительно. Однако, если $n = m$, т. е. в задаче нет независимых переменных ($n - m = 0$), в применении графического метода смысла нет. В этом случае все ограничения задачи пересекаются в одной точке. И если эта точка удовлетворяет ограничениям на знаки переменных задачи, то она и будет опти-

мальным решением. Если $n = m + 1$, т. е. независимая переменная всего одна ($n - m = 1$), то в применении графического метода также нет смысла. В этом случае решение задачи сводится к решению одного линейного неравенства. Поэтому реально имеет смысл решение ЗЛП графическим методом лишь в пространстве R^2 (на плоскости) при $n = m + 2$, т. е. если независимых переменных две ($n - m = 2$). На практике такие задачи обычно не встречаются, однако графический метод хорошо иллюстрирует некоторые основные понятия, используемые при решении задач линейного программирования большой размерности [1].

Алгоритм графического метода

Пусть имеем ЗЛП

$$f(x) = c_0 + \sum_{j=1}^n c_j x_j \rightarrow \text{extr}$$

при ограничениях

$$\sum_{j=1}^n a_{ij} x_j \{ \leq, =, \geq \} b_i, \quad i = 1, 2, \dots, m,$$

т. е. для решения ЗЛП данным методом задачу не обязательно приводить к СФ. Тогда:

1. Если в задаче всего две переменные, то полагаем, что индексы $l_1 = 1$, $l_2 = 2$ (это индексы переменных, которые будут отложены по координатным осям графика), и переходим на шаг 3, иначе на шаг 2.

2. Приведем задачу к виду с двумя переменными. Для этого выделяем в ограничениях базис:

$$x_{k_i} + \sum_{\substack{j=1 \\ j \neq k_1 \dots k_m}}^n \bar{a}_{ij} x_j = \bar{b}_i, \quad x_{k_i} \geq 0, \quad i = 1, 2, \dots, m, \quad (4.22)$$

Здесь k_i – индексы базисных переменных. Например, это можно сделать описанными выше способами выделения базиса. Перепишем (4.22) в виде

$$x_{k_i} = \bar{b}_i - \sum_{\substack{j=1 \\ j \neq k_1 \dots k_m}}^n \bar{a}_{ij} x_j, \quad i = 1, 2, \dots, m. \quad (4.23)$$

С учетом условия, что количество независимых переменных задачи равно двум, в правой части выражений (4.23) останутся только две неизвестные, т. е.

$$x_{k_i} = \bar{b}_i - \bar{a}_{il_1} x_{l_1} - \bar{a}_{il_2} x_{l_2}, \quad i = 1, 2, \dots, m. \quad (4.24)$$

Здесь l_1 и l_2 – индексы независимых переменных ($l_1 \neq l_2$, $l_1 \neq k_i$, $l_2 \neq k_i$). Тогда, с учетом ограничений на знак базисных переменных $x_{k_i} \geq 0$, получим

$$\begin{aligned} \bar{b}_i - \bar{a}_{il_1} x_{l_1} - \bar{a}_{il_2} x_{l_2} &\geq 0 \Rightarrow \\ \Rightarrow \bar{a}_{il_1} x_{l_1} + \bar{a}_{il_2} x_{l_2} &\leq \bar{b}_i, \quad i = 1, 2, \dots, m. \end{aligned} \quad (4.25)$$

Также избавляемся от базисных переменных в ЦФ. Таким образом, в задаче останутся две переменные – x_{l_1} и x_{l_2} .

3. Строим координатную сетку в декартовой системе координат пространства R^2 . По осям сетки откладываем переменные x_{l_1} и x_{l_2} .

4. Строим ОДР. Для этого изображаем на графике область действия каждого ограничения задачи. Если ограничение является равенством вида $a_{il_1} x_{l_1} + a_{il_2} x_{l_2} = b_i$, то областью действия ограничения будет только сама прямая $a_{il_1} x_{l_1} + a_{il_2} x_{l_2} = b_i$. Если ограничение является неравенством вида $a_{il_1} x_{l_1} + a_{il_2} x_{l_2} \{ \leq, \geq \} b_i$, то областью действия неравенства будет полуплоскость по одну из сторон от прямой $a_{il_1} x_{l_1} + a_{il_2} x_{l_2} = b_i$. Чтобы определить, какая это полуплоскость, можно проверить, удовлетворяет ли данному ограничению точка начала координат $O(0,0)$ или какая-то другая точка. Областью допустимых решений S задачи будет выпуклый многогранник (треугольник, четырехугольник и т. д., в вырожденных случаях – точка или прямая), являющийся пересечением областей действия всех ограничений.



.....

Возможен частный случай, когда система ограничений несовместима. В этом случае ЗЛП не имеет решения, т. к. ОДР $S = \emptyset$ (рис. 4.1), и работа алгоритма заканчивается. Если бы на рисунке 4.1 все штриховки в четырехугольнике были направлены вовнутрь, то ОДР была бы непустой и задача ЛП была бы непротиворечива.

.....

5. Строим линию уровня ЦФ $f(x) = \text{const}$ таким образом, чтобы она пересекала ОДР. Так как ЦФ линейная, то соответствующие ей точки будут лежать на некоторой прямой. При изменении величины константы прямая подвергается параллельному переносу. Смещаем эту линию в направлении градиента (при поиске максимума) или антиградиента (при поиске минимума)

до точки последнего касания границы ОДР. Это и будет точка оптимального решения задачи. А значение константы, которое ей соответствует, будет оптимальным значением ЦФ.

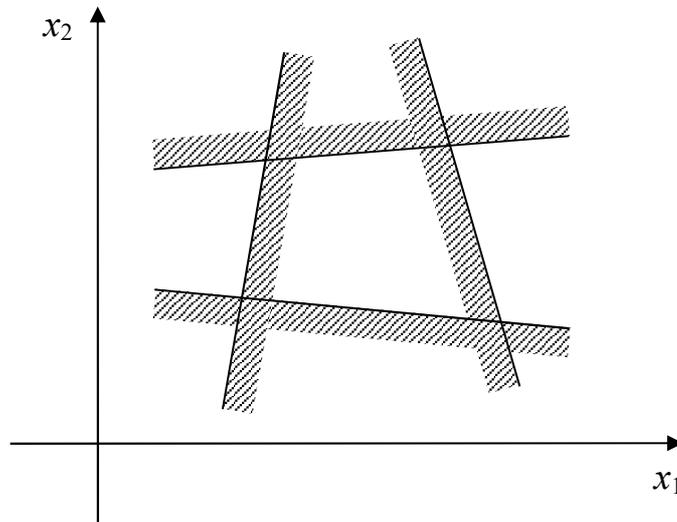
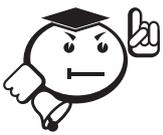


Рис. 4.1 – ОДР ЗЛП – пустое множество



.....
 Возможны частные случаи, когда ОДР не пуста, а ЗЛП либо имеет множество решений, либо вообще не имеет решений.

Рассмотрим эти случаи [6]:

1. При параллельном переносе прямая $f(x) = \text{const}$ совпадает с одной из сторон многогранника решений. Имеем бесконечное множество решений (рис. 4.2). В данном примере каждая точка отрезка AE является оптимальным решением.

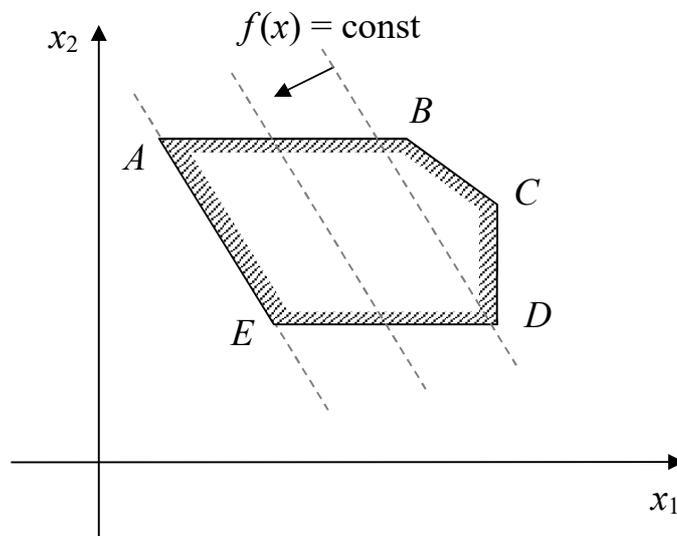


Рис. 4.2 – ЗЛП, имеющая бесконечное множество решений

2. ОДР не ограничена сверху («стакан», рис. 4.3).

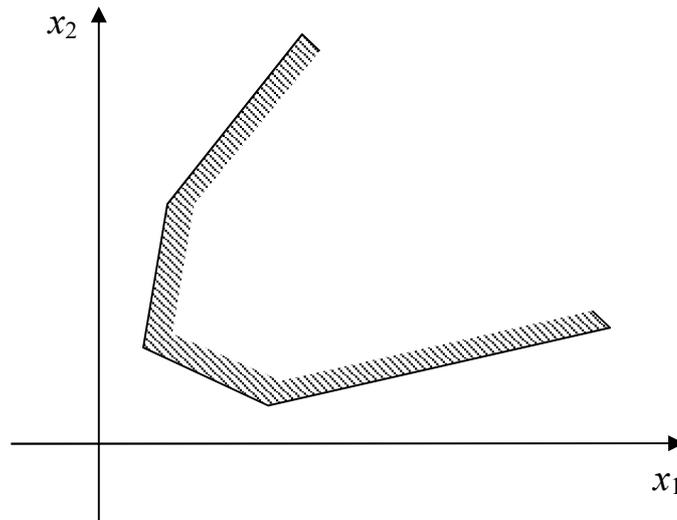


Рис. 4.3 – ОДР ЗЛП не ограничена сверху

В этом случае, если направление антиградиента направлено ко «дну стакана» (вниз), то задача будет иметь решение только на минимум, в противном случае – только на максимум.

3. ОДР не ограничена снизу («перевернутый стакан», рис. 4.4).

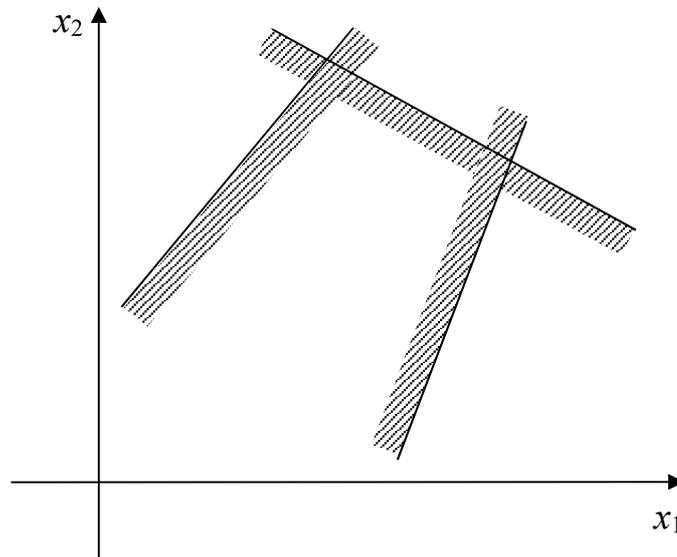


Рис. 4.4 – ОДР ЗЛП не ограничена снизу

Аналогично, если направление антиградиента направлено ко «дну стакана» (вверх), то задача будет иметь решение только на минимум, в противном случае – только на максимум.

4. Если в процессе решения выделялись базисные переменные, то найти их значения можно из (4.24).

4.7 Основы симплекс-метода

В основе симплекс-метода (СМ) лежит следующая теорема [1].



.....
Теорема. Если ЗЛП разрешима, то экстремум целевой функции $f(x)$ достигается хотя бы в одной из угловых точек допустимого множества S этой задачи.

Так как различные базисные решения системы (4.14) соответствуют различным вариантам выбора m из общего числа n переменных x_j , то число допустимых базисных решений (угловых точек) не превышает

$$C_n^m = \frac{n!}{m!(n-m)!}.$$

Поэтому ЗЛП можно решать посредством перебора конечного числа угловых точек допустимого множества S , сравнивая значения целевой функции в этих точках – наилучший вариант.



..... **Пример**

Если $n = 50$, $m = 25$ (задача небольшой размерности), то ЭВМ, выполняя 10^5 переборов вариантов в секунду, будет искать решение 40 лет.

Идея СМ состоит в направленном переборе угловых точек допустимого множества S с последовательным уменьшением целевой функции $f(x)$.

Симплекс-метод разработан Дж. Данцигом. Наиболее простой метод, используемый для решения ЗЛП, – это *симплексный метод Данцига*.

Здесь начало координат переносится в угловую точку допустимой области и рассматриваются линии, соединяющие эту угловую точку со смежными угловыми точками. Исследуется вариация целевой функции $f(x)$ вдоль этих линий, и если какая-нибудь из них дает улучшение, начало координат переносится в угловую точку, в которую ведется линия. Процесс затем повторяется для новой угловой точки и прекращается, когда достигается угловая точка, где $f(x)$ не увеличивается (не уменьшается) вдоль любой такой линии [3, 5].

Алгоритм симплекс-метода

1. Выделить в задаче допустимый базис, т. е. необходимо преобразовать задачу к виду (4.14)–(4.18).

2. Положить $k = 0$, $a_{ij}^0 = \bar{a}_{ij}$, $b_i^0 = \bar{b}_i$, $p_0^0 = p_0$, $p_j^0 = p_j$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$.

3. На данном шаге решения для упрощения будем полагать, что базисными являются переменные x_1, \dots, x_m , а независимыми – переменные x_{m+1}, \dots, x_n . Таким образом, имеем ЗЛП:

$$f(x) = p_0^k + \sum_{j=m+1}^n p_j^k x_j, \quad (4.26)$$

$$x_i = b_i^k - \sum_{j=m+1}^n a_{ij}^k x_j, \quad i = 1, 2, \dots, m, \quad (4.27)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n.$$

Текущая угловая точка:

$$x^k = (b_1^k, b_2^k, \dots, b_m^k, 0, 0, \dots, 0)^T. \quad (4.28)$$

Тогда:

- а) если в (4.26) все коэффициенты $p_j^k \geq 0$, $j = m+1, \dots, n$, то в угловой точке (4.28) достигается минимум целевой функции $f(x)$ на допустимом множестве S ЗЛП и этот минимум равен p_0^k ;
- б) если среди отрицательных коэффициентов p_j^k , $j = m+1, \dots, n$ есть такой (например, p_i^k), что в (4.27) все коэффициенты $a_{il}^k \leq 0$, $l = 1, 2, \dots, m$, то целевая функция $f(x)$ не ограничена снизу на S и задача не имеет решений;
- в) если хотя бы один из коэффициентов p_j^k , $j = m+1, \dots, n$ отрицателен (например, $p_i^k < 0$) и при этом среди коэффициентов a_{il}^k в (4.27) есть хотя бы один положительный, то существует угловая точка $x^{k+1} \in S$ такая, что $f(x^{k+1}) < f(x^k)$.

В случаях (а) и (б) процесс решения ЗЛП заканчивается. В случае (в) переходим на шаг 4.

При использовании симплекс-метода задачу удобнее записывать в виде *симплекс-таблицы* (СТ). Для симплекс-таблиц также указывают номер итерации симплекс-метода k , на которой они были получены – СТ-0, СТ-1, СТ-2 и т. д. Например, задачу (4.26)–(4.27) можно представить следующей симплекс-таблицей:

СТ- k	x_{m+1}	x_{m+2}	\dots	x_n	b
x_1	$a_{1,m+1}^k$	$a_{1,m+2}^k$	\dots	$a_{1,n}^k$	b_1^k
x_2	$a_{2,m+1}^k$	$a_{2,m+2}^k$	\dots	$a_{2,n}^k$	b_2^k
\dots	\dots	\dots	\dots	\dots	\dots
x_m	$a_{m,m+1}^k$	$a_{m,m+2}^k$	\dots	$a_{m,n}^k$	b_m^k
p	p_{m+1}^k	p_{m+2}^k	\dots	p_n^k	$-p_0^k$

В заголовки строк записываем базисные переменные, в заголовки столбцов – независимые. В ячейках на пересечении строк базисных и столбцов независимых переменных записываем соответствующие коэффициенты a_{ij}^k системы ограничений (4.27). В последний столбец записываем коэффициенты вектора правых частей ограничений b_i^k . Эти же коэффициенты являются значениями базисных переменных в текущей угловой точке x^k . Значения независимых переменных равны 0. В последнюю строку таблицы записываются коэффициенты ЦФ (4.26), причем коэффициент p_0^k пишется последним и с обратным знаком.

4. Пусть в (4.26) или последней строке СТ- k коэффициент $p_l^k < 0$, $l \neq 0$ и в (4.27) или СТ- k имеются положительные коэффициенты a_{il}^k . Найдем индекс строки t из условия (правило минимального частного):

$$\frac{b_t^k}{a_{il}^k} = \min_i \left\{ \frac{b_i^k}{a_{il}^k} \right\} \forall a_{il}^k > 0, \quad (4.29)$$

т. е. минимум берется по всем номерам $i = 1, 2, \dots, m$, для которых $a_{il}^k > 0$. Тогда элемент a_{il}^k будем называть *разрешающим* или *опорным*.

Найдем решение системы ограничений из (4.13), т. е.

$$\sum_{j=1}^n a_{ij} x_j = b_i; \quad i = 1, 2, \dots, m,$$

считая свободными переменные $x_{m+1}, \dots, x_{l-1}, x_t, x_{l+1}, \dots, x_n$, т. е. поменяв местами свободную переменную x_l с базисной переменной x_t . Система уравнений (4.14) в этом случае запишется следующим образом:

$$x_i + \sum_{j=m+1}^n \left(a_{ij}^k - a_{il}^k \frac{a_{ij}^k}{a_{il}^k} \right) x_j - \frac{a_{il}^k}{a_{il}^k} x_t = b_i^k - a_{il}^k \frac{b_t^k}{a_{il}^k},$$

$$j \neq l, i = 1, 2, \dots, m, i \neq t, \quad (4.30)$$

$$x_l + \sum_{j=m+1}^n \frac{a_{lj}^k}{a_{il}^k} x_j + \frac{1}{a_{il}^k} x_t = \frac{b_l^k}{a_{il}^k},$$

а зависимость целевой функции от новых свободных переменных примет вид:

$$f(x) = \sum_{j=m+1}^n \left(p_j^k - p_l^k \frac{a_{lj}^k}{a_{il}^k} \right) x_j - \frac{p_l^k}{a_{il}^k} x_t + p_0^k + p_l^k \frac{b_t^k}{a_{il}^k}. \quad (4.31)$$

Компоненты нового базисного решения x^{k+1} можно найти, приравняв нулю свободные переменные x_j , $j = m+1, \dots, n$, $j \neq l$, и x_t , и найдя при этом условия значения базисных переменных из (4.30). Базисное решение x^{k+1} является допустимым, то есть угловой точкой множества ОДР S , причем гарантируется выполнение неравенства $f(x^{k+1}) < f(x^k)$.

Преобразования (4.30)–(4.31) проще выполнять над симплекс-таблицей СТ- t , а не системой ограничений и ЦФ напрямую. Рассмотрим вид симплекс-таблицы в общем случае, когда индексы базисных и свободных переменных могут быть любыми (для удобства строки и столбцы матрицы коэффициентов пронумерованы):

		1	...	h	...	$n-m$	
	СТ- t	x_q	...	x_l	...	x_r	b
1	x_s	a_{1q}^k	...	a_{1l}^k	...	a_{1r}^k	b_1^k
...
z	x_t	a_{zq}^k	...	a_{zl}^k	...	a_{zr}^k	b_z^k
...
m	x_u	a_{mq}^k	...	a_{ml}^k	...	a_{mr}^k	b_m^k
	p	p_q^k	...	p_l^k	...	p_r^k	$-p_0^k$

Опорный элемент a_{kl}^t находится на пересечении столбца таблицы с номером h , в котором записана переменная x_l , и строки с номером z , в которой записана переменная x_t . Индекс h в дальнейшем решении не используется, т. к. не важно, в каком столбце находится выбранная независимая переменная x_l .

Также обратите внимание, что номер столбца элемента a_{ij}^k означает номер независимой переменной, которой соответствует этот элемент, а номер строки – это номер ограничения.

Тогда для получения новой симплекс-таблицы СТ- $t+1$ необходимо выполнить следующие операции:

4.1. Поменять местами переменные x_t и x_l , остальные переменные оставить на прежних местах. Переменная x_t становится независимой, а переменная x_l – базисной.

4.2. На место опорного элемента (в столбец с индексом h) поставить число

$$a_{zt}^{k+1} = \frac{1}{a_{zl}^k}. \quad (4.32)$$

4.3. На остальных местах разрешающей (опорной) строки записать соответствующие элементы исходной таблицы, деленные на опорный элемент:

$$a_{zj}^{k+1} = \frac{a_{zj}^k}{a_{zl}^k}, \quad b_z^{k+1} = \frac{b_z^k}{a_{zl}^k}, \quad (4.33)$$

где j – индексы всех независимых переменных, кроме t .

4.4. На свободные места разрешающего столбца поставить со знаком минус соответствующие элементы исходной таблицы, деленные на опорный элемент:

$$a_{it}^{k+1} = -\frac{a_{il}^k}{a_{zl}^k}, \quad p_t^{k+1} = -\frac{p_l^k}{a_{zl}^k}, \quad (4.34)$$

где $i = 1, \dots, z-1, z+1, \dots, m$.

4.5. Оставшиеся свободные места в новой СТ- $t+1$ заполнить построчно следующим образом. Из строки элементов исходной таблицы вычесть произведение ее элемента из разрешающего столбца на уже заполненную разрешающую строку новой таблицы:

$$\begin{aligned} a_{ij}^{k+1} &= a_{ij}^k - a_{il}^k a_{zj}^{k+1} = a_{ij}^k - a_{il}^k \frac{a_{zj}^k}{a_{zl}^k}, & b_i^{k+1} &= b_i^k - a_{il}^k b_z^{k+1} = b_i^k - a_{il}^k \frac{b_z^k}{a_{zl}^k}, \\ p_j^{k+1} &= p_j^k - p_l^k a_{zj}^{k+1} = p_j^k - p_l^k \frac{a_{zj}^k}{a_{zl}^k}, & p_0^{k+1} &= p_0^k + p_l^k b_z^{k+1} = p_0^k + p_l^k \frac{b_z^k}{a_{zl}^k}, \end{aligned} \quad (4.35)$$

где $i = 1, \dots, z-1, z+1, \dots, m$, j – индексы всех независимых переменных, кроме t .

В итоге получим таблицу

		1	...	h	...	$n - m$	
	СТ- $t + 1$	x_q	...	x_t	...	x_r	b
1	x_s	a_{1q}^{k+1}	...	a_{1t}^{k+1}	...	a_{1r}^{k+1}	b_1^{k+1}
...
z	x_l	a_{zq}^{k+1}	...	a_{zt}^{k+1}	...	a_{zr}^{k+1}	b_z^{k+1}
...
m	x_u	a_{mq}^{k+1}	...	a_{mt}^{k+1}	...	a_{mr}^{k+1}	b_m^{k+1}
	p	p_q^{k+1}	...	p_t^{k+1}	...	p_r^{k+1}	$-p_0^{k+1}$

5. Положить $k = k + 1$ и вернуться на шаг 3.



.....

Если симплекс-метод используется в рамках метода искусственного базиса, то столбцы, соответствующие искусственным переменным, из полученной симплекс-таблицы можно вычеркивать. В данном примере, если переменная x_l была искусственной, столбец h можно вычеркнуть. И решение можно заканчивать, как только в базисе не осталось ни одной искусственной переменной.

.....

Так как число угловых точек допустимого множества S не превышает C_n^m , то случай (в) может повторяться конечное число раз, т. е. в результате конечного числа шагов перехода к новой угловой точке будет найдено решение ЗЛП либо будет сделано заключение о том, что она не имеет решения.

4.8 Целочисленное программирование



.....

Целочисленным (иногда его называют также **дискретным**) программированием называется раздел математического программирования, изучающий экстремальные задачи, в которых на искомые переменные накладывается условие целочисленности.

.....

Если в формулировку ЗЛП добавить требование целочисленности решения:

$$x_j \in Z, \quad j = 1, 2, \dots, n,$$

где Z – множество целых чисел, или просто

$$x \in Z,$$

то получим задачу целочисленного программирования (ЗЦП). Таким образом, в канонической форме ЗЦП имеет вид [1–2]:

$$f(x) = c_0 + c^T x \rightarrow \min, \quad (4.36)$$

$$Ax = b, \quad b \geq 0, \quad (4.37)$$

$$x \geq 0, \quad x \in Z. \quad (4.38)$$

Изучение этого раздела обусловлено тем, что огромное количество экономических задач носит дискретный характер. Это связано, как правило, с физической неделимостью многих элементов расчета: например, нельзя построить два с половиной завода, купить полтора автомобиля и т. д. В ряде случаев такие задачи решаются обычными методами, например, симплексным методом, с последующим округлением до целых чисел. Однако такой подход оправдан, когда отдельная единица составляет очень малую часть всего объема (например, товарных запасов); в противном случае он может внести значительные искажения в действительно оптимальное решение.

4.8.1 Графический метод решения ЗЦП

Данный метод практически во всем аналогичен графическому методу решения ЗЛП. На графике строится область допустимых решений (ОДР) S , на ней выделяется область $S_Z \subseteq S$, границы которой имеют целочисленные координаты. Отличие в том, что выполнять параллельный перенос линии уровня ЦФ (в направлении градиента при поиске максимума и антиградиента при поиске минимума) нужно не до крайней точки ОДР, а до крайней точки множества S_Z [6]. Или можно найти крайнюю точку ОДР, а затем, если ее координаты не являются целочисленными, двигать линию уровня в обратную сторону (в направлении антиградиента при поиске максимума и градиента при поиске минимума), пока она не пересечет первую точку множества S_Z . Множество S_Z , в общем случае, выпуклым не является.



Решение ЗЦП может не существовать, даже если ОДР задачи не пуста, но при этом не содержит ни одной точки с целочисленными координатами.

4.8.2 Метод Гомори



Для решения целочисленных задач разработан специальный **метод отсекающих плоскостей** (его называют также **метод Гомори**). Метод отсекающих плоскостей состоит в построении дополнительных ограничений и применении модифицированного симплексного метода.

Согласно методу Гомори [6], первый этап решения целочисленных задач не отличается от обычного расчета по симплексному алгоритму. Если среди значений переменных в оптимальном плане есть дробные, то составляется дополнительное ограничение, отсекающее дробную часть решения, но оставляющее в силе все прочие условия, которым должен удовлетворять оптимальный план. Это дополнительное ограничение присоединяется к исходным ограничениям задачи, и вновь применяется процедура симплексного метода. Алгоритм Гомори позволяет прийти к оптимальному целочисленному решению за конечное число шагов.

Введем следующее обозначение: под записью $[x]$ будем подразумевать целую часть, а под записью $\{x\}$ – дробную часть числа x . Если x – целое число, т. е. $x \in Z$, то $[x] = x$, $\{x\} = 0$. Число всегда равно сумме его целой и дробной части, т. е. $x = [x] + \{x\}$. Целая часть числа получается его округлением в меньшую сторону ($[x] \leq x$).



Пример

Например, для $x = 1,2$ имеем $[x] = 1$, $\{x\} = 0,2$. Проверка: $1 + 0,2 = 1,2$; $1 \leq 1,2$. Другой пример: для $x = -1,2$ получим $[x] = -2$, $\{x\} = 0,8$. Проверка:

$-2 + 0,8 = -1,2; -2 \leq -1,2$. Таким образом, дробная часть числа всегда будет положительной.

.....
Алгоритм метода Гомори

1. С помощью симплекс-метода находится оптимальное решение x^* ЗЛП без учета требования целочисленности (4.38). Если для x^* условие (4.38) выполняется, то задача решена. В противном случае среди чисел b_i^k последнего столбца СТ- k , определяющей решение x^* , есть такие, что $\{b_i^k\} > 0$.

2. Среди нецелых элементов b_i^k выбирается произвольный элемент b_z^k (например, с максимальной дробной частью $\{b_z^k\}$). По z -й строке СТ составляется дополнительное ограничение вида:

$$-\sum_j \{a_{zj}^k\} x_j \leq -\{b_z^k\},$$

где j – индексы всех свободных переменных.

С помощью вспомогательной переменной $x_{n+1} \geq 0$ это ограничение представляется в виде равенства:

$$x_{n+1} - \sum_j \{a_{zj}^k\} x_j = -\{b_z^k\} \quad (4.39)$$

и вводится в СТ дополнительной строкой с номером $m+1$ (тем самым количество ограничений задачи увеличивается до $m+1$, а количество переменных – до $n+1$), причем

$$a_{m+1,j}^k = -\{a_{zj}^k\}, \quad b_{m+1}^k = -\{b_z^k\}.$$

Так как $b_{m+1}^k < 0$, то после дополнения строкой (4.39) СТ перестает соответствовать допустимому базисному решению ЗЛП, которую она описывает.

3. Для перехода к допустимому базисному решению производятся следующие операции:

а) строка с отрицательным свободным членом b_{m+1}^k считается опорной;

б) если все коэффициенты $a_{m+1,j}^k \geq 0$, то задача не имеет решения, в противном случае номер l разрешающего столбца находится из условия:

$$\frac{b_{m+1}^k}{a_{m+1,l}^k} = \min_j \left\{ \frac{b_{m+1}^k}{|a_{m+1,j}^k|} \right\} \forall a_{m+1,j}^k < 0; \quad (4.40)$$

в) совершается преобразование СТ с опорным элементом $a_{m+1,l}^k$. Если в новой СТ по-прежнему есть хотя бы один отрицательный свободный член, то описанная процедура повторяется, начиная с операции (а), необходимое число раз.

Если в новой СТ все элементы $b_i^{k+1} \geq 0$, то допустимое базисное решение найдено. Отметим, что выбор опорного элемента $a_{m+1,l}^k$ гарантирует неотрицательность коэффициентов b_i^{k+1} новой СТ. Поэтому найденное допустимое решение является и оптимальным.

4. Если найденное на шаге 3 решение ЗЛП не удовлетворяет условию целочисленности, то вернуться на шаг 2.

5. Если критерий оптимальности решения ЗЛП выполняется (см. алгоритм симплекс-метода, пункты 3а, 3б), то закончить решение, иначе переход на шаг 1.

Описанный алгоритм позволяет найти решение полностью целочисленной ЗЛП или установить отсутствие решения за конечное число итераций.



Контрольные вопросы по главе 4

1. Какие задачи оптимизации относятся к классу ЗЛП? Назовите этапы разработки модели линейного программирования.
2. Приведите формы записи ЗЛП. Каким образом ЗЛП можно представить в стандартной форме?
3. Что является допустимым решением (планом) ЗЛП? Какое решение (план) ЗЛП называется оптимальным?
4. Что является допустимой областью решений ЗЛП? Где расположено оптимальное решение на допустимой области? Приведите примеры противоречивой ЗЛП, ЗЛП с множеством оптимальных решений, а также неограниченного оптимума ЗЛП.
5. Сформулируйте алгоритм поиска начального базиса в задаче линейного программирования.
6. В чем состоит суть графического метода решения ЗЛП?
7. В чем заключается суть алгоритма симплекс-метода?
8. Дайте понятие целочисленного программирования.

9. Опишите графический метод решения задач целочисленного программирования.
10. Опишите суть метода Гомори.

5 Транспортная задача

В данной главе рассмотрим ряд специальных оптимизационных задач, сводящихся к ЗЛП. Одной из таких задач является *транспортная задача*, с помощью которой можно получить ответ на вопросы следующего типа:

- Как составить план перевозок однородного груза из нескольких исходных пунктов в пункты назначения таким образом, чтобы общая стоимость перевозок была минимальной?

Другой является *задача о назначениях*, с помощью которой можно получить ответ на вопросы типа:

- Как назначить (отсюда и название задачи) людей на различные должности?
- Как распределить рабочих по станкам, чтобы общая выработка была наибольшей?
- Как распределить рабочих по станкам, чтобы затраты на заработную плату были наименьшими?
- Как наилучшим образом распределить экипажи самолетов?

5.1 Классификация методов

Для решения ЗЛП можно использовать следующие методы [3, 12]:

1. *Симплекс-метод*. Размерность задач при этом получается достаточно высокой – в них по 16 неизвестных. Таким образом, решая задачи симплекс-методом, придется оперировать симплекс-таблицами размером 8×8 . А для задач, подобных задаче о доставке хлеба, нужно учесть дополнительное ограничение $x_{ij} \leq 1$ и потом еще, возможно, применить метод Гомори, чтобы решение стало целочисленным, что еще больше увеличивает размерность симплекс-таблиц.
2. *Метод потенциалов*. Для решения *транспортных задач* (специфическая форма ЗЛП) существуют специальные методы, один из них – это метод потенциалов.
3. *Венгерский метод*. Предназначен для решения подкласса транспортных задач, называемых *задачами о назначениях*.

5.2 Понятия транспортной задачи и транспортной модели



.....

Содержательная постановка **транспортной задачи (ТЗ)** заключается в следующем: составить план перевозок однородного груза из нескольких исходных пунктов (заводов) в исходные пункты назначения (потребления) таким образом, чтобы общая стоимость перевозок была наиболее экономичной (минимальной).

.....

ТЗ можно применять при рассмотрении ряда практических ситуаций, связанных с управлениями запасами, составлением сменных графиков, назначением служащих на работу, регулированием расхода воды в водохранилищах и многими другими. ТЗ имеет ряд важных приложений, в число которых входят задача о назначениях и задача с промежуточными пунктами. В то же время ТЗ и ее обобщения представляют собой частные случаи сетевых моделей.

Имеется m пунктов отправления A_1, A_2, \dots, A_m , которые в дальнейшем будем называть поставщиками. В них сосредоточено определенное количество единиц некоторого однородного продукта, количество которого обозначим как a_i ($i = 1, \dots, m$). Данный продукт потребляется в n пунктах B_1, B_2, \dots, B_n , которые будем называть потребителями. Объем потребления обозначим как b_j ($j = 1, \dots, n$). Известны расходы на перевозку единицы продукта из пункта A_i в пункт B_j , которые равны c_{ij} и приведены в матрице транспортных расходов $C = \{c_{ij}\}$.

При решении ТЗ используется понятие *транспортная модель (ТМ)*, которая может быть изображена в виде сети с m исходными пунктами и n пунктами назначения (см. рис. 5.1). Исходным пунктам и пунктам назначения соответствуют вершины. Дуга, соединяющая исходный пункт с пунктом назначения, представляет маршрут, по которому перевозится продукция.

Требуется составить такой план прикрепления потребителей к поставщикам, т. е. план перевозок, при котором весь продукт вывозится из пунктов A_i в пункты B_j в соответствии с потребностью и общая величина транспортных издержек будет минимальной.

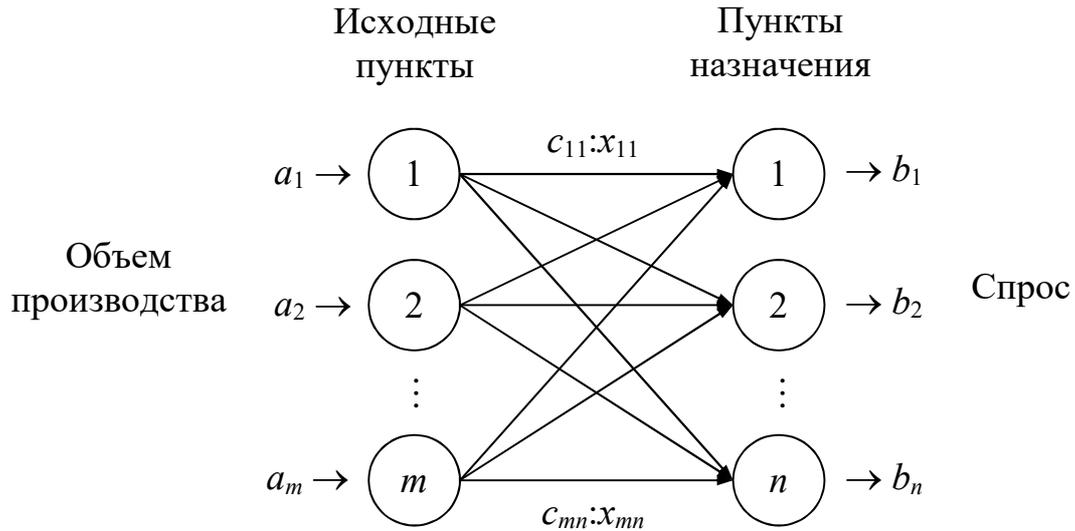


Рис. 5.1 – Транспортная модель

Обозначим количество продукта, перевозимого из пункта A_i в пункт B_j , через x_{ij} , тогда целевая функция задачи будет иметь вид:

$$f(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \quad (5.1)$$

где $X = \{x_{ij}\}$, а ограничения выглядят следующим образом:

$$\sum_{j=1}^n x_{ij} \leq a_i, \quad i = 1, \dots, m, \quad (5.2)$$

$$\sum_{i=1}^m x_{ij} \geq b_j, \quad j = 1, \dots, n, \quad (5.3)$$

$$x_{ij} > 0. \quad (5.4)$$

Первая группа ограничений (5.2) указывает, что суммарный объем перевозок продукции из некоторого исходного пункта не может превышать произведенного количества этой продукции. Ограничения (5.3) требуют, чтобы суммарные перевозки продукции в некоторый пункт потребления полностью удовлетворяли спрос на эту продукцию.

Из модели видно, что суммарный объем производства в исходных пунктах $\sum_{i=1}^m a_i$ должен быть не меньше суммарного спроса в пунктах назначения

$\sum_{j=1}^n b_j$. Если

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j, \quad (5.5)$$

то модель называют *сбалансированной (закрытой) транспортной моделью*. Она отличается от первоначальной модели тем, что ограничения (5.2) и (5.3) заменяются, соответственно, ограничениями:

$$\sum_{j=1}^n x_{ij} = a_i, \quad i = 1, \dots, m, \quad (5.6)$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = 1, \dots, n. \quad (5.7)$$

Для решения транспортной задачи можно использовать симплекс-метод, метод потенциалов и т. д. Но для этого сначала необходимо получить начальное базисное решение на основе либо метода искусственного базиса, либо метода симплексного преобразования таблицы ограничений. Однако для транспортной задачи разработаны свои методы поиска начального базиса. Это связано со следующими особенностями транспортной задачи:

- система ограничений закрытой транспортной задачи представляет собой систему равенств (5.6), (5.7) (т. е. транспортная задача задана в стандартной канонической форме ЗЛП);
- коэффициенты при переменных системы ограничений равны единице;
- каждая переменная входит в систему ограничений два раза: один раз – в систему (5.6) и один раз – в систему (5.7).

Далее рассмотрим некоторые методы поиска начального базиса – *метод северо-западного угла, метод наименьшей стоимости* и *метод Фогеля*, а затем методы решения ТЗ.

5.3 Первоначальное закрепление потребителей за поставщиками

Первым этапом решения закрытой транспортной задачи является составление начального распределения (начального плана перевозок или начального базиса). При этом следует отметить, что благодаря условию сбалансированности (5.5), ранг системы линейных уравнений (5.6), (5.7) равен $m + n - 1$. Таким образом, из общего числа $m \cdot n$ неизвестных базисных неизвестных будет $m + n - 1$. Вследствие этого при любом допустимом базисном распределении в матрице перевозок (таблице поставок) будет занято ровно $m + n - 1$ клеток, которые будем называть *базисными* в отличие от остальных, *независимых* или *свободных*, клеток.

1 Метод северо-западного угла

Следуя методу северо-западного угла (МСЗУ), начинают с того, что приписывают переменной x_{11} (расположенной в северо-западном углу таблицы) максимальное значение, допускаемое ограничениями на спрос и объем производства. После этого вычеркивают соответствующий столбец (или строку), фиксируя при этом, что остальные переменные вычеркнутого столбца (строки) полагаются равными нулю. Если ограничения, представляемые столбцом и строкой, выполняются одновременно, то можно исключить либо столбец, либо строку.

Далее, в невычеркнутой строке (столбце) максимальное значение приписывается первому невычеркнутому элементу. Процесс завершается, когда остается невычеркнутой одна строка (один столбец). Оставшимся переменным приписываются оставшиеся значения спроса (объема производства).

Алгоритм метода северо-западного угла

1. Положить индексы опорного элемента $z=1, l=1$, номер итерации $k=0$, компоненты начального базиса $X^0=0, a^0=a, b^0=b$.

2. Положить $X^{k+1}=X^k, a^{k+1}=a^k, b^{k+1}=b^k$.

3. Рассмотреть три ситуации:

а) если незаполненной осталась только одна строка, т. е. $z=m$, то ее оставшиеся элементы заполнить оставшимися значениями спроса:

$$x_{zj}^{k+1} = b_j^k, \quad j = l, l+1, \dots, n,$$

и закончить решение;

б) если незаполненной остался только один столбец, т. е. $l=n$, то его оставшиеся элементы заполнить оставшимися значениями предложения:

$$x_{il}^{k+1} = a_i^k, \quad i = z, z+1, \dots, m,$$

и закончить решение;

в) иначе вычислить значение переменной x_{zl}^{k+1} :

$$x_{zl}^{k+1} = \min(a_z^k, b_l^k).$$

4. Модифицировать значение спроса и предложения (производства):

$$a_z^{k+1} = a_z^k - x_{zl}^{k+1}, \quad b_l^{k+1} = b_l^k - x_{zl}^{k+1}.$$

5. Вычеркнуть одну строку или один столбец:

а) если $a_z^{k+1} = 0$ и $z < m$, то вычеркнуть z -ю строку и переместить опорный элемент вниз, т. е. $z = z + 1$;

б) если $b_l^{k+1} = 0$ и $l < n$, то вычеркнуть l -й столбец и переместить опорный элемент вправо, т. е. $l = l + 1$;

в) если выполняются оба условия (а) и (б), то можно, на выбор, либо вычеркнуть строку и перейти вниз, либо вычеркнуть столбец и перейти вправо, но не то и другое одновременно.

6. Положить $k = k + 1$, вернуться на шаг 2.

Метод северо-западного угла необязательно дает хорошее начальное решение транспортной задачи. Рассмотрим методы выбора более удачного маршрута.

2 Метод наименьшей стоимости

Метод наименьшей стоимости (МНС) учитывает коэффициенты ЦФ (стоимости). Вычисления производятся следующим образом. Выбирается переменная, которой соответствует наименьшая стоимость во всей таблице, и ей придается максимально большое значение (если таких переменных несколько, то берется любая из них). Вычеркивается соответствующий столбец или строка. Если ограничения по столбцу и строке выполняются одновременно, то вычеркивается либо столбец, либо строка. После вычисления новых решений спроса и объема производства для всех невычеркнутых строк и столбцов процесс повторяется при возможно большем значении той переменной, которой соответствует минимальная стоимость среди невычеркнутых. Процедура завершается, когда остается невычеркнутой одна строка (или один столбец). Оставшимся переменным приписываются оставшиеся значения спроса (объема производства).

Алгоритм метода наименьшей стоимости

1. Положить номер итерации $k = 0$, компоненты начального базиса $X^0 = 0$, $a^0 = a$, $b^0 = b$.

2. Положить $X^{k+1} = X^k$, $a^{k+1} = a^k$, $b^{k+1} = b^k$.

3. Рассмотреть три ситуации:

а) если невычеркнутой осталась только одна строка с индексом z , то ее оставшиеся элементы заполнить оставшимися значениями спроса:

$$x_{zj}^{k+1} = b_j^k,$$

где индекс j соответствует всем незачеркнутым столбцам, и закончить решение;

б) если невычеркнутой остался только один столбец с индексом l , то его оставшиеся элементы заполнить оставшимися значениями предложения

$$x_{il}^{k+1} = a_i^k,$$

где индекс i соответствует всем незачеркнутым строкам, и закончить решение;

в) иначе определить индексы опорного элемента z, l :

$$c_{zl} = \min_{i,j} c_{ij},$$

где индекс i соответствует всем незачеркнутым строкам, а индекс j – всем незачеркнутым столбцам, и вычислить значение переменной x_{zl}^{k+1} :

$$x_{zl}^{k+1} = \min(a_z^k, b_l^k).$$

4. Модифицировать значение спроса и предложения (производства):

$$a_z^{k+1} = a_z^k - x_{zl}^{k+1}, \quad b_l^{k+1} = b_l^k - x_{zl}^{k+1}.$$

5. Вычеркнуть одну строку или один столбец:

а) если $a_z^{k+1} = 0$, то вычеркнуть z -ю строку;

б) если $b_l^{k+1} = 0$, то вычеркнуть l -й столбец;

в) если выполняются оба условия (а) и (б), то можно, на выбор, вычеркнуть либо строку, либо столбец, но не то и другое одновременно.

6. Положить $k = k + 1$, вернуться на шаг 2.

3 Метод Фогеля

Метод Фогеля (МФ) является эвристическим и часто дает оптимальное или субоптимальное решение.

В данном методе на каждой итерации решения вычисляется штраф для каждой строки (столбца) вычитанием наименьшего элемента этой строки (столбца) из следующего за ним по величине элемента той же строки (столбца).

Затем отмечается строка или столбец с максимальным штрафом. Если их несколько, то отмечается любая (любой) из них. В отмеченной строке (столбце) выбирается переменная с минимальной стоимостью, и ей придается максимально возможное значение, исходя из ограничений. Далее корректируется объем производства и спроса, и вычеркивается строка или столбец, соответствующие выполненному ограничению. Если ограничения по строке и столбцу выполняются одновременно, то вычеркивается либо строка, либо столбец, а оставшемуся столбцу (строке) приписывается нулевой спрос (объем производства). Строка или столбец с нулевым объемом производства (спросом) в дальнейших вычислениях не используется.

Метод заканчивает свою работу, если:

- а) невычеркнутой остается одна строка (столбец) с положительным объемом производства (спроса). В этом случае оставшиеся базисные элементы определяются с помощью МНС;
- б) невычеркнутым строкам и столбцам соответствуют нулевые объемы производства и спроса. В этом случае оставшиеся базисные элементы также определяются с помощью МНС.



.....
 Строки и столбцы с нулевыми значениями объема производства и спроса не должны использоваться при вычислении штрафов.

Алгоритм метода Фогеля

1. Положить номер итерации $k = 0$, компоненты начального базиса $X^0 = 0$, $a^0 = a$, $b^0 = b$.

2. Положить $X^{k+1} = X^k$, $a^{k+1} = a^k$, $b^{k+1} = b^k$.

3. Рассмотреть три ситуации:

а) если невычеркнутой остается одна строка (столбец) с положительным объемом производства (спроса), то найти оставшиеся базисные элементы, используя метод наименьшей стоимости;

б) если всем невычеркнутым строкам и столбцам соответствуют нулевые объемы производства и спроса, то найти оставшиеся базисные элементы, используя метод наименьшей стоимости;

в) иначе определить индексы опорного элемента z , l . Для этого сначала вычислить штраф для каждой строки (столбца), вычитая наименьший элемент этой строки (столбца) из следующего за ним по величине элемента той же строки (столбца), и выбрать среди них максимальный:

$$p_i^k = c_{is} - c_{it}, \quad c_{it} = \min_j c_{ij}, \quad c_{is} = \min_{j \neq t} c_{ij},$$

$$r_j^k = c_{sj} - c_{ij}, \quad c_{ij} = \min_i c_{ij}, \quad c_{sj} = \min_{i \neq t} c_{ij},$$

$$pr_{\max}^k = \max \left(\max_i p_i^k, \max_j r_j^k \right).$$

Если максимальный штраф соответствует строке с индексом z , т. е. $pr_{\max}^k = p_z^k$, то считаем ее опорной, а опорный столбец ищем из условия

$$c_{zl} = \min_j c_{zj}.$$

Если максимальный штраф соответствует столбцу с индексом l , т. е. $pr_{\max}^k = r_l^k$, то считаем его опорным, а опорную строку ищем из условия

$$c_{zl} = \min_i c_{il}.$$

Индекс i везде соответствует всем незачеркнутым строкам с ненулевым предложением, а индекс j – всем незачеркнутым столбцам с ненулевым спросом. Затем вычислить значение переменной x_{zl}^{k+1} :

$$x_{zl}^{k+1} = \min(a_z^k, b_l^k).$$

4. Модифицировать значение спроса и предложения (производства):

$$a_z^{k+1} = a_z^k - x_{zl}^{k+1}, \quad b_l^{k+1} = b_l^k - x_{zl}^{k+1}.$$

5. Вычеркнуть одну строку или один столбец:

а) если $a_z^{k+1} = 0$, то вычеркнуть z -ю строку;

б) если $b_l^{k+1} = 0$, то вычеркнуть l -й столбец;

в) если выполняются оба условия (а) и (б), то можно, на выбор, вычеркнуть либо строку, либо столбец, но не то и другое одновременно.

6. Положить $k = k + 1$, вернуться на шаг 2.

5.4 Решение транспортной задачи симплекс-методом

Закрытая транспортная задача может быть решена симплексным методом.

Для этого необходимо переобозначить переменные следующим образом:

$$x_1 = x_{1,1}, \quad x_2 = x_{1,2}, \quad \dots, \quad x_n = x_{1,n};$$

$$x_{n+1} = x_{2,1}, \quad x_{n+2} = x_{2,2}, \quad \dots, \quad x_{2n} = x_{2,n};$$

.....

$$x_{(m-1)n+1} = x_{m,1}, \quad x_{(m-1)n+2} = x_{m,2}, \quad \dots, \quad x_{mn} = x_{m,n}.$$

Точно так же переобозначаем коэффициенты c_{ij} :

$$c_1 = c_{1,1}, \quad c_2 = c_{1,2}, \quad \dots, \quad c_n = c_{1,n};$$

$$c_{n+1} = c_{2,1}, \quad c_{n+2} = c_{2,2}, \quad \dots, \quad c_{2n} = c_{2,n};$$

.....

$$c_{(m-1)n+1} = c_{m,1}, \quad c_{(m-1)n+2} = c_{m,2}, \quad \dots, \quad c_{mn} = c_{m,n}.$$

Задача (5.1) с ограничениями (5.6), (5.7) примет следующий вид:

$$f(x) = \sum_{k=1}^{mn} c_k x_k \rightarrow \min, \quad (5.8)$$

$$\sum_{i=1}^m x_{(i-1)n+j} = b_j, \quad j = 1, \dots, n, \quad (5.9)$$

$$\sum_{j=1}^n x_{(i-1)n+j} = a_i, \quad i = 1, \dots, m, \quad (5.10)$$

$$x_k \geq 0, \quad k = 1, \dots, mn.$$

Таким образом, пришли к ЗЛП в стандартной форме, которую можно решить симплекс-методом.

5.5 Решение транспортной задачи методом потенциалов

Метод потенциалов позволяет улучшить начальный базисный план ТЗ, полученный рассмотренными выше методами, и получить оптимальное решение ТЗ.

Алгоритм метода потенциалов

1. Построить систему потенциалов. Введем специальные показатели u_i для каждой строки i , которые можно интерпретировать как цену продукта в пункте поставщика, а каждому столбцу j (j -му потребителю) устанавливается потенциал v_j , который можно принять условно за цену продукта в пункте потребителя. В простейшем случае цена продукта в пункте потребителя равна его цене в пункте поставщика плюс транспортные расходы на его доставку, т. е.

$$v_j = u_i + c_{ij}. \quad (5.11)$$

Совокупность уравнений вида (5.11), составленных для всех заполненных клеток (всех базисных неизвестных), образует систему $m + n - 1$ линейных уравнений с $m + n$ неизвестными u_i и v_j . Эта система всегда совместна, причем значение одного из неизвестных можно задавать произвольно (например, $u_1 = 0$), тогда значения остальных неизвестных находятся из системы однозначно.

2. Оценить оптимальность распределения, определив оценки для всех клеток матрицы перевозок, или, иначе, транспортной таблицы (ТТ). Оценки d_{ij} определяются по формуле

$$d_{ij} = (u_i + c_{ij}) - v_j. \quad (5.12)$$

Используя ранее принятую интерпретацию, выражение $(u_i + c_{ij})$ можно трактовать как сумму цены продукта у поставщика и стоимости перевозки; эта сумма путем вычитания сравнивается с ценой продукта у соответствующего потребителя v_j . Очевидно, оценки заполненных клеток равны нулю (цена потребителя покрывает цену поставщика и стоимость перевозок). Таким образом,

об оптимальности распределения можно судить по величинам оценок свободных клеток. Если оценка некоторой свободной клетки отрицательна, это можно интерпретировать так: цена, предлагаемая соответствующим потребителем, больше суммы цены поставщика и стоимости перевозки, т. е. если бы эта клетка была занята, то можно было бы получить дополнительный экономический эффект. Следовательно, условие оптимальности распределения будет следующим: *оценки свободных клеток матрицы перевозок должны быть неотрицательными*. Оценки клеток по формуле (5.12) удобно представить в виде матрицы оценок $D = \{d_{ij}\}$.



.....

Кроме того, следует отметить, что если оценки всех свободных клеток строго больше нуля, то любой другой план, предусматривающий занятие хотя бы одной из этих клеток, будет менее оптимален. Это говорит о том, что найденный оптимальный план будет являться *единственным*. Наличие нулевых оценок свободных клеток в оптимальном плане перевозок, наоборот, свидетельствует о *неединственности* оптимального плана.

.....

3. Выбрать опорный элемент. Чтобы улучшить неоптимальный план перевозок, выбирается клетка матрицы перевозок с отрицательной оценкой; если таких клеток несколько, то обычно (но необязательно) выбирается клетка с наибольшей по абсолютной величине отрицательной оценкой.

4. Построить *контур* для опорного элемента. Для выбранной клетки строится замкнутая линия, начальная вершина которой лежит в выбранной клетке, а все остальные вершины находятся в занятых клетках; при этом направления отдельных отрезков контура могут быть только горизонтальными и вертикальными. Вершиной контура, кроме первой, является занятая клетка, где отрезки контура образуют один прямой угол (нельзя рассматривать как вершины клетки, где горизонтальные и вертикальные отрезки контура пересекаются). Очевидно, число отрезков контура, как и его вершин, будет четным. В вершинах контура расставляются поочередно знаки «+» и «-», начиная со знака «+» в выбранной свободной клетке.

5. Перераспределить поставки. Величина перераспределяемой поставки определяется как наименьшая из величин поставок в вершинах контура со знаком «-», и на эту величину увеличиваются поставки в вершинах со знаком «+»

и уменьшаются поставки в вершинах со знаком «-». Это правило гарантирует, что в вершинах контура не появится отрицательных поставок, начальная выбранная клетка окажется занятой, в то время как одна из занятых клеток при этом обязательно освободится. Если величина перераспределяемой поставки равна поставкам не в одной, а в нескольких вершинах контура со знаком «-», то освобождается только одна клетка, обычно с наибольшей стоимостью перевозки, а все другие такие клетки остаются занятыми с нулевой поставкой. После этого вернуться на шаг 1.



.....

*Транспортные задачи, в базисном плане перевозок которых имеют место занятые клетки с нулевой поставкой (или в первоначальном распределении, или в процессе итераций), называются **вырожденными**.*

.....

В случае вырожденной транспортной задачи существует опасность зацикливания, т. е. бесконечного повторения итераций (бесконечного перебора одних и тех же базисных комбинаций занятых клеток). Как правило, в практических задачах транспортного типа зацикливание не встречается; тем не менее, следует знать, что существуют специальные правила, позволяющие выйти из цикла, если зацикливание все же произойдет. При отсутствии вырождения метод потенциалов конечен и приводит к оптимальному плану перевозок за конечное число шагов.

5.6 Задача о назначениях

Рассмотрим еще один вид задач, называемых *задачами о назначениях* (ЗоН). Математически такие задачи относятся к тому же типу распределительных задач, что и рассмотренная ранее транспортная задача. Отличие в том, что в них объемы наличных и требующихся ресурсов для выполнения каждой работы равны единице ($a_i = b_j = 1$), а все переменные x_{ij} либо равны единице, если i -й работник назначен на j -ю работу, либо равны нулю в других случаях. Исходные данные задачи о назначениях группируются в таблице, которая называется *матрицей оценок*, а результаты – в *матрице назначений*.

Задача о назначениях в общем виде может быть сформулирована следующим образом. Имеется n работников, которые могут выполнять n работ, причем использование i -го работника на j -й работе, например, приносит до-

ход c_{ij} . Требуется поручить каждому из работников выполнение одной вполне определенной работы, чтобы максимизировать, в данном случае, суммарный доход.

Введем переменные:

$$x_{ij} = \begin{cases} 1, & \text{если работник } i \text{ выполняет работу } j; \\ 0 & \text{в противном случае.} \end{cases}$$

Задача состоит в том, чтобы найти распределение $X = \{x_{ij}\}$ работников по работам (т. е. найти матрицу назначений), которое максимизирует целевую функцию

$$f(X) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max \quad (5.13)$$

при ограничениях

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n; \quad (5.14)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n, \quad (5.15)$$

причем x_{ij} равны либо 0, либо 1 (так называемые *булевы переменные*) для всех $i, j = 1, \dots, n$.

Ограничения (5.14) отражают условие того, что за каждым работником может быть закреплена только одна работа, а ограничения (5.15) означают, что для выполнения каждой работы может быть выделен только один работник.

Если в задаче о назначениях элементы матрицы расходов $C = \{c_{ij}\}$ представляют собой, например, время выполнения каждым работником любой из работ, или издержки при выполнении работ, то целевая функция этой задачи будет минимизироваться.

К ЗоН относится рассмотренная ранее задача оптимизации плана доставки хлеба. Очевидно, что ее можно решить, например, с помощью симплекс-метода. Как было сказано выше, для решения таких задач может потребоваться метод Гомори. Однако если решение задачи единственное, то его можно найти одним только симплекс-методом. Если же решений несколько, тогда результат может получиться дробным, в таком случае потребуется применение метода Гомори.

Также ЗоН можно решить с помощью методов решения ТЗ. Например, сначала получить начальный базис, используя МСЗУ, МСН или МФ, а потом улучшить решение методом потенциалов.

Но наиболее простым будет решение, полученное специальными методами решения ЗоН.

5.7 Венгерский метод решения задачи о назначениях

Рассмотрим метод решения задачи о назначениях. Основная идея этого метода: оптимальность решения задачи не нарушается при уменьшении (увеличении) элементов строки (столбца) на одну и ту же величину d_i . Решение считают оптимальным, если все измененные искусственно затраты $c'_{ij} \geq 0$, $i, j = 1, \dots, n$, и можно отыскать такой набор x_{ij} , при котором достигается экстремум целевой функции (5.13).

Алгоритм венгерского метода

1. Получить нули в каждой строке.

1.1. Найти наименьший элемент d_i в каждой строке и вычесть из всех ее элементов.

1.2. Аналогично в каждом столбце определить его минимальный элемент d_j и вычесть из всех его элементов.

2. Проверить оптимальность решения.

2.1. Рассмотреть одну из строк ТТ, имеющую меньшее число нулей, отметить один из нулей этой строки и зачеркнуть все остальные нули этой строки и того столбца, в котором находится этот нуль.

2.2. Аналогичные операции выполнить последовательно для всех строк.

2.3. Если назначения, которые получены при всех отмеченных нулях, являются полными, то есть число отмеченных нулей равно n , то решение является оптимальным. В противном случае перейти к шагу 3.

3. Поиск минимального набора строк и столбцов, содержащих нули.

3.1. Отметить:

а) все строки, в которых нет ни одного отмеченного нуля;

б) все столбцы, содержащие перечеркнутый нуль хотя бы в одной из отмеченных строк;

в) все строки, содержащие отмеченные нули хотя бы в одном из отмеченных звездочкой столбцов.

3.2. Шаги (3.1б) и (3.1в) повторять поочередно до тех пор, пока есть что отмечать.

3.3. После этого зачеркнуть каждую непомеченную строку и каждый помеченный столбец с целью провести минимальное число горизонтальных и вертикальных прямых, пересекающих по крайней мере один раз все нули.

4. Перестановка некоторых нулей.

4.1. Определить наименьшее число из тех клеток, через которые не проведены прямые (не зачеркнуты).

4.2. Это число вычесть из каждого числа невычеркнутых столбцов и прибавить к каждому числу вычеркнутых строк в вычеркнутых столбцах.

4.3. Вернуться на шаг 2.



..... Контрольные вопросы по главе 5

1. Сформулируйте модель транспортной задачи.
2. Приведите классификацию методов решения ТЗ.
3. Дайте определение сбалансированной модели ТЗ. Приведите способы балансировки ТМ.
4. Приведите приближенные методы решения ЗЛП на примере транспортной задачи.
5. Как решить транспортную задачу симплекс-методом?
6. В чем заключается суть метода потенциалов?
7. Сформулируйте задачу о назначениях.
8. Как решить задачу о назначениях симплекс-методом?
9. Как решить задачу о назначениях, используя методы решения ТЗ?
10. Опишите венгерский метод решения задачи о назначениях.

6 Нелинейное программирование

Усложним задачу еще больше. Если в предыдущих случаях мы рассматривали задачи условной оптимизации с линейными ЦФ и ограничениями (ЗЛП и ЗЦП, а также их разновидности – ТЗ и ЗоН), то сейчас рассмотрим нелинейные задачи. Такие задачи называют *задачами нелинейного программирования* (ЗНП), и под ними понимают класс задач, в которых ЦФ или ограничения (или и то, и другое вместе) нелинейные. Как уже указывалось ранее, ограничения существенно уменьшают размеры области, в которой проводится поиск экстремума. Далее рассмотрим методы решения таких задач.

6.1 Классификация методов

Выполним классификацию методов решения ЗНП в зависимости от типов ЗО, для решения которых они предназначены [15]:

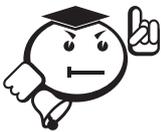
1. ЗНП с ограничениями-неравенствами:

$$f(x) \rightarrow \min, \quad x \in R^n, \quad (6.1)$$

$$g_j(x) \geq 0, \quad j = 1, 2, \dots, m. \quad (6.2)$$

Здесь $f(x)$ и $g_j(x)$ – нелинейные функции. Для решения таких задач применяются следующие методы:

- линеаризации;
- допустимых направлений Зойтендейка;
- проекции градиента для случая нелинейных ограничений;
- один из вариантов метода условного градиента;
- ряд штрафных методов.



.....

В методах условного градиента и проекции градиента требуется находить проекцию точки на ОДР задачи. Поиск проекции точки даже на такие фигуры, как парабола, эллипс и т. п., является достаточно сложной самостоятельной нелинейной задачей.

.....

2. ЗНП с ограничениями-равенствами:

$$f(x) \rightarrow \min, \quad x \in R^n, \quad (6.3)$$

$$h_k(x) = 0, \quad k = 1, 2, \dots, m. \quad (6.4)$$

Здесь $f(x)$ и $h_k(x)$ – также нелинейные функции. Для решения таких задач применяются следующие методы:

- замены переменных;
- множителей Лагранжа;
- линеаризации;
- ряд штрафных методов.

3. ЗНП общего вида:

$$f(x) \rightarrow \min, \quad x \in R^n, \quad (6.5)$$

$$g_j(x) \geq 0, \quad j = 1, 2, \dots, m, \quad (6.6)$$

$$h_k(x) = 0, \quad k = 1, 2, \dots, p. \quad (6.7)$$

Это самый сложный тип ЗНП. Задачи такого вида может решать метод линеаризации, но он очень неустойчивый. Также можно использовать штрафные функции с комбинацией двух видов штрафов – первого для ограничений-равенств и второго для ограничений-неравенств. В некоторых случаях удастся свести неравенства к равенствам (или наоборот), тогда для решения можно использовать другие методы, перечисленные в первых двух пунктах.

4. ЗНП с линейными ограничениями:

$$f(x) \rightarrow \min, \quad x \in R^n, \quad (6.8)$$

$$g_j(x) = a^T x \geq 0, \quad j = 1, 2, \dots, m. \quad (6.9)$$

Для решения таких задач, в принципе, можно использовать методы, перечисленные в первом пункте, т. к. это частный случай ЗНП с ограничениями-неравенствами. Но при этом может ухудшиться сходимость. Дополнительно можно использовать:

- метод Франка – Вульфа;
- один из вариантов метода условного градиента;
- метод проекции градиента для случая линейных ограничений.

5. ЗНП вида

$$f(x) \rightarrow \min, \quad x \in R^n, \quad (6.10)$$

$$a_i \leq x_i \leq b_i, \quad i = 1, 2, \dots, n. \quad (6.11)$$

Здесь ОДР представляет собой прямоугольник (при $n > 2$ – прямоугольный параллелепипед). Задачи такого типа решаются одним из вариантов метода условного градиента, хотя могут быть решены и другими методами, например перечисленными в пункте 4.

6.2 Задачи с ограничениями в виде равенств

Рассмотрим общую задачу оптимизации с ограничениями-равенствами (6.3)–(6.4). Функции $f(x)$ и $h_k(x)$ предполагаются непрерывными и дифференцируемыми [1].

6.2.1 Метод замены переменных

Задача нелинейного программирования, в принципе, может быть решена как задача безусловной оптимизации, путем исключения из целевой функции (6.3) независимых переменных с помощью заданных равенств (6.4). Наличие ограничений в виде равенств фактически позволяет уменьшить размерность исходной задачи с n до $(n - m)$. Для решения задачи безусловной минимизации можно использовать известные методы, изложенные ранее.

Пусть имеется m ограничений равенств, причем $m < n$. Если $m = n$, то задача оптимизации сводится к решению системы уравнений (6.4) и оптимизация здесь не нужна.

Таким образом, экстремальная точка функции $f(x)$ ищется лишь по тем значениям x , которые удовлетворяют независимой системе (6.4) при $m < n$, т. е. в некоторой части D пространства R^n ($D \subset R^n$).

Данный метод называется *методом замены переменных* (МЗП) и является аналитическим.

Алгоритм МЗП

1. Представить первые m переменных x_1, \dots, x_m , относительно которых мы хотим разрешить систему (6.4), в виде вектора y , а остальные переменные – в виде вектора u , т. е. вектор x можно представить в виде:

$$x = \begin{pmatrix} y \\ u \end{pmatrix}; \quad y = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \equiv \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}; \quad u = \begin{pmatrix} x_{m+1} \\ \vdots \\ x_n \end{pmatrix} \equiv \begin{pmatrix} u_1 \\ \vdots \\ u_{n-m} \end{pmatrix}. \quad (6.12)$$



.....
 В общем случае можно выбирать не первые переменные, а любые m переменных.

2. Решить систему (6.4):

$$y = \Psi(u), \quad u \in R^{n-m}. \quad (6.13)$$

Если $m = n$, то перейти на шаг 4. В противном случае необходимо подставить y в целевую функцию $F(u) = f(y, u) = f[\Psi(u), u]$. Получим функцию $(n - m)$ переменных u_1, \dots, u_{n-m} , на которую не наложено никаких ограничений, т. е. получим задачу оптимизации без ограничений.

3. Использовать один из известных методов безусловной оптимизации для идентификации стационарных точек целевой функции.

4. Выбрать из полученных точек оптимальное решение. Для этого необходимо использовать рассмотренные далее необходимые и достаточные условия оптимальности решения ЗНП.

МЗП применим лишь в случаях, когда уравнения ограничения можно разрешить относительно некоторого конкретного набора независимых переменных. При наличии большого числа ограничений МЗП становится весьма трудоемкой процедурой. Кроме того, возможны ситуации, когда уравнение не удается разрешить относительно переменной, например:

$$h(x) = x_1^2 \cdot x_3 + x_2 \cdot x_3^2 + x_2^{-1} \cdot x_1.$$

В этом случае целесообразно использовать метод множителей Лагранжа.

6.2.2 Метод множителей Лагранжа

С помощью *метода множителей Лагранжа* (ММЛ) устанавливают необходимое условие, позволяющее идентифицировать точки экстремума в задаче оптимизации с ограничениями-равенствами. При этом задача с ограничениями преобразуется в эквивалентную задачу безусловной оптимизации, в которой фигурируют некоторые неизвестные параметры – множители Лагранжа [1–2, 5, 9, 14].

Рассмотрим задачу оптимизации:

$$f(x) \rightarrow \min, \quad (6.14)$$

$$h(x) = 0. \quad (6.15)$$

В соответствии с ММЛ эта задача преобразуется в задачу оптимизации без ограничений

$$L(x, \lambda) = f(x) + \lambda h(x) \rightarrow \min. \quad (6.16)$$

Здесь $L(x, \lambda)$ – функция Лагранжа, $\lambda = \text{const}$ – множитель Лагранжа, на знак которого никаких требований не накладывается.

ММЛ можно распространить на случай, когда задача оптимизации имеет несколько ограничений-равенств. Рассмотрим общую задачу (6.3)–(6.4). Функция Лагранжа принимает следующий вид:

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i h_i(x), \quad (6.17)$$

где $\lambda_1, \dots, \lambda_m$ – множители Лагранжа, т. е. неизвестные параметры, значения которых необходимо определить.

Алгоритм ММЛ

1. Составить функцию Лагранжа.
2. Найти частные производные

$$\frac{\partial L}{\partial x_j}, \frac{\partial L}{\partial \lambda_i}, \quad j = 1, 2, \dots, n, \quad i = 1, 2, \dots, m$$

и, приравняв их нулю, составить расширенную систему $(n + m)$ уравнений с $(n + m)$ неизвестными x_j, λ_i :

$$\begin{cases} \nabla_x L = 0 \\ \nabla_\lambda L = 0 \end{cases} \Rightarrow \begin{cases} \frac{\partial L}{\partial x_j} = \frac{\partial f(x)}{\partial x_j} + \sum_{i=1}^m \lambda_i \frac{\partial h_i(x)}{\partial x_j} = 0, \quad j = 1, 2, \dots, n, \\ \frac{\partial L}{\partial \lambda_i} = h_i(x) = 0, \quad i = 1, 2, \dots, m. \end{cases} \quad (6.18)$$



.....
 Это и есть *необходимое условие экстремума функции Лагранжа*.

3. Решить полученную систему и определить стационарные точки функции L .

4. Классифицировать полученные стационарные точки на основе вычисления элементов матрицы Гессе функции L , рассматриваемой как функции только аргумента x :

$$H_L(x, \lambda) = \begin{pmatrix} \frac{\partial^2 L}{\partial x_1^2} & \dots & \frac{\partial^2 L}{\partial x_1 \partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial^2 L}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 L}{\partial x_n^2} \end{pmatrix}. \quad (6.19)$$

Выбрать из полученных точек оптимальное решение x^* .

Точка (x^*, λ^*) является седловой точкой функции Лагранжа $L(x, \lambda)$, если функцию Лагранжа рассматривать как функцию переменных x и λ .

Для некоторых задач расширенная система $(n + m)$ уравнений с $(n + m)$ неизвестными может не иметь решений, и ММЛ окажется неприемлемым. Однако на практике такие задачи встречаются достаточно редко.

6.2.3 Экономическая интерпретация множителей Лагранжа

Множители Лагранжа рассматривались выше как параметры, значения которых выбираются таким образом, чтобы выполнялись ограничения задачи. С экономической точки зрения множители Лагранжа интерпретируются как неясные (теневые) цены ресурсов, определяемых ограничениями; оптимальные значения множителей Лагранжа играют важную роль в анализе чувствительности решений [1]. Для того чтобы пояснить эту интерпретационную схему, рассмотрим следующую задачу оптимизации с двумя переменными и одним ограничением в виде равенства:

$$\begin{aligned} f(x_1, x_2) &\rightarrow \min, \\ h(x_1, x_2) &= b, \end{aligned} \quad (6.20)$$

где константа b характеризует наличие некоторого ресурса. Составим функцию Лагранжа:

$$L(x, \lambda) = f(x) + \lambda \cdot [h(x) - b].$$

Предположим, что стационарная точка функции L соответствует глобальному минимуму:

$$\begin{cases} \frac{\partial L}{\partial x_1} = \frac{\partial f}{\partial x_1} + \lambda \frac{\partial h}{\partial x_1} = 0, \\ \frac{\partial L}{\partial x_2} = \frac{\partial f}{\partial x_2} + \lambda \frac{\partial h}{\partial x_2} = 0, \end{cases} \quad (6.21)$$

$$\frac{\partial L}{\partial \lambda} = h(x) - b = 0. \quad (6.22)$$

Пусть λ^* – оптимальное значение множителя Лагранжа, а x^* – оптимальное решение задачи.

Далее, пусть минимум функции $L(x, \lambda)$ при $\lambda = \lambda^*$ достигается в точке $x = x^*$, причем

$$\begin{aligned} h(x^*) &= b, \\ f(x^*) &= L(x^*, \lambda^*) = f^*. \end{aligned}$$

Очевидно, что оптимальные значения (x^*, λ^*) связаны функциональной зависимостью с величиной b , задающей границу наличия дефицитного ресурса.

Изменения f^* (оптимального значения целевой функции), обусловленные изменениями b , описываются частной производной $\partial f^* / \partial b$. По правилу дифференцирования сложной функции имеем:

$$\frac{\partial f^*}{\partial b} = \frac{\partial f^*}{\partial x_1^*} \cdot \frac{\partial x_1^*}{\partial b} + \frac{\partial f^*}{\partial x_2^*} \cdot \frac{\partial x_2^*}{\partial b}. \quad (6.23)$$

Дифференцируя обе части ограничения $h(x) - b = 0$, получаем:

$$\frac{dh}{db} = \frac{\partial h}{\partial x_1^*} \cdot \frac{\partial x_1^*}{\partial b} + \frac{\partial h}{\partial x_2^*} \cdot \frac{\partial x_2^*}{\partial b} - 1 = 0. \quad (6.24)$$

Умножим обе части равенства (6.24) на λ^* и суммируем с выражением (6.23), получим:

$$\frac{\partial f^*}{\partial b} = -\lambda^* + \sum_{j=1}^2 \left[\frac{\partial f^*}{\partial x_j^*} + \lambda^* \cdot \frac{\partial h}{\partial x_j^*} \right] \cdot \frac{\partial x_j^*}{\partial b}. \quad (6.25)$$

Так как x^* и λ^* удовлетворяют уравнениям (6.22) и (6.23), то равенство (6.25) приводится к виду:

$$\frac{\partial f^*}{\partial b} = -\lambda^*. \quad (6.26)$$

Таким образом, из выражения (6.26) следует, что скорость изменения оптимального значения f^* , вызываемого изменением b , определяется оптимальным значением λ^* . Другими словами, величина изменения оптимального значения целевой функции, обусловленного единичным увеличением правой части ограничения, задается множителем Лагранжа. В зависимости от знака λ^* значения f^* при изменении b могут увеличиваться или уменьшаться. При m ограничениях имеем:

$$\frac{\partial f^*}{\partial b_i} = -\lambda_i^*, \quad i = 1, 2, \dots, m.$$

6.3 Необходимые и достаточные условия оптимальности

Ранее были рассмотрены необходимые и достаточные условия оптимальности решения задач оптимизации без ограничений. Ограничения на управляемые переменные существенно уменьшают размеры области, в которой проводится поиск экстремума (ОДР). На первый взгляд может показаться, что уменьшение размеров ОДР должно упростить процедуру поиска экстремума. Но, напротив, процесс оптимизации становится более сложным, т. к. установленные выше критерии оптимальности нельзя использовать при наличии ограничений. Может нарушаться даже основное условие – равенство нулю первой производной в стационарной точке [1–3, 8].

Например, безусловный минимум функции $f(x) = (x - 2)^2$ имеет место при $x^* = 2$ (т. е. $\nabla f(x^*) = 0$). Но если задача решается с учетом ограничения $x \geq 4$, то условный минимум достигается в точке $x^* = 4$, где $\nabla f(x^*) = 4 \neq 0$. Может показаться, что условный минимум достигается на границе ОДР, но и это не так. Например, если решать задачу с учетом ограничения $x \leq 4$, то минимумом останется точка $x^* = 2$.

Если ограничений в задаче несколько, то процесс ее решения усложняется еще сильнее, т. к. условный минимум:

- может совпадать с безусловным минимумом, т. е. являться внутренней точкой ОДР, т. е. не лежать на поверхности ни одного из ограничений (в этом случае $\nabla f(x^*) = 0$);
- может лежать на поверхности некоторых ограничений, но не лежать на поверхности других, т. е. являться граничной точкой ОДР (в этом случае градиент $\nabla f(x^*)$ может быть нулевым, а может не быть);
- может лежать на пересечении поверхностей всех ограничений (в этом случае обычно $\nabla f(x^*) \neq 0$, но не обязательно).

Будем называть ограничение-неравенство $g(x)$ *активным* в некоторой точке x , если данная точка лежит на его поверхности (т. е. обращает его в равенство), и *неактивным*, если данная точка не лежит на его поверхности.



..... Пример 6.1

Например, ограничение $g(x) = 4 - x_1^2 - x_2^2 \geq 0$ в точке $x = (0, 2)^T$ является активным, т. к. $4 - 0 - 4 = 0$, а в точке $x = (1, 1)^T$ – неактивным, т. к. $4 - 1 - 1 = 2 > 0$.

.....

Если в точке x ограничение $g(x)$ выполняется, то такая точка называется *внутренней* (т. е. она лежит внутри ОДР, определяемой данным ограничением), и *внешней*, если ограничение $g(x)$ для нее не выполняется (т. е. она лежит вне ОДР).



..... Пример 6.2

Например, для ограничения $g(x) = 4 - x_1^2 - x_2^2 \geq 0$ точка $x = (1, 1)^T$ является внутренней, т. к. $4 - 1 - 1 = 2 \geq 0$, а точка $x = (2, 2)^T$ – внешней, т. к. $4 - 4 - 4 = -4 < 0$.

.....

6.3.1 Необходимые и достаточные условия оптимальности задач с ограничениями общего вида

Выше было установлено, что множители Лагранжа можно использовать при построении критериев оптимальности для задач оптимизации с ограничениями в виде равенств. Кун и Таккер обобщили этот подход на случай общей задачи нелинейного программирования с ограничениями как в виде равенств, так и в виде неравенств.

Рассмотрим общую задачу нелинейного программирования

$$\begin{aligned} f(x) &\rightarrow \min, \\ g_i(x) &\begin{cases} \geq 0, \\ \leq 0, \end{cases} \quad i = 1, 2, \dots, m, \\ h_l(x) &= 0, \quad l = 1, 2, \dots, p, \end{aligned} \tag{6.27}$$

где $f(x)$, $g_i(x)$, $h_l(x)$ – непрерывные и дифференцируемые функции.

Необходимые и достаточные условия первого порядка выражаются теоремой, приведенной ниже.



.....
Теорема. Для того чтобы точка x^* была глобальным минимумом задачи (6.27), необходимо и достаточно, чтобы выполнялись следующие условия:

- а) ЦФ $f(x)$ была выпукла;
- б) функция-ограничение $g_i(x)$ – вогнута, если $g_i(x) \geq 0$, и выпукла, если $g_i(x) \leq 0$;
- в) ограничения-равенства $h_l(x)$ – линейные функции;
- г) существовали такие числа $\mu_i^* \geq 0$ и λ_l^* (произвольного знака), что

$$\nabla f(x^*) \mp \sum_{i=1}^m \mu_i^* \nabla g_i(x^*) + \sum_{l=1}^p \lambda_l^* \nabla h_l(x^*) = 0, \quad (6.28)$$

$$\mu_i^* g_i(x^*) = 0, \quad i = 1, 2, \dots, m, \quad (6.29)$$

$$h_l(x^*) = 0, \quad l = 1, 2, \dots, p. \quad (6.30)$$

Знак «минус» перед вторым слагаемым ЦФ ставится в том случае, если $g_i(x) \geq 0$, и «плюс» – если $g_i(x) \leq 0$.

Для выполнения условий Куна – Таккера необходимо, чтобы:

- 1) существовала такая точка \bar{x} , что $g_i(\bar{x}) > 0$ (либо $g_i(\bar{x}) < 0$, если $g_i(x) \leq 0$) и $h_l(\bar{x}) = 0$;
- 2) в точке x^* градиенты $\nabla g_i(x^*)$ и $\nabla h_l(x^*)$ были линейно независимы для всех $i \in I = \{i \mid g_i(x^*) = 0\}$ и $l = 1, 2, \dots, p$, т. е. для тех i , для которых ограничения g_i активны в точке x^* .

В случаях задачи с ограничениями только типа равенств

$$\begin{aligned} f(x) &\rightarrow \min, \\ h_l(x) &= 0, \quad l = 1, 2, \dots, p, \end{aligned}$$

функция Лагранжа имеет вид

$$L(x, \lambda) = f(x) + \sum_{l=1}^p \lambda_l h_l(x).$$

Для этой функции условия оптимальности первого порядка выражаются следующими уравнениями:

$$\nabla_x L(x, \mu) = \nabla f(x) + \sum_{l=1}^p \lambda_l \nabla h_l(x),$$

$$\nabla_{\lambda} L(x, \mu) = h_l(x) = 0, \quad l = 1, 2, \dots, p.$$

Таким образом, условия Куна – Таккера (6.28) и (6.30) совпадают с условиями оптимальности для задачи Лагранжа.

Если данные условия выполняются, то точка является оптимальной. Но если нет, то нельзя утверждать, что она не оптимальна. В таких случаях следует использовать критерий оптимальности второго порядка.

6.3.2 Необходимые и достаточные условия оптимальности второго порядка

Выше рассматривались необходимые и достаточные условия первого порядка – это условия Куна – Таккера. Данные условия применимы к задачам условной оптимизации и формулируются через градиенты ЦФ и функций, входящих в ограничения. При этом на функции ограничения $g_i(x)$ накладываются жесткие условия выпуклости (или вогнутости), а ограничения-равенства $h_l(x)$ должны быть линейными, что выполняется не для всех ЗНП. Необходимые и достаточные условия оптимальности второго порядка для задач с дважды дифференцируемыми функциями построены Мак-Кормиком.

Рассмотрим следующую задачу нелинейного программирования:

$$\begin{aligned} f(x) &\rightarrow \min, \quad x \in R^n, \\ g_i(x) &\geq 0, \quad i = 1, 2, \dots, m, \\ h_l(x) &= 0, \quad l = 1, 2, \dots, p. \end{aligned} \quad (6.31)$$

Пусть $f(x), g_i(x), h_l(x)$ – дважды дифференцируемые функции, x^* – допустимое решение (6.31).

Множество активных в точке x^* ограничений определяется множеством $I = \{i \mid g_i(x^*) = 0\}$.

Предположим, что $\nabla g_i(x^*)$ при $i \in I$ и $\nabla h_l(x^*)$ при $l = 1, 2, \dots, p$ линейно независимы.



.....

Теорема. Для того чтобы точка x^* была точкой строгого локального минимума, необходимо и достаточно выполнение следующих условий:

1) существует такой вектор (λ^*, μ^*) , что вектор (x^*, μ^*, λ^*) определяет точку Куна – Таккера;

2) для любого ненулевого вектора $y \in R^n$, удовлетворяющего системе уравнений и неравенств:

$$\nabla g_i(x^*)^T \cdot y = 0, \quad i \in I_1 = \{i \mid g_i(x^*) = 0, \mu_i^* > 0\}, \quad (6.32)$$

$$\nabla g_i(x^*)^T \cdot y \geq 0, \quad i \in I_2 = \{i \mid g_i(x^*) = 0, \mu_i^* = 0\}, \quad (6.33)$$

$$\nabla h_l(x^*)^T \cdot y = 0, \quad l = 1, 2, \dots, p, \quad (6.34)$$

выполняется неравенство

$$y^T H_L(x^*, \lambda^*, \mu^*) y > 0, \quad (6.35)$$

где H_L – матрица Гессе функции $L(x, \lambda, \mu)$:

$$H_L(x, \lambda, \mu) = \left\| \frac{\partial^2 L}{\partial x_i \partial x_j} \right\|_{\substack{x=x^* \\ \lambda=\lambda^* \\ \mu=\mu^*}},$$

$$L(x, \lambda, \mu) = f(x) - \sum_{i=1}^m \mu_i g_i(x) + \sum_{l=1}^p \lambda_l h_l(x).$$



.....

 Множество $I = I_1 \cup I_2$ содержит номера всех активных ограничений в точке x^* .

6.4 Методы штрафов

Здесь мы продолжим изучение ЗО с ограничениями. Нас будут интересовать методы, существенно отличающиеся от рассмотренных выше методов. Их общий принцип заключается в замене исходной задачи на решение последовательности экстремальных задач без ограничений [1, 5–6, 8, 10, 14].

Рассмотрим задачу:

$$f(x) \rightarrow \min, \quad x \in R^n, \quad (6.36)$$

$$g_i(x) \geq 0, \quad i = 1, 2, \dots, m; \quad (6.37)$$

$$h_l(x) = 0, \quad l = 1, 2, \dots, p.$$

Предполагается, что точка x^* является решением этой задачи, известно некоторое начальное приближение x^0 , возможно недопустимое, т. е. не удовлетворяющее отношениям (6.37). С помощью рассматриваемых далее алгоритмов в пространстве R^n строится конечная последовательность точек $x^t, t = 0, 1, \dots, k$, которая начинается с заданной точки x^0 и заканчивается точкой x^k , дающей наилучшее приближение к точке экстремума x^* среди всех точек построенной последовательности. В качестве точек последовательности $\{x^t\}$ берутся стационарные точки штрафной функции – ЦФ вспомогательной задачи безусловной минимизации.

Конкретные методы, основанные на указанной общей схеме, определяются видом штрафной функции, а также правилами, по которым производится пересчет штрафных параметров по окончании очередного цикла безусловной минимизации. Идея преобразования задачи с ограничениями в последовательность задач без ограничений, построенную надлежащим образом, представляется заманчивой, главным образом, в связи с наличием эффективных и надежных методов безусловной минимизации. При этом будем надеяться искать условный минимум с приемлемой точностью путем решения относительно небольшого числа не слишком сложных задач.



.....

*В зависимости от того, являются ли элементы последовательности $\{x^t\}$ допустимыми или недопустимыми точками, говорят соответственно о методах **внутренней** (рис. 6.1) или **внешней точки** (рис. 6.2). Иногда их называют **методами внутреннего или внешнего штрафа**. Методы внутренних штрафных функций называют также **методами барьерных функций**.*

.....

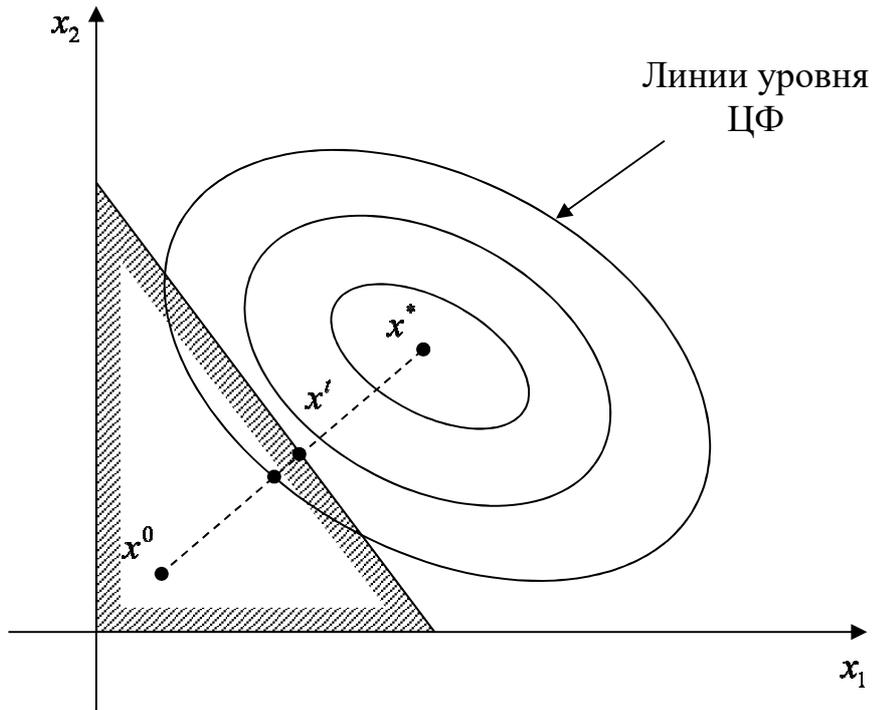


Рис. 6.1 – Метод внутренних штрафов

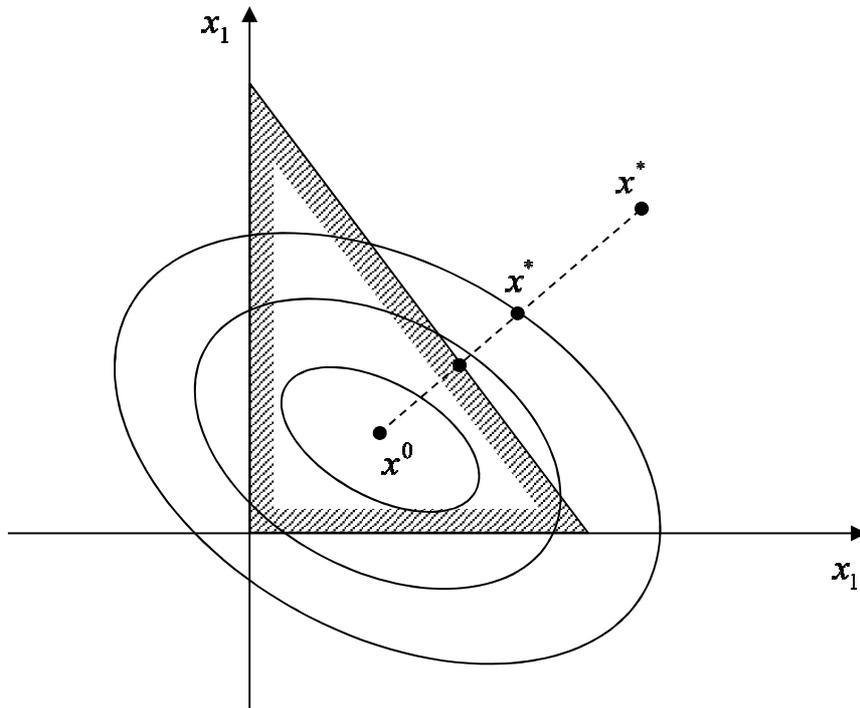


Рис. 6.2 – Метод внешних штрафов

6.4.1 Понятие штрафных функций

Штрафная функция (ШФ) определяется выражением:

$$P(x, R) = f(x) + H[R, g(x), h(x)], \quad (6.38)$$

где R – набор штрафных параметров, H – штраф-функция R и ограничений,

$$H[R, g(x), h(x)] = R \cdot \varphi[g(x), h(x)].$$

Выражение для штрафа H определяется таким образом, чтобы допустимые точки задачи имели преимущество перед недопустимыми в отношении безусловной минимизации ШФ.

Методы внутренней точки связаны с такими функциями H , при которых стационарные точки функции $P(x, R)$ оказываются заведомо допустимыми. Эти методы называют также методами барьеров, поскольку здесь штраф как бы создает вдоль границы допустимой области S барьер из бесконечно больших значений функции P .

Методы штрафных функций (МШФ) представляют большой интерес лишь при выполнении следующих требований:

- 1) решения подзадач должны стремиться к решению исходной задачи нелинейного программирования вида

$$\lim x^t \rightarrow x^*, t \rightarrow k < \infty;$$

- 2) сложность минимизации $P(x, R)$ должна быть того же порядка, что и для функции $f(x)$;
- 3) правило пересчета $R^{t+1} = F(R^t)$ должно быть простым.

Рассмотрим широко используемые типы штрафов и различные процедуры учета ограничений при переходе к задачам безусловной минимизации.

6.4.2 Квадратичный штраф

Квадратичный штраф (КШ) используется для учета ограничений-равенств и имеет вид:

$$H = R \cdot [h(x)]^2. \quad (6.39)$$

При минимизации этот штраф препятствует отклонению величины $h(x)$ от нуля (как вправо, так и влево). При увеличении R стационарная точка соответствующей ШФ $P(x, R)$ приближается к точке x^* , т. к. в пределе $h(x^k) = 0$. Поскольку допустимая область S определяется ограничением $h(x) = 0$, то в допустимой области значение штрафа $H = 0$, а вне допустимой области $H > 0$ и тем больше, чем дальше точка x^t выйдет за пределы допустимой области и чем больше R .

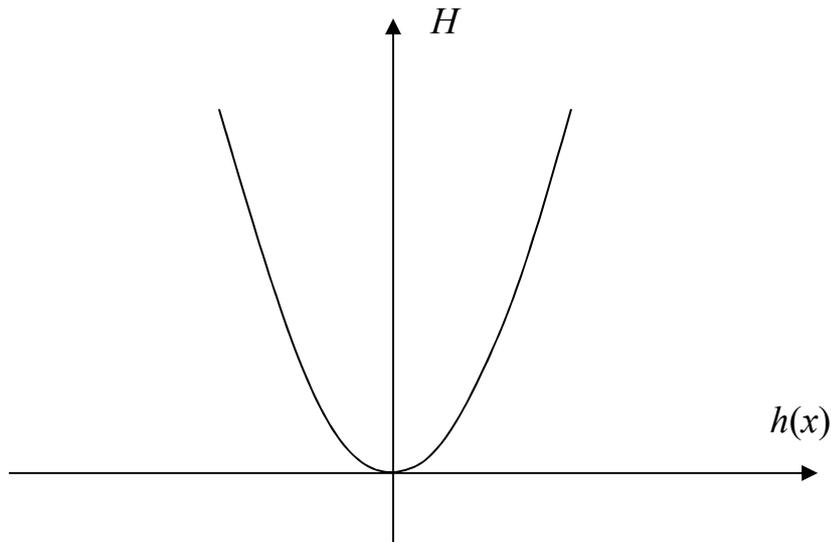


Рис. 6.3 – Квадратичный штраф

Если бы можно было положить $R = \infty$, то точное решение соответствующей подзадачи на безусловный минимум совпало бы с решением исходной задачи. Но при этом линии уровня ШФ становятся слишком вытянутыми – получаем *овражную функцию*. При этом сложность безусловной минимизации резко возрастает.

Предпочтительнее стратегия многоэтапного последовательного приближения к решению:

- 1) процесс начинают с относительно малого R , например $R = 0,01$, и находят точку минимума функции $P(x, 0,01)$;
- 2) увеличивают R , например на порядок ($R = 0,1$), и находят минимум $P(x, 0,1)$, используя ранее полученную точку в качестве начальной;
- 3) параметр R увеличивают до тех пор, пока не окажется, что элементы итерационных последовательностей x^t , $f(x^t)$, $P(x^t, R)$ изменяются от шага к шагу достаточно мало. Однако при этом увеличивается число итераций и время решения задачи. В этом основная трудность МШФ.

6.4.3 Логарифмический штраф

Рассмотрим несколько типов штрафов для ограничений-неравенств. *Логарифмический штраф* (ЛШ) имеет вид:

$$H = -R \cdot \ln[g(x)].$$

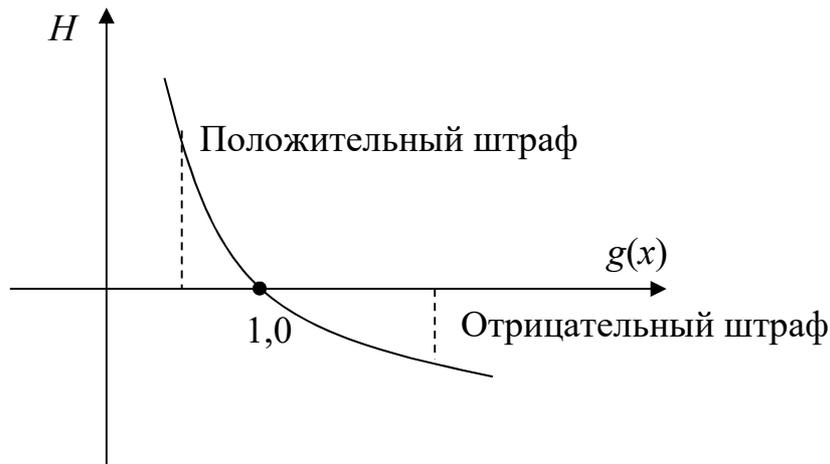


Рис. 6.4 – Логарифмический штраф

Штраф положителен при всех x , таких, что $0 < g(x) < 1$, и отрицателен при $g(x) > 1$. В данном случае вводится как бы искусственная дискриминация точек допустимой области: внутренним точкам отдается предпочтение перед граничными точками. Отрицательных значений штрафа можно было бы избежать, положив $H = 0$ для таких x , что $g(x) > 1$. Однако при этом у градиента ∇P появились бы разрывы.

Логарифмический штраф – это барьерная функция, неопределенная в недопустимых точках (т. е. для таких точек x , в которых $g(x) < 0$). Поэтому в тех случаях, когда приходится иметь дело с недопустимыми точками (например, когда точка $x^0 \notin S$), требуется процедура специального типа, обеспечивающая попадание в допустимую область S .

Итерационный процесс желательно начинать из допустимой области точки x^0 при положительном начальном значении R (например, $R = 10$ или $R = 100$). После решения каждой подзадачи безусловной минимизации параметр R уменьшается и в пределе стремится к нулю.

6.4.4 Штраф типа обратной функции

Штраф типа обратной функции (ШОФ) имеет вид:

$$H = R \frac{1}{g(x)}.$$

Здесь H , в отличие от логарифмического штрафа, не имеет отрицательных значений в допустимой области (рис. 6.5). Данный штраф называют также барьерной функцией.

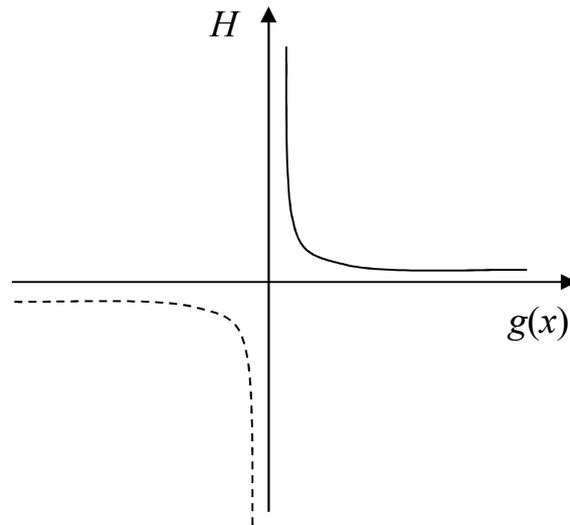


Рис. 6.5 – Штраф – обратная функция

Очевидна необходимость предотвращения появления недопустимых точек, в которых $H < 0$. В точках $x \in S$ вблизи границы значения штрафа положительны и быстро убывают при удалении от границы внутрь допустимой области. На самой границе $P(x, R)$ и ∇P не определены.

Итерации начинаются с точки x^0 – начальной допустимой точки при $R > 0$, причем в процессе вычисления $R \rightarrow 0$.

6.4.5 Штраф типа квадрата срезки

Штраф типа квадрата срезки (ШКС) имеет вид:

$$H = R \cdot \langle g(x) \rangle^2,$$

$$\langle g(x) \rangle = \begin{cases} g(x), & \text{если } g(x) \leq 0, \\ 0, & \text{если } g(x) > 0. \end{cases}$$

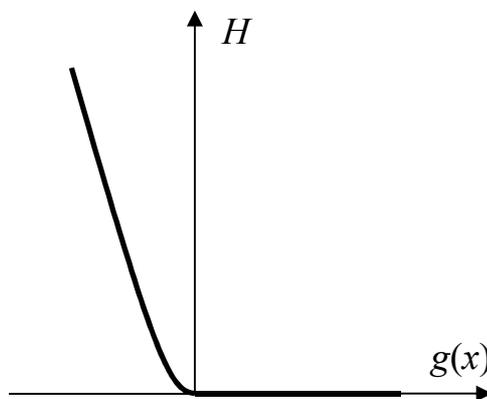


Рис. 6.6 – Штраф – квадрат срезки

Отметим, что H – внешний штраф, и стационарные точки функции $P(x, R)$ могут оказаться недопустимыми. С другой стороны, недопустимые точки не создают в данном случае дополнительных сложностей по сравнению с допустимыми точками. Различие между ними состоит лишь в том, что в допустимых и граничных точках $H = 0$.

Достоинства: функция $P(x, R)$ определена и непрерывна всюду. Вычисления проводятся с положительными R ; после решения очередной задачи безусловной минимизации R увеличивается.

6.4.6 Выбор штрафного параметра

Рассмотренные примеры демонстрируют, с одной стороны, принципиальную возможность использования методов ШФ, а с другой стороны – их общие недостатки, а именно: сходимость метода связана с ярко выраженной нелинейностью («овражностью») ШФ, и здесь традиционные методы безусловной минимизации могут оказаться неэффективными.

При практическом использовании любой из рассмотренных выше штрафных функций требуется выбрать начальное значение R и изменять его после решения каждой подзадачи безусловной минимизации с тем, чтобы обеспечить сходимость последовательности стационарных точек x^i .

Значение R следует выбирать следующим образом: при использовании внешних штрафов при переходе от одной подзадачи к другой значение R должно увеличиваться, а при использовании внутренних (барьерных) штрафов – должно уменьшаться.

При этом необходимо установить, как именно будет изменяться R при переходе от одной подзадачи к другой. Например, для обычного квадратичного штрафа, учитывающего ограничения равенства, целесообразно начинать с $R = 0$, т. е. безусловной минимизации функции, а затем последовательно увеличивать R на некоторое ΔR . Вычислительные эксперименты показывают, что точно найденная стационарная точка $x^i(R)$ перемещается вдоль некоторой гладкой кривой к точке x^* .

С другой стороны, пересчет параметра R , имеющий целью обеспечить сходимость метода, автоматически приводит к ухудшению обусловленности вспомогательных задач (появление овражной структуры штрафной функции). Во многих методах безусловной оптимизации используется матрица Гессе – $\nabla^2 P(x, R)$. Часть собственных значений этой матрицы зависит от величины R .

При этом обусловленность матрицы Гессе $\nabla^2 P(x, R)$ неограниченно ухудшается, когда $R \rightarrow \infty$. Тем самым усложняется процесс решения вспомогательных задач безусловной оптимизации.

6.4.7 Обобщенный алгоритм

Построение эффективных алгоритмов на основе метода штрафных функций не представляет особых трудностей. Приведем простейший алгоритм, в котором используется штрафная функция:

1. Задать значения:

- n – размерность вектора x ;
- m – число ограничений-неравенств;
- p – число ограничений-равенств;
- ε_1 – параметр окончания одномерного поиска;
- ε_2 – параметр окончания процедуры безусловной минимизации;
- ε_3 – параметр окончания работы алгоритма;
- x^0 – начальное приближение для x^* ;
- R^0 – начальный вектор штрафных параметров.

2. Построить $P(x, R) = f(x) + H[R, g(x), h(x)]$.

3. Найти значение x^{t+1} , доставляющее минимум функции $P(x^{t+1}, R^t)$ при фиксированном R^t . В качестве начальной точки использовать x^t , в качестве параметра окончания шага – константу ε_2 .

4. Проверить выполнение условий окончания поиска:

$$\left| P(x^{t+1}, R^t) - P(x^t, R^{t-1}) \right| \leq \varepsilon_3, \quad \|x^{t+1} - x^t\| \leq \varepsilon_3.$$

Да: положить $x^* = x^{t+1}$ и закончить работу алгоритма.

Нет: перейти на шаг 5.

5. Положить $R^{t+1} = R^t + \Delta R^t$ в соответствии с используемым правилом пересчета штрафных параметров и перейти шаг 2.

При наличии ограничений в форме неравенств и равенств используют следующие штрафные функции:

$$P(x, R) = f(x) + R \cdot \sum_{j=1}^m \frac{1}{g_j(x)} + \frac{1}{R} \cdot \sum_{l=1}^p [h_l(x)]^2, \quad R^0 \square 1, \quad R^t \rightarrow 0,$$

$$P(x, R) = f(x) - R \cdot \sum_{j=1}^m \ln[g_j(x)] + \frac{1}{R} \cdot \sum_{l=1}^p [h_l(x)]^2, R^0 \square 1, R^t \rightarrow 0,$$

$$P(x, R) = f(x) + R \cdot \sum_{j=1}^m \langle g_j(x) \rangle^2 + R \cdot \sum_{l=1}^p [h_l(x)]^2, R^0 \square 1, R^t \rightarrow \infty$$

и т. п.

6.5 Методы, основанные на линеаризации

Наиболее изученными в теории оптимизации являются два класса задач: задачи без ограничений и задачи с линейными ограничениями. В большинстве случаев для решения задач с нелинейными ограничениями используются методы, разработанные для упомянутых классов задач оптимизации [3]. В предыдущих разделах применялся первый подход – попытка свести ЗНП к задачам без ограничений.

В данном разделе рассматривается второй подход, основанный на линеаризации, который позволяет свести общую задачу к задаче с линейными ограничениями (или даже линейной ЦФ). Использование линеаризации дает возможность применять методы линейного программирования.

Все рассматриваемые ниже методы основываются на разложении нелинейной функции общего вида $f(x)$ в ряд Тейлора до членов первого порядка в окрестности некоторой точки x^0 :

$$f(x) = f(x^0) + \nabla f^T(x^0) \cdot (x - x^0) + O(\|x - x^0\|)^2,$$

где $O(\|x - x^0\|)^2$ – отбрасываемый член второго порядка малости.

Таким образом, функция $f(x)$ аппроксимируется в точке x^0 линейной функцией:

$$\bar{f}(x, x^0) \cong f(x^0) + \nabla f(x^0) \cdot (x - x^0), \quad (6.40)$$

где x^0 – точка линеаризации. Таким же образом линеаризуются ограничения задачи, если они представлены нелинейными функциями.

6.5.1 Базовый метод линеаризации

Рассмотрим общую задачу нелинейного программирования:

$$\begin{aligned} f(x) &\rightarrow \min; \\ g_i(x) &\geq 0, \quad i = 1, \dots, m, \end{aligned}$$

$$h_l(x) = 0, \quad l = 1, \dots, p,$$

$$a_j \leq x_j \leq b_j.$$

Применим к ней описанный выше подход и сведем ее к ЗЛП.

Алгоритм линейризации. Исходные данные – ЦФ, ограничения, начальная точка $x^0 \in S$, точность решения по аргументу и по функции – абсолютная (ε_x и ε_y) или относительная (δ_x и δ_y).

1. Задать номер итерации $k = 0$. Вычислить $f(x^0)$.

2. Вычислить $g_i(x^k)$, $h_l(x^k)$, $\nabla f(x^k)$, $\nabla g_i(x^k)$, $\nabla h_l(x^k)$. На тот случай, если безусловное оптимальное решение лежит внутри ОДР (т. е. условный минимум совпадает с безусловным минимумом), можно проверить условие $\|\nabla f(x^k)\| \leq \varepsilon_y$. Если оно выполняется, то положить $x^* \approx x^k$ и закончить вычисления.

3. Иначе линейризовать задачу в точке x^k – текущей оценке решения. Использование непосредственной линейризации приводит к следующей задаче:

$$f(x^k) + \nabla f(x^k) \cdot (x - x^k) \rightarrow \min,$$

$$g_i(x^k) + \nabla g_i(x^k) \cdot (x - x^k) \geq 0,$$

$$h_l(x^k) + \nabla h_l(x^k) \cdot (x - x^k) = 0,$$

$$a_j \leq x_j \leq b_j.$$

Эта задача представляет собой ЗЛП.

4. Решить ЗЛП и найти новое приближение x^{k+1} , которое может и не принадлежать допустимой области решений S . Если $x^{k+1} \notin S$, то оптимальное значение линейризованной целевой функции, удовлетворяющее неравенству:

$$\bar{f}(x^{k+1}, x^k) < \bar{f}(x^k, x^k),$$

может не быть точной оценкой истинного значения оптимума, и следует вернуться на шаг 2. Иначе перейти к шагу 5.

5. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $|f(x^{k+1}) - f(x^k)| \leq \varepsilon_y$ (если заданы относительные погрешности, то $\|x^{k+1} - x^k\| / \|x^{k+1}\| \leq \delta_x$, $|f(x^{k+1}) - f(x^k)| / |f(x^{k+1})| \leq \delta_y$), положить $x^* \approx x^{k+1}$ и закончить вычисления. Иначе выполнить присваивание $k = k + 1$ и вернуться на шаг 2.

Для сходимости к экстремуму достаточно, чтобы для последовательности точек $\{x^k\}$, полученных в результате решения последовательности подзадач ЛП, выполнялось следующее условие: значение целевой функции и невязки по ограничениям в точке x^{k+1} должно быть меньше их значений в точке x^k .



.....
 Линеаризацию следует использовать с большой осторожностью, поскольку иногда она дает весьма грубое приближение.

При решении задачи значения $f(x)$, $g(x)$ и $h(x)$ монотонно улучшаются. Однако такая монотонность характерна для задач, функции которых являются «умеренно» нелинейными. В случае функций с ярко выраженной нелинейностью монотонность улучшения нарушается и алгоритм перестает сходиться.

Существуют три способа усовершенствования методов непосредственной линеаризации:

- 1) использование линейного приближения для отыскания направления спуска;
- 2) глобальная аппроксимация нелинейной функции задачи при помощи кусочно-линейчатой функции;
- 3) применение последовательных линеаризаций на каждой итерации для уточнения допустимой области S .

Ниже мы рассмотрим эти три способа.

6.5.2 Алгоритм Франка – Вульфа

Наиболее простым способом линеаризации является замена общей нелинейной задачи на задачу, получаемую при помощи линеаризации всех функций исходной задачи. Таким образом, получается задача линейного программирования, которую можно решать при помощи методов линейного программирования.

Рассмотрим задачу нелинейного программирования с линейными ограничениями:

$$f(x) \rightarrow \min, \quad (6.41)$$

$$Ax \leq b, \quad x \geq 0, \quad (6.42)$$

где $f(x)$ – нелинейная целевая функция.

Допустимая область S является выпуклым многогранником, образованным пересечением полуплоскостей. Так как $f(x)$ нелинейная, то оптимальное решение может не совпадать с вершиной (или угловой точкой) S . Кроме того, если $f(x)$ не выпуклая, то задача нелинейного программирования может иметь несколько локальных минимумов.

Рассмотрим задачу линейного программирования, полученную линеаризацией исходной задачи нелинейного программирования в некоторой допустимой точке $x^0 \in S$:

$$\begin{aligned} \bar{f}(x, x^0) &\rightarrow \min, \\ Ax &\leq b, \quad x \geq 0. \end{aligned}$$

Данная задача линейного программирования имеет оптимальное решение в угловой точке $\bar{x}^* \in S$.

Важен вопрос о соотношении \bar{x}^* и x^* – решением исходной задачи, причем исходная задача может иметь несколько локальных минимумов. Выясним, будет ли $\bar{x}^* \rightarrow x^*$. В силу допустимости точек x^0 и \bar{x}^* имеет место неравенство:

$$\bar{f}(x^0, x^0) \geq \bar{f}(\bar{x}^*, x^0).$$

Это следует из того факта, что \bar{x}^* является минимумом функции $\bar{f}(x, x^0)$. Следовательно, если воспользоваться формулой для \bar{f} , то получается следующее неравенство:

$$\bar{f}(x^0, x^0) = f(x^0) \geq f(x^0) + \nabla f^T(x^0)(\bar{x}^* - x^0) \quad (6.43)$$

или

$$\nabla f^T(x^0)(\bar{x}^* - x^0) \leq 0. \quad (6.44)$$

Очевидно, что вектор $(\bar{x}^* - x^0)$ задает направление спуска.

В задаче оптимизации без ограничений использование направления спуска в качестве направления поиска эффективно лишь при применении специальных правил изменения шага или одномерного поиска. Заметим, что одномерный поиск из точки x^0 в направлении $(\bar{x}^* - x^0)$ приводит к точке \bar{x}^* . Поскольку \bar{x}^* является угловой точкой области S и точка $x^0 \in S$, то все точки на отрезке прямой между ними также допустимы (т. к. S – выпуклая область). Таким образом, одномерный поиск должен производиться на следующем отрезке прямой:

$$x = x^0 + \alpha(\bar{x}^* - x^0), \quad 0 \leq \alpha \leq 1. \quad (6.45)$$

Следовательно, решение задачи линейного программирования, даже не дающее хорошего приближения к экстремуму, позволяет получить важную информацию: определить направление поиска и точки пересечения соответствующего луча с границей допустимой области S .

В результате решения задачи

$$f(x^0 + \alpha(\bar{x}^* - x^0)) \rightarrow \min_{\alpha}, \quad 0 \leq \alpha \leq 1 \quad (6.46)$$

находится точка $x^1 \in S$, такая, что $f(x^1) < f(x^0)$.

Поскольку величина $\nabla f(x^1) \neq 0$, то полученная точка x^1 может служить точкой линеаризации для построения следующей аппроксимации. Решение последовательности задач линейного программирования и одномерный поиск продолжается до тех пор, пока расстояние между точками последовательных оптимумов x^k не станет меньше значения ε .

На изложенной информации основан следующий оптимизационный метод Франка – Вульфа (МФВ).

Алгоритм метода Франка – Вульфа. Исходные данные – ЦФ, ограничения, начальная точка $x^0 \in S$, точность решения по аргументу и по функции – абсолютная (ε_x и ε_y) или относительная (δ_x и δ_y).

1. Задать номер итерации $k = 0$. Вычислить $f(x^0)$.

2. Вычислить $\nabla f(x^k)$. Если $\|\nabla f(x^k)\| \leq \varepsilon_y$, то положить $x^* \approx x^k$ и закончить вычисления.

3. Решить следующую задачу линейного программирования:

$$\begin{aligned} \nabla f(x^k) \cdot y &\rightarrow \min, \\ Ay &\leq b, \quad y \geq 0. \end{aligned}$$

Пусть y^k – оптимальное решение этой задачи.

4. Найти α^k , представляющее собой решение задачи при

$$\Phi(\alpha) = f(x^k + \alpha(y^k - x^k)) \rightarrow \min_{\alpha}, \quad 0 \leq \alpha \leq 1.$$

5. Вычислить $x^{k+1} = x^k + \alpha^k(y^k - x^k)$.

6. Проверить близость к решению. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $|f(x^{k+1}) - f(x^k)| \leq \varepsilon_y$ (если заданы относительные погрешности, то $\|x^{k+1} - x^k\| / \|x^{k+1}\| \leq \delta_x$,

$|f(x^{k+1}) - f(x^k)| / |f(x^{k+1})| \leq \delta_y$), положить $x^* \approx x^{k+1}$ и закончить вычисления.

Иначе выполнить присваивание $k = k + 1$ и вернуться на шаг 2.



.....
 В формулировке подзадачи линейного программирования опущены постоянные члены целевой функции $\bar{f}(x, x^k)$, т. к. они не влияют на решение.

6.5.3 Метод допустимых направлений Зойтендейка

В рассмотренном выше методе линейная аппроксимация используется *не для определения очередного приближения, а только для выявления подходящего направления поиска в каждой точке*. В этом случае точку экстремума вдоль данного направления можно определить непосредственной проверкой значений целевой функции и функций, входящих в ограничение. Аналогичная стратегия применяется в градиентных методах безусловной оптимизации. Здесь мы продолжим исследование этого подхода для задач с нелинейными ограничениями.

В задачах с ограничениями линеаризуются и ограничения, и целевые функции, а направления необходимо выбирать так, чтобы они приводили к допустимым точкам $x \in S$.

Рассмотрим задачу с ограничениями в виде неравенств:

$$\begin{aligned} f(x) &\rightarrow \min, \\ g_i(x) &\geq 0, \quad i = 1, 2, \dots, m. \end{aligned} \quad (6.47)$$

Пусть x^0 – начальная точка, удовлетворяющая ограничениям:

$$g_i(x^0) \geq 0, \quad i = 1, 2, \dots, m.$$

Предположим, что некоторые ограничения являются активными в точке x^0 :

$$I^0 = \{i : g_i(x^0) = 0, i \in I\}, \quad I^0 \subset I = \{1, 2, \dots, m\}.$$

Вектор d определяет допустимое направление для поиска, если d – направление спуска, т. е.

$$\nabla f^T(x) \cdot d < 0, \quad (6.48)$$

и точки луча

$$x(\alpha) = x^0 + \alpha d, \quad \text{где } \alpha \geq 0 \quad (6.49)$$

являются допустимыми на небольшом расстоянии от x^0 .

Неравенство $\nabla f^T(x) \cdot d < 0$ получается следующим образом. Разложим в ряд Тейлора функцию $f(x)$ в точке x^0 :

$$f(x) = f(x^0) + \nabla f^T(x^0)(x - x^0).$$

В направлении спуска в допустимых точках должно выполняться неравенство: $f(x) < f(x^0)$, следовательно, $\nabla f^T(x^0)(x - x^0) < 0$. Вводя обозначение $x - x^0 = d$, получим искомый результат.

Точки $x(\alpha)$ будут допустимыми, если для всех активных ограничений выполняется условие:

$$\tilde{g}_i(x, x^0) = g_i(x^0) + \nabla g_i^T(x^0)(x - x^0) \geq 0. \quad (6.50)$$

Так как по предположению

$$g_i(x^0) = 0, \quad \forall i \in I^0 \text{ и } x - x^0 = d,$$

то выражение (6.50) эквивалентно следующему условию для d :

$$\nabla g_i^T(x^0) \cdot d \geq 0, \quad \forall i \in I^0. \quad (6.51)$$

Основная идея: на каждом шаге итерации определяется допустимое направление, т. е. вектор d и скалярный параметр $\theta > 0$, такие, чтобы выполнялись следующие неравенства:

$$\nabla f^T(x^0) \cdot d \leq -\theta, \quad \nabla g_i^T(x^0) \cdot d \geq 0, \quad i \in I^0, \quad (6.52)$$

а значение θ выбирается по возможности большим.

При реализации на ЭВМ допустимые направления удобно нормировать, вводя границы

$$-1 \leq d_i \leq 1, \quad i = 1, 2, \dots, n. \quad (6.53)$$

Такой способ выбора вектора d обеспечивает разумный компромисс между движением внутрь области допустимых решений без нарушения ограничений и движением по направлению наискорейшего спуска.

После того, как вектор d выбран, очередное приближение может быть определено поиском минимального значения α вдоль прямой

$$x = x^0 + \alpha d^0 \quad (6.54)$$

до тех пор, пока либо ЦФ $f(x)$ не достигнет экстремума, либо какое-то из ограничений не окажется нарушенным.

Обычно для каждого ограничения $g_i(x) \geq 0$ находятся значения $\alpha_i > 0$, при которых

$$g_i(x^0 + \alpha d^0) = 0, \quad (6.55)$$

а затем определяется $\bar{\alpha}$ как наименьшее из α_i

$$\bar{\alpha} = \min_i \{\alpha_i\}.$$

При известном значении $\bar{\alpha}$ можно использовать любую процедуру одномерного поиска для определения α , которое минимизирует функцию $f(x^0 + \alpha d^0)$ на отрезке $[0, \bar{\alpha}]$.

Данный алгоритм называется *методом допустимых направлений Зойтендейка* (МДНЗ) [1, 6, 8].

Алгоритм метода допустимых направлений Зойтендейка. Исходные данные – ЦФ, ограничения, начальная точка $x^0 \in S$, точность решения по аргументу и по функции – абсолютная (ε_x и ε_y) или относительная (δ_x и δ_y), нулевой порог для определения активных ограничений ε_z .

1. Задать номер итерации $k = 0$. Вычислить $f(x^0)$.

2. Вычислить $\nabla f(x^k)$, $g_i(x^k)$, $\nabla g_i(x^k)$. Если $\|\nabla f(x^k)\| \leq \varepsilon_y$, то положить $x^* \approx x^k$ и закончить вычисления.

3. В данной допустимой точке $x^k \in S$ определить множество индексов тех ограничений, которые активны в точке x^k , т. е.

$$I^k = \{i : g_i(x^k) = 0, i = 1, 2, \dots, m\}.$$

Однако если алгоритм реализуется численно, из-за погрешностей вычислений на ЭВМ может наблюдаться не точное равенство активного ограничения нулю в точке x^k , а приближенное. Либо с этой проблемой можно столкнуться при округлении промежуточных результатов расчетов. В этом случае принимают

$$I^k = \{i : |g_i(x^k)| \leq \varepsilon_z, i = 1, 2, \dots, m\},$$

где ε_z – заданная погрешность определения активных ограничений.

4. Решить ЗЛП:

$$\begin{aligned} -\theta &\rightarrow \min, \\ \nabla f^T(x^k) \cdot d &\leq -\theta, \\ \nabla g_i^T(x^k) \cdot d &\geq \theta, \quad i \in I^k, \\ -1 &\leq d_j \leq 1, \quad j = 1, 2, \dots, n. \end{aligned}$$

Пусть d^k и θ^k – полученное решение. Если $\theta^k \leq 0$, то положить $x^* \approx x^k$ и закончить вычисления, т. к. дальнейшее улучшение невозможно.

5. Иначе найти

$$\bar{\alpha} = \min \left\{ \bar{\alpha}_i : g_i(x^k + \alpha d^k) = 0, \alpha \geq 0, i = 1, 2, \dots, m \right\}.$$

Если не существует $\bar{\alpha}_i \geq 0$, положить $\bar{\alpha}_i = \infty$ (в программной реализации можно взять просто большое число, например, $\bar{\alpha}_i = 10^{10}$). При поиске $\bar{\alpha}$ как $\min \left\{ \bar{\alpha}_i : g_i(x^k + \alpha d^k) = 0 \right\}$ может оказаться, что не для всех $i = 1, 2, \dots, m$ будет выполняться равенство $g_i(x^k + \bar{\alpha} d^k) = 0$, значит, мы определяем сразу I^{k+1} – множество индексов, для которых $g_i(x^{k+1}) = 0$.

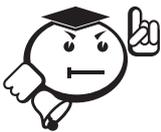
6. Найти α^k , представляющее собой решение задачи при

$$\Phi(\alpha) = f(x^k + \alpha d^k) \rightarrow \min_{\alpha}, 0 \leq \alpha \leq \bar{\alpha}.$$

7. Положить $x^{k+1} = x^k + \alpha^k d^k$.

8. Проверить близость к решению. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $|f(x^{k+1}) - f(x^k)| \leq \varepsilon_y$ (если заданы относительные погрешности, то $\|x^{k+1} - x^k\| / \|x^{k+1}\| \leq \delta_x$, $|f(x^{k+1}) - f(x^k)| / |f(x^{k+1})| \leq \delta_y$), положить $x^* \approx x^{k+1}$ и закончить вычисления.

Иначе выполнить присваивание $k = k + 1$ и вернуться на шаг 2.



.....
 В любой машинной программе поиск граничного значения и одномерный поиск экстремума можно реализовать с помощью известных методов одномерной оптимизации.

В алгоритме Зойтендейка учитываются только активные ограничения в данной допустимой точке, при этом получается зигзагообразный процесс, замедляющий решение, а в некоторых случаях приводящий к «заеданию» – тип ложной сходимости.

К методам, свободным от этих недостатков, можно отнести метод ε -возмущений, а также метод Топкинса и Вейнотта.

Методами допустимых направлений нельзя непосредственно пользоваться для решения задач с нелинейными ограничениями-равенствами $h_k(x) = 0$. В этом случае эти ограничения-равенства необходимо ослабить, допустив ограниченное перемещение вне поверхности, задаваемой ограничениями

$$|h_k(x)| \leq \varepsilon.$$

Однако если ε мало, то одномерный поиск осуществляется небольшими шагами и скорость процесса невелика. С другой стороны, если допускается ограниченное движение вне границ области допустимых решений S , облегчающее выбор допустимого направления, то итеративное решение соответствующих ограничений уравнений нужно проецировать на область S (рис. 6.7).

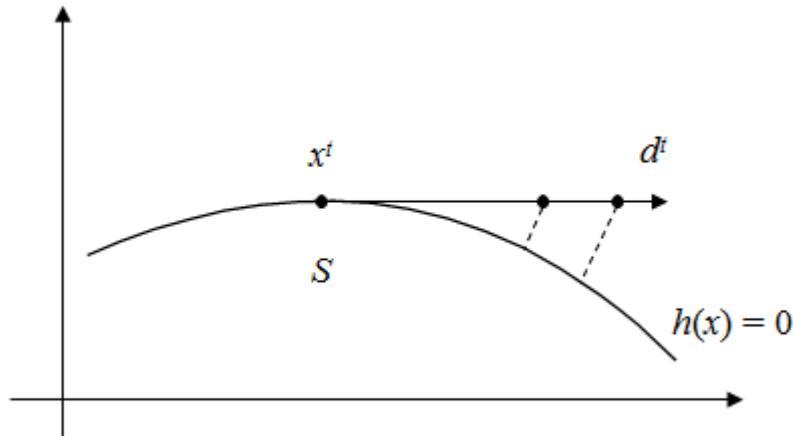


Рис. 6.7 – Проекция на область допустимых решений ЗО

Другим существенным недостатком метода допустимых направлений является необходимость решения подзадач ЛП.

Ниже рассматриваются методы, которые не имеют указанных недостатков.

6.5.4 Метод условного градиента

В *методе условного градиента* (МУГ) рассматривается ЗНП [5]:

$$f(x) \rightarrow \min, \quad (6.56)$$

$$g_i(x) \leq 0, \quad i = 1, 2, \dots, m, \quad (6.57)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n, \quad (6.58)$$

где $f(x)$ – выпуклая функция.

Пусть $x^k \in S$ – очередное приближение исходной задачи НП и $\nabla f(x^k) \neq 0$.

Тогда в окрестности точки x^k ЦФ $f(x)$ представима в виде:

$$\tilde{f}(x, x^k) = f(x^k) + \nabla f^T(x^k) \cdot (x - x^k) + O(\|x - x^k\|)$$

и линейная функция

$$f_k(x) = \nabla f^T(x^k) \cdot (x - x^k)$$

является приближением разности $\tilde{f}(x, x^k) - f(x^k)$ с точностью до величины $O(\|x - x^k\|)$ в некоторой окрестности точки x^k .

Поставим вспомогательную задачу минимизации на множестве S линейной функции $f_k(x)$, т. е.

$$f_k(x) = \nabla f^T(x^k) \cdot (x - x^k) \rightarrow \min \quad (6.59)$$

при тех же ограничениях (6.57), (6.58).

Пусть y^k – решение этой задачи. Следующее приближение x^{k+1} к точке минимума x^* исходной ЦФ $f(x)$ на множестве S найдем по формуле:

$$x^{k+1} = x^k + \alpha_k (y^k - x^k), \quad \alpha_k \in [0, 1]. \quad (6.60)$$

В силу выпуклости S следует, что $x^{k+1} \in S$. Величина α_k из (6.60) может вычисляться различными способами. Например,

$$\alpha_k = \min \{1, \alpha_k^*\},$$

где α_k^* найдено из условия наискорейшего спуска по направлению $d_k = y^k - x^k$:

$$\Phi(\alpha_k^*) = \min_{\alpha} f(x^k + \alpha(y^k - x^k)).$$

Другой способ вычисления значения α_k . В начале выполнения итерации (6.60) полагают $\alpha_k = 1$, после чего проверяют условие:

$$f(x^{k+1}) < f(x^k). \quad (6.61)$$

Если это условие нарушается, то α_k уменьшают в 2 раза (до тех пор, пока неравенство (6.61) не будет выполнено) и переходят к следующей итерации (6.60).

Алгоритм метода условного градиента. Исходные данные – ЦФ, ограничения, начальная точка $x^0 \in S$, точность решения по аргументу и по функции – абсолютная (ε_x и ε_y) или относительная (δ_x и δ_y).

1. Задать номер итерации $k = 0$. Вычислить $f(x^0)$.

2. Вычислить $\nabla f(x^k)$. Если $\|\nabla f(x^k)\| \leq \varepsilon_y$, то положить $x^* \approx x^k$ и закончить вычисления.

3. Решить вспомогательную задачу

$$f_k(x) = \nabla f^T(x^k) \cdot (x - x^k) \rightarrow \min$$

при ограничениях (6.57), (6.58). Пусть y^k – оптимальное решение этой задачи.

4. Найти α^k , представляющее собой решение задачи при

$$\Phi(\alpha) = f(x^k + \alpha(y^k - x^k)) \rightarrow \min_{\alpha}, \quad 0 \leq \alpha \leq 1.$$

5. Вычислить $x^{k+1} = x^k + \alpha^k(y^k - x^k)$.

6. Проверить близость к решению. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $|f(x^{k+1}) - f(x^k)| \leq \varepsilon_y$ (если заданы относительные погрешности, то $\|x^{k+1} - x^k\| / \|x^{k+1}\| \leq \delta_x$, $|f(x^{k+1}) - f(x^k)| / |f(x^{k+1})| \leq \delta_y$), положить $x^* \approx x^{k+1}$ и закончить вычисления.

Иначе выполнить присваивание $k = k + 1$ и вернуться на шаг 2.

Отметим, что в общем случае задача (6.59) является, вообще говоря, задачей нелинейного программирования ($g_i(x)$ – нелинейные функции). Укажем случаи, когда поиск решения y^k не представляет затруднений.

1. Допустимое множество $S = \{x \in R^n \mid a_j \leq x_j \leq b_j, j = 1, 2, \dots, n\}$ является n -мерным параллелепипедом. Тогда

$$y_j^k = \begin{cases} a_j, & \text{если } \partial f(x^k) / \partial x_j > 0, \\ b_j, & \text{если } \partial f(x^k) / \partial x_j < 0, \\ (a_j + b_j) / 2, & \text{если } \partial f(x^k) / \partial x_j = 0. \end{cases} \quad (6.62)$$

Покажем это. Пусть $\nabla f(x^k) = g$, $\nabla f^T(x^k) \cdot x^k = d$. В этом случае имеем:

$$\begin{cases} g^T x - d \rightarrow \min, \\ x_j - x_{n+j} = a_j, \\ x_j + x_{2n+j} = b_j, \\ j = 1, 2, \dots, n. \end{cases}$$

Имеем ЗЛП. Пусть $n = 2$. Найдем начальную угловую точку. Имеем

$$x_1 - x_3 = b_1, \quad x_2 - x_4 = b_2, \quad x_1 + x_5 = a_1, \quad x_2 + x_6 = a_2.$$

Составим симплекс-таблицу:

x_1	x_2	x_3	x_4	x_5	x_6	
1	0	-1	0	0	0	a_1
0	1	0	-1	0	0	a_2
1	0	0	0	1	0	b_1
0	1	0	0	0	1	b_2

Выделим базис:

x_1	x_2	x_3	x_4	x_5	x_6	
1	0	-1	0	0	0	a_1
0	1	0	-1	0	0	a_2
0	0	1	0	1	0	$b_1 - a_1$
0	0	0	1	0	1	$b_2 - a_2$

Таким образом, получим начальное базисное решение $x^0 = (b_1, b_2; 0, 0; b_1 - a_1; b_2 - a_2)^T$. Подставим в ЦФ это значение, получим: $f(x) = g_1 x_3 + g_2 x_4 + p_0$, где $p_0 = g_1 b_1 + g_2 b_2 - d$. Если $g_1, g_2 > 0$, то в точке x^0 достигнут минимум, т. е. $x_1^* = a_1$, $x_2^* = a_2$. Если g_1 или $g_2 < 0$, то решаем задачу дальше:

	x_3	x_4	
x_1	-1	0	a_1
x_2	0	-1	a_2
x_5	1	0	$b_1 - a_1$
x_6	0	1	$b_2 - a_2$
	g_1	g_2	$-p_0$

Если $g_1 < 0$, то разрешающим столбцом будет первый столбец. В этом столбце разрешающая строка третья. После преобразования получим

$$x_1 = a_1 + (b_1 - a_1) = b_1,$$

$$x_2 = b_2.$$

Если $g_2 < 0$, то аналогично получим:

$$x_2 = a_2 + (b_2 - a_2) = b_2,$$

$$x_1 = b_1.$$

2. Допустимое множество S задано линейными ограничениями и условием неотрицательности переменных. Тогда (6.59) – ЗЛП и ее решение можно найти с помощью симплекс-метода. В этом случае МУГ напоминает МФВ.

3. Допустимое множество

$$S = \left\{ x \in R^n \mid \sum (x_j - a_j)^2 \leq R_0^2 \right\}, \quad (6.63)$$

то есть S – шар радиуса R_0 с центром в точке a . Тогда

$$y^k = a - R_0 \frac{\nabla f(x^k)}{\|\nabla f(x^k)\|}. \quad (6.64)$$

Покажем это. Пусть $\nabla f(x^k) = g$, $\nabla f^T(x^k) \cdot x^k = d$. Тогда имеем следующую задачу:

$$\begin{cases} f(x) = g^T x - d \rightarrow \min, \\ \sum (x_j - a_j)^2 \leq R_0^2. \end{cases}$$

Составляем функцию Лагранжа L :

$$L = g^T x - d + \lambda \cdot \left[\sum (x_j - a_j)^2 + b^2 - R_0^2 \right] \rightarrow \min.$$

Здесь слагаемое b^2 добавлено для приведения неравенства к равенству: при $b^2 \geq 0$ имеем $\sum (x_j - a_j)^2 + b^2 = R_0^2$. Ищем минимум:

$$\begin{cases} \frac{\partial L}{\partial x_j} = g_j + 2\lambda(x_j - a_j) = 0, \\ \frac{\partial L}{\partial y} = 2\lambda b = 0 \Rightarrow b = 0, \\ \frac{\partial L}{\partial \lambda} = \sum (x_j - a_j)^2 + y^2 - R_0^2 = 0, \end{cases}$$

$$x_j - a_j = -\frac{1}{2\lambda} g_j, \quad (2\lambda)^2 = \frac{\sum g_j^2}{\sum (x_j - a_j)^2},$$

$$\frac{1}{2\lambda} = \frac{R_0}{\sqrt{\sum g_j^2}}, \quad x_j = a_j - g_j \frac{R_0}{\sqrt{\sum g_j^2}}$$

или

$$y^k = a - \nabla f(x^k) \frac{R_0}{\|\nabla f(x^k)\|}.$$

В этом случае МУГ напоминает метод проекции градиента, который будет рассмотрен далее.

6.6 Метод проекции градиента

Существуют две разновидности *метода проекции градиента* (МПГ) – для решения задач с линейными и нелинейными ограничениями [1, 5–6, 9–10].

6.6.1 Случай линейных ограничений

Рассмотрим ЗО при единственном линейном ограничении в виде равенства

$$f(x) \rightarrow \min, \quad (6.65)$$

$$h(x) = \sum_{j=1}^n a_j x_j = a^T x = b. \quad (6.66)$$

В заданной точке x^k , в которой $\nabla f(x^k) \neq 0$, делается попытка найти направление поиска, которое бы лежало на поверхности ограничения и являлось направлением спуска. Такое направление можно получить геометрически, ортогонально проецируя вектор, противоположный $\nabla f(x^k)$, на поверхность ограничения (см. рис. 6.8).

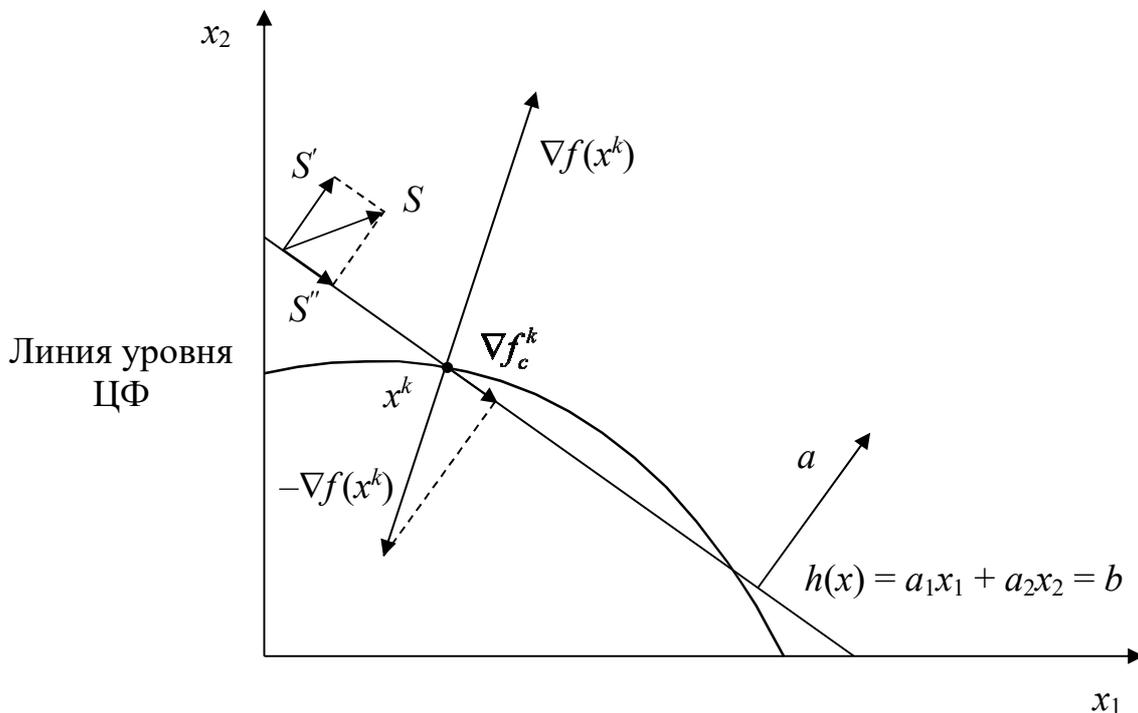


Рис. 6.8 – Метод проекции градиента

Здесь ∇f_c^k – проекция антиградиента на поверхность ограничений, которая приводит в допустимые точки. Действительно, $\forall \alpha \geq 0$ точки, заданные соотношением

$$x = x^k - \alpha \cdot \nabla f_c^k, \quad (6.67)$$

удовлетворяют линейному ограничению:

$$a^T x = a^T x^k - \alpha a^T \nabla f_c^k,$$

где $a^T x^k = b$, $a^T \nabla f_c^k = 0$.

Если $a \perp \nabla f_c$, то $a^T x = b$.

Это направление задает спуск, т. к. угол между векторами ∇f_c^k и $-\nabla f_c^k$ больше 90° . Процесс ортогонального проектирования состоит в разложении вектора на две ортогональные компоненты: одна параллельна поверхности, заданной ограничением, вторая перпендикулярна к ней. Параллельная компонента является искомой проекцией градиента.

Пусть вектор a – нормаль к поверхности ограничения. Отметим, что из выражения $a^T S = 0$ следует допустимость направления, задаваемого вектором S (S параллельно поверхности).

Таким образом, все векторы, перпендикулярные к поверхности ограничения, должны быть параллельны к a . Следовательно, для любого вектора S его компонента S' , перпендикулярная к поверхности ограничения, равняется значению a , умноженному на константу.

Обозначим через S'' компоненту S , параллельную поверхности ограничения. Тогда S'' удовлетворяет соотношению

$$a^T S'' = 0. \quad (6.68)$$

Таким образом, любой вектор можно представить в виде векторной суммы:

$$S = S' + S'', \quad (6.69)$$

где $S' = \lambda a$, а S'' удовлетворяет уравнению $a^T S'' = 0$.

Найдем λ . Рассмотрим скалярное произведение $a^T S$. В силу (6.69) и (6.68), имеем

$$a^T S = a^T \lambda a + a^T S'' = \lambda a^T a, \quad (6.70)$$

откуда

$$\lambda = (a^T a)^{-1} a^T S. \quad (6.71)$$

Из (6.69) найдем $S'' = S - S' = S - \lambda a$. Подставим сюда (6.71), получим

$$S'' = S - a (a^T a)^{-1} a^T S = \left[I - a (a^T a)^{-1} a^T \right] S, \quad (6.72)$$

где I – единичная матрица, порядок которой согласован с S . Матрица $P = I - a(a^T a)^{-1} a^T$ – проекционная матрица. Она проектирует вектор S на плоскость, задаваемую ограничением $h(x)$.

Отметим, что P является симметрической и положительно полуопределенной. Симметричность P очевидна. Для доказательства положительной полуопределенности рассмотрим произведение $y^T P y$ для произвольного $y \neq 0$. Тогда

$$y^T P y = y^T \left[I - a(a^T a)^{-1} a^T \right] y = y^T y - \frac{(y^T a)(a^T y)}{a^T a} = \frac{(y^T y)(a^T a) - (y^T a)^2}{a^T a}.$$

Используя неравенство Шварца $(y^T a)^2 \leq (y^T y) \cdot (a^T a)$, убеждаемся, что числитель неотрицателен.



Пример 6.3

Вычислить проекцию вектора $S = (0; 1)^T$ на поверхность, задаваемую ограничениями $h(x) = 2x_1 + x_2 = 3$ в точке $(3/4; 3/2)$.

Уравнение прямой, проходящей через данную точку M_0 , перпендикулярно к данной плоскости, задается уравнением

$$\frac{x - x_0}{A} = \frac{y - y_0}{B} = \frac{z - z_0}{C},$$

где $Ax + By + Cz + D = 0$ – заданная плоскость, $M_0(x_0, y_0, z_0)$ – заданная точка. В нашем случае точка $M_0 = (3/4, 3/2)$. Плоскость – это прямая, т. е. $A = 2$, $B = 1$, $C = 0$, $D = -3$. Таким образом, получим:

$$\frac{x_1 - 3/4}{2} = \frac{x_2 - 3/2}{1},$$

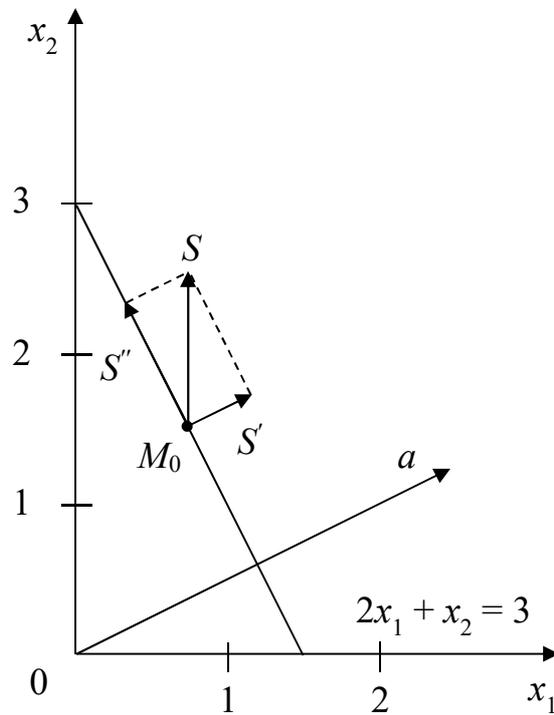
откуда следует уравнение прямой: $2x_2 = x_1 + 9/4$. Перенесем параллельно эту прямую в начало координат, получим $2x_2 = x_1$. В качестве нормали возьмем вектор с координатами $\nabla h(x) = (2; 1)^T$.

Проекционная матрица задается выражением:

$$P = I - a \cdot (a^T a)^{-1} a^T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 2 \\ 1 \end{pmatrix} \cdot \frac{1}{5} \cdot (2 \quad 1) = \begin{pmatrix} 0,2 & -0,4 \\ -0,4 & 0,8 \end{pmatrix}.$$

Проекция S'' имеет вид:

$$S'' = P \cdot S = \begin{pmatrix} 0,2 & -0,4 \\ -0,4 & 0,8 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -0,4 \\ 0,8 \end{pmatrix}.$$

Рис. 6.9 – Проекция вектора S на поверхность

Проекция $S' = S - S'' = \begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} -0,4 \\ 0,8 \end{pmatrix} = \begin{pmatrix} 0,4 \\ 0,2 \end{pmatrix}$. Очевидно, что S' и S''

ортогональны, т. е.

$$(S')^T S'' = (0,4 \quad 0,2) \cdot \begin{pmatrix} -0,4 \\ 0,8 \end{pmatrix} = 0,$$

и S'' перпендикулярен a :

$$a^T S'' = (2 \quad 1) \cdot \begin{pmatrix} -0,4 \\ 0,8 \end{pmatrix} = 0.$$

.....
Способ построения проекционной матрицы P легко распространяется на случай многих ограничений:

$$h_k(x) = a_k^T x = b_k, \quad k = 1, 2, \dots, m, \quad S' = A^T \lambda,$$

$$S'' = P \cdot S = \left[I - A^T (AA^T)^{-1} A \right] S,$$

$$\lambda = (AA^T)^{-1} AS - \text{вектор},$$

где A – матрица со строками a_k^T , т. е.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

где $a_{i1}, a_{i2}, \dots, a_{in}$ – коэффициенты i -го ограничения.

Пусть $S'' = -P \cdot \nabla f(x^k)$ – направление спуска. Если $S'' = 0$, то точка x^k удовлетворяет необходимым условиям Лагранжа. Вектор множителей Лагранжа задается выражением:

$$\lambda = (AA^T)^{-1} A \cdot \nabla f(x). \quad (6.73)$$

Первое утверждение следует из того, что матрица P симметрическая и положительно полуопределена. Действительно, имеем

$$x^{k+1} = x^k - \alpha P \cdot \nabla f(x^k), \quad \alpha \geq 0;$$

$$f(x^{k+1}) = f(x^k - \alpha P \cdot \nabla f(x^k)) = f(x^k) - \alpha \nabla f^T(x^{k+1}) \cdot P \cdot \nabla f(x^k).$$

Так как $\nabla f^T(x^{k+1}) \cdot P \cdot \nabla f(x^k) \geq 0$, $\alpha \geq 0$, то $f^{k+1} \leq f^k$, что и требовалось доказать.

Имеем $S'' = -P \cdot \nabla f$, т. е. S'' – это проекция вектора ∇f на поверхность ограничений. Поэтому, если $S'' = 0$, значит, градиент ∇f перпендикулярен поверхности ограничений.

Рассмотрим второе свойство. Так как $S'' = -P \cdot \nabla f$, то произведение $\nabla f^T S'' = -\nabla f^T \cdot P \cdot \nabla f \leq 0$. Если $S'' = -P \cdot \nabla f = 0$, то $\nabla f \perp S''$, т. е. ∇f перпендикулярен поверхности ограничений. Тогда из формулы

$$S = S' + S'' = A^T \lambda + S'' = 0$$

следует

$$\nabla f = A^T \lambda. \quad (6.74)$$

Так как строками матрицы A являются векторы коэффициентов в линейных ограничениях, то (6.74) представляет собой другую форму записи необходимого условия Лагранжа:

$$\nabla f - \sum_{k=1}^m \lambda_k a_k = 0, \quad (6.75)$$

где a_k – k -й вектор ограничений. Отсюда мы видим, что λ_k – это множители Лагранжа, т. е. множители функции

$$L = f(x) - \sum_{k=1}^m h_k(x) \lambda_k.$$

Необходимое условие экстремума:

$$\frac{\partial L}{\partial x_i} = 0 \Rightarrow \nabla f - \sum \lambda_k \nabla h_k = 0,$$

$$\nabla h_k = a_k.$$

Нетрудно получить выражение для λ :

$$S = S' + S'',$$

$$AS = AA^T \lambda + \underbrace{AS''}_{=0} \Rightarrow \lambda = (AA^T)^{-1} AS = (AA^T)^{-1} A \cdot \nabla f,$$

т. е.

$$\lambda = (AA^T)^{-1} A \cdot \nabla f, \quad (6.76)$$

где $(AA^T)^{-1}$ существует только в том случае, если значения a_k линейно независимы. Если имеются линейно зависимые ограничения, их следует исключить из рассмотрения.

Алгоритм метода проекции градиента для случая линейных ограничений. Исходные данные – ЦФ, ограничения, начальная точка $x^0 \in S$, точность решения по аргументу и по функции – абсолютная (ε_x и ε_y) или относительная (δ_x и δ_y). Ограничения необходимо привести к виду $a_i x \geq b_i$, $i = 1, 2, \dots, m$.

1. Задать номер итерации $k = 0$. Вычислить $f(x^0)$.

2. Вычислить $\nabla f(x^k)$, $g_i(x^k)$. Если $\|\nabla f(x^k)\| \leq \varepsilon_y$, то положить $x^* \approx x^k$ и закончить вычисления.

3. Составить матрицу A ограничений из строк, соответствующих активным ограничениям.

4. Вычислить проекционный оператор $P = I - A^T (AA^T)^{-1} A$ в предположении, что векторы a_k линейно независимы. Вычислить вектор проекции $S^k = -P \cdot \nabla f(x^k)$.

5. Если $\|S^k\| > \varepsilon_x$, то перейти на шаг 6. Иначе вычислить множители Лагранжа λ по формуле

$$\lambda = (AA^T)^{-1} \cdot A \cdot \nabla f(x^k).$$

Если все $\lambda_i \geq 0$, то положить $x^* \approx x^k$ и закончить вычисления – оптимальное решение найдено. В противном случае ограничение с наибольшим по модулю множителем Лагранжа исключить из множества активных ограничений и перейти на шаг 4.

6. Определить максимальную длину шага:

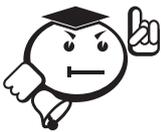
$$\alpha_{\max} = \min \left\{ \begin{array}{l} \infty, \text{ если } a_k S^k = 0 \text{ или } \frac{b_k - a_k x^k}{a_k S^k} \leq 0 \\ \frac{b_k - a_k x^k}{a_k S^k}, \text{ если } a_k S^k \neq 0 \text{ и } \frac{b_k - a_k x^k}{a_k S^k} > 0 \end{array} \right\}, \quad k = 1, 2, \dots, m.$$

7. Решить задачу одномерного поиска параметра α^k :

$$\Phi(\alpha) = f(x^k + \alpha S^k) \rightarrow \min_{\alpha}, \quad 0 \leq \alpha \leq \alpha_{\max}.$$

8. Положить $x^{k+1} = x^k + \alpha^k S^k$.

9. Проверить близость к решению. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $|f(x^{k+1}) - f(x^k)| \leq \varepsilon_y$ (если заданы относительные погрешности, то $\|x^{k+1} - x^k\| / \|x^{k+1}\| \leq \delta_x$, $|f(x^{k+1}) - f(x^k)| / |f(x^{k+1})| \leq \delta_y$), положить $x^* \approx x^{k+1}$ и закончить вычисления. Иначе выполнить присваивание $k = k + 1$ и вернуться на шаг 2.



Мы рассмотрели алгоритм, в котором используются линейные равенства. Но его можно легко распространить на неравенства, используя либо дополнительные переменные, либо активные ограничения, что предпочтительнее, т. к. второй способ позволяет уменьшить размерность.

При использовании второго способа с помощью получаемой по формуле (6.73) оценки множителей Лагранжа осуществляется поочередное исключение ограничений из множества активных ограничений. Модификация выглядит следующим образом. В заданной точке x^k для определения активного множества проверяются ограничения в виде неравенств: $a_j^T x \geq b_j$, $j = 1, 2, \dots, m$.

6.6.2 Случай нелинейных ограничений

Один из подходов к решению ЗНП состоит в такой модификации градиентных методов безусловной минимизации, чтобы в процессе построения последовательных приближений к точке минимума учитывались ограничения на допустимое множество S [7].

Рассмотрим ЗНП:

$$\begin{aligned} f(x) &\rightarrow \min, \\ g_i(x) &\leq 0, \quad i = 1, 2, \dots, m, \\ x &\in S \subset R^n, \end{aligned}$$

где S – выпуклое замкнутое множество, $f(x)$ – выпуклая дифференцируемая на множестве S функция.

На каждой итерации метода предусмотрена процедура возврата очередного приближения градиентного спуска

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k)$$

на допустимое множество S , если $x^{k+1} \notin S$. Такой возврат проводится посредством проектирования x^{k+1} на S , т. е. замены x^{k+1} на ближайшую точку множества S .



.....

Пусть заданы замкнутое множество $S \subset R^n$ и точка $z \in R^n$. Точка $z_S = P_S(z)$ называется **проекцией** точки z на множество S , если $\rho(z_S, z) = \min_{x \in S} \rho(x, z)$, где $\rho(x, y) = \|x - y\|$ – расстояние между точкой x и точкой y в пространстве R^n .

.....

Ясно, что для точки $z \in S$ проекция $P_S(z)$ совпадает с точкой z .

Таким образом, в методе проекции градиента последовательные приближения x^k к точке минимума x^* целевой функции $f(x)$ на множестве S вычисляются по формулам

$$x^{k+1} = P_S[x^k - \alpha^k \nabla f(x^k)], \quad k = 0, 1, \dots, \quad x^0 \in S. \quad (6.77)$$

В зависимости от способа вычисления α^k , различают несколько вариантов МПГ (ниже приводятся два наиболее распространенных):

1) α^k находятся как и в методе наискорейшего спуска безусловной минимизации, т. е.

$$\Phi(\alpha) = f(x^k - \alpha \nabla f(x^k)) \rightarrow \min_{\alpha};$$

2) в предположении, что $\nabla f(x)$ удовлетворяет на множестве S условию Липшица, т. е.

$$\|\nabla f(x') - \nabla f(x'')\| \leq \Lambda \|x' - x''\| \quad (6.78)$$

для $\forall x', x'' \in S$, полагают $\alpha^k \equiv \alpha$, $k = 0, 1, \dots$, где α – произвольное число из интервала $(0, 2/\Lambda]$. Если известна минимальная константа Липшица Λ из выражения (6.78), то выбирается $\alpha^k = 1/\Lambda$.

Алгоритм метода проекции градиента для случая нелинейных ограничений. Исходные данные – ЦФ, ограничения, начальная точка $x^0 \in S$, точность решения по аргументу и по функции – абсолютная (ε_x и ε_y) или относительная (δ_x и δ_y).

1. Задать номер итерации $k = 0$. Вычислить $f(x^0)$.

2. Вычислить $\nabla f(x^k)$. Если $\|\nabla f(x^k)\| \leq \varepsilon_y$, то положить $x^* \approx x^k$ и закончить вычисления.

3. Найти α^k , представляющее собой решение задачи при

$$\Phi(\alpha) = f(x^k - \alpha \nabla f(x^k)) \rightarrow \min_{\alpha},$$

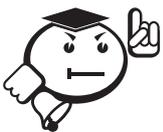
либо из соотношения (6.78), т. е. $\alpha^k \in (0, 2/\Lambda]$, например, $\alpha^k = 1/\Lambda$.

4. Вычислить следующее приближение по формуле (6.77):

$$x^{k+1} = P_S [x^k - \alpha^k \nabla f(x^k)].$$

5. Проверить близость к решению. Если $\|x^{k+1} - x^k\| \leq \varepsilon_x$ и $|f(x^{k+1}) - f(x^k)| \leq \varepsilon_y$ (если заданы относительные погрешности, то $\|x^{k+1} - x^k\| / \|x^{k+1}\| \leq \delta_x$, $|f(x^{k+1}) - f(x^k)| / |f(x^{k+1})| \leq \delta_y$), положить $x^* \approx x^{k+1}$ и закончить вычисления.

Иначе выполнить присваивание $k = k + 1$ и вернуться на шаг 2.



.....
 Определение проекции $P_S(z)$ для точки $z \notin S$ является самостоятельной ЗНП:

$$\tilde{f}(x) = \|x - z\|^2 = \sum_{j=1}^n (x_j - z_j)^2 \rightarrow \min, \quad (6.79)$$

решение которой может вызвать затруднения. Решить ее можно, например, с помощью метода множителей Лагранжа.

.....

Рассмотрим задачу проектирования, которая решается в явном виде.



..... **Пример 6.4**

Найдем проекцию $P_S(z)$ точки $z \in R^n$ на множество S , где

$$S = \left\{ x \in R^n \mid \|x\|^2 = \sum_{j=1}^n x_j^2 \leq R_0^2 \right\} -$$

шар радиуса R_0 в пространстве R^n с центром в начале координат.

Решение. Запишем задачу поиска проекции как ЗНП:

$$\begin{cases} f(x) = \sum_{j=1}^n (x_j - z_j)^2 \rightarrow \min, \\ \sum_{j=1}^n x_j^2 \leq R_0^2. \end{cases} \quad (6.80)$$

Рассмотрим возможные частные случаи:

1) пусть $z \in S$, т. е. $\sum_{j=1}^n z_j^2 \leq R_0^2$. Тогда $P_S(z) = z$;

2) пусть $z \notin S$, т. е. $\sum_{j=1}^n z_j^2 = R^2 > R_0^2$. Запишем неравенство ЗНП (6.80) в

виде равенства, добавив новую переменную $x_{n+1} = y^2 \geq 0$. В результате получим задачу на условный экстремум с ограничением-равенством:

$$f(x) = \sum_{j=1}^n (x_j - z_j)^2 \rightarrow \min,$$

$$h(x, y) = \sum_{j=1}^n x_j^2 + y^2 - R_0^2 = 0.$$

Составляем функцию Лагранжа и минимизируем ее:

$$L(x, y, \lambda) = \sum_{j=1}^n (x_j - z_j)^2 + \lambda \left(\sum_{j=1}^n x_j^2 + y^2 - R_0^2 \right) \Rightarrow$$

$$\begin{cases} \frac{\partial L}{\partial x_j} = 2(x_j - z_j) + 2\lambda x_j = 0, \\ \frac{\partial L}{\partial y} = 2\lambda y = 0, \\ \frac{\partial L}{\partial \lambda} = \sum x_j^2 + y^2 - R_0^2 = 0. \end{cases}$$

Решив эту систему $n+2$ уравнений с учетом $\sum_{j=1}^n z_j^2 = R^2$, находим

$x_j = z_j R_0/R$, $j=1, \dots, n$, $1 + \lambda = R_0/R$, $y = 0$, т. е.

$$P_s(z) = \begin{cases} z, & \text{если } \sum_{j=1}^n z_j^2 \leq R_0^2, \\ \frac{z}{\sqrt{\sum z_j^2}} R_0, & \text{если } \sum_{j=1}^n z_j^2 > R_0^2. \end{cases}$$

В тех случаях, когда ОДР задачи не выпукла либо выражение для $P_s(z)$ невозможно выразить аналитически (например, если ОДР будет являться не шаром, а эллипсоидом), использование МПГ вызывает затруднения.



Контрольные вопросы по главе 6

1. Назовите достоинства и недостатки прямых методов решения ЗНП.
2. В чем суть метода замены переменных при решении ЗНП? Приведите пример.
3. В чем заключается основная идея метода множителей Лагранжа?
4. Сформулируйте НУ и ДУ первого порядка оптимальности ЗНП.
5. Сформулируйте общий принцип методов штрафов.
6. В чем сущность метода линеаризации для ЗНП общего вида?
7. На основе какого подхода разработан алгоритм Франка – Вульфа?
8. В чем заключается основная идея метода допустимых направлений Зойтендейка?
9. В чем заключается идея метода условного градиента?
10. В чем суть проекционного метода решения ЗНП?

Заключение

Несмотря на то что методы оптимизации как самостоятельное научное направление сложились еще в середине 1960-х гг., их развитие продолжается по сей день, поэтому в одном издании невозможно рассмотреть все существующие классы задач и методы их решения.

В данном учебнике рассмотрены лишь некоторые теоретические разделы методов оптимизации. В частности, не были рассмотрены материалы:

- теории двойственности (в линейном и нелинейном программировании);
- квадратичного программирования;
- сепарабельного программирования;
- вариационного исчисления;
- теории оптимального управления и т. д.

Практикум по методам оптимизации (решение оптимизационных задач) вынесен в отдельное методическое пособие. В нем приведены примеры решения задач всеми изложенными в учебнике методами.

Оптимизационные задачи встречаются повсюду – в программировании, экономике, транспорте, в целом – практически во всех отраслях человеческой деятельности. Специалист должен уметь распознавать, когда стоящая перед ним задача относится к классу оптимизационных, и выбирать адекватные методы ее решения. Коллектив авторов надеется, что данное учебное пособие поможет Вам в этом.

Литература

1. Реклейтис Г. Оптимизация в технике : в 2 кн. / Г. Реклейтис, А. Рейвиндран, К. Рэгсдел. – М. : Мир, 1986. – Кн. 1. – 349 с.; Кн. 2. – 320 с.
2. Мину М. Математическое программирование. Теория и алгоритмы / М. Мину. – М. : Наука, 1990. – 488 с.
3. Рубан А. И. Оптимизация систем : учеб. пособие / А. И. Рубан. – Томск : Изд-во ТГУ, 1984. – 198 с.
4. Дэннис Дж. Численные методы безусловной оптимизации и решения нелинейных уравнений / Дж. Дэннис, Р. Шнабель. – М. : Мир, 1988. – 440 с.
5. Карманов В. Г. Математическое программирование : учеб. пособие / В. Г. Карманов. – М. : ФИЗМАТЛИТ, 2004. – 264 с.
6. Сборник задач по математике для вузов / Э. А. Вуколов и др. ; под ред. А. В. Ефимова, А. С. Поспелова. – М. : ФИЗМАТЛИТ, 2002. – Ч. 3. – 576 с.
7. Жермен-Лакур П. Математика и САПР : в 2 кн. / П. Жермен-Лакур, П. Л. Жорж, Ф. Пистр, П. Безье. – М. : Мир, 1989. – Кн. 1. – 204 с.; Кн. 2. – 264 с.
8. Химмельблау Д. Прикладное нелинейное программирование / Д. Химмельблау. – М. : Мир, 1991. – 534 с.
9. Введение в методы оптимизации / М. Аоки. – М. : Наука, 1988. – 344 с.
10. Полак Э. Численные методы оптимизации. Единый подход / Э. Полак. – М. : Мир, 1994. – 376 с.
11. Уайлд Д. Дж. Методы поиска оптимума / Д. Дж. Уайлд. – М. : Наука, 1997. – 268 с.
12. Габасов Р. Методы оптимизации / Р. Габасов, Ф. М. Кириллова. – Минск : Изд-во БГУ, 1988. – 352 с.
13. Бахвалов Н. С. Численные методы / Н. С. Бахвалов. – М. : БИНОМ, 2008. – 636 с.
14. Гилл Ф. и др. Практическая оптимизация / Ф. Гилл и др. – М. : Мир, 1992. – 509 с.

15. Коршунов Ю. М. Математические основы кибернетики : учеб. пособие для втузов / Ю. М. Коршунов. – М. : Энергоатомиздат, 1987. – 496 с.
16. Боглаев Ю. П. Вычислительная математика и программирование : учеб. пособие для студентов втузов / Ю. П. Боглаев. – М. : Высш. шк., 1990. – 544 с.
17. Рубан А. И. Методы оптимизации : учеб. пособие / А. И. Рубан. – 2-е изд., испр. и доп. – Красноярск : КГТУ, 2004. – 528 с.

Список условных обозначений и сокращений

\in – знак принадлежности ($a \in A$ – элемент a принадлежит множеству A ,
 $x \in [a, b]$ – точка x принадлежит отрезку $[a, b]$)

\notin – не принадлежит

\subset – знак включения, подмножества ($A \subset B$ – множество A является подмножеством B)

$()$ – (открытый) интервал ((a, b) – интервал, которому принадлежат все значения от a до b , но не включая a и b)

$[]$ – закрытый интервал или отрезок ($[a, b]$ – отрезок, которому принадлежат все значения от a до b включительно)

\Rightarrow – знак следования ($a \Rightarrow b$ – из a следует b)

\rightarrow – стремится ($a \rightarrow b$ – a стремится к b)

$\{ \}$ – знак множества ($\{a, b\}$ – множество, состоящее из элементов a и b)

\emptyset – пустое множество

\forall – квантор общности ($\forall x$ – для всех/любого/каждого x)

\exists – квантор существования ($\exists x$ – найдется/существует x)

$| |$ – модуль или определитель ($|a|$ – модуль скалярной величины a , $|A|$ – определитель матрицы A)

$\| \|$ – норма матрицы или вектора

T – оператор транспонирования матрицы или вектора

$+\infty$ – бесконечно большое положительное число

$-\infty$ – бесконечно большое отрицательное число

Δ – приращение (Δx – приращение аргумента x)

Σ – знак суммы

Λ – константа Липшица

$A()$ – декартовы координаты точки ($A(x, y)$ – точка A с декартовыми координатами (x, y))

const – постоянная величина, константа

E – единичная матрица

extr – экстремум

\bar{f} – значение функции в критической или стационарной точке

f^* – экстремальное значение функции f

$f'(x) = df/dx$ – производная функции скалярного аргумента $f(x)$

$f''(x) = d^2f/dx^2$ – вторая производная функции скалярного аргумента $f(x)$

$f^{(n)}(x) = d^n f/dx^n$ – производная функции скалярного аргумента $f(x)$

порядка n

$f'_x = \partial f/\partial x$ – частная производная функции векторного аргумента ($f'_z(x, y, z)$ – частная производная функции трех аргументов $f(x, y, z)$ по аргументу z)

$f''_{xy} = \partial^2 f/\partial x\partial y$ – вторая частная производная функции векторного аргумента ($f''_{xz}(x, y, z)$ – вторая частная производная функции трех аргументов $f(x, y, z)$ по аргументам x и z)

$\text{grad } f(x) = \nabla f(x)$ – вектор-градиент функции

$H_f(x) = \nabla^2 f(x)$ – матрица Гессе (матрица вторых производных функции)

\inf – точная нижняя граница множества

L – функция Лагранжа

\max – максимум

\min – минимум

$Q[a, b]$ – множество унимодальных функций, заданных на отрезке $[a, b]$

R^n – линейное пространство размерности n

S – область допустимых решений (ОДР)

\sup – точная верхняя граница множества

\bar{x} – критическая или стационарная точка

x^* – экстремальное значение аргумента x

Z – множество целых чисел

ВМ – венгерский метод

ГМ – графический метод решения ЗЛП

ГУ – граничные условия

Д-задача – двойственная задача

ДУ – достаточные условия

ЗКП – задача квадратичного программирования

ЗЛП – задача линейного программирования
ЗНП – задача нелинейного программирования
ЗоН – задача о назначении
ЗЦП – задача целочисленного программирования
ЗО – задача оптимизации
КМ – квазиньютоновские методы
КП – квадратичное программирование
КФ – квадратичная форма
ЛП – линейное программирование
МБФГШ – метод Бroyдена – Флетчера – Гольдфарба – Шенно
МГ – метод Гомори
МД – метод дихотомии
МДНЗ – метод допустимых направлений Зойтендейка
МДФП – метод Дэвидона – Флетчера – Пауэлла
МЗС – метод золотого сечения
МЗП – метод замены переменных
МК – метод Коши
МЛ – метод линеаризации
ММ – метод Марквардта
ММК – метод Миля – Кентрелла
ММЛ – метод множителей Лагранжа
ММН – модифицированный метод Ньютона
МН – метод Ньютона
МНС – метод наименьшей стоимости
МО – методы оптимизации
МП – метод Пауэлла (в контексте безусловной оптимизации)
МП – метод потенциалов (в контексте условной оптимизации)
МПП – метод проекции градиента
МППР – метод Полака – Рибьера
МСГ – метод сопряженных градиентов
МСЗУ – метод северо-западного угла
МСН – метод сопряженных направлений
МСТ – метод средней точки
МУГ – метод условного градиента
МФ – метод Фибоначчи (в контексте безусловной оптимизации)
МФ – метод Фогеля (в контексте условной оптимизации)

МФВ – метод Франка – Вульфа
МФР – метод Флетчера – Ривза
МХД – метод Хука – Дживса
МШФ – методы штрафных функций
НП – нелинейное программирование
НУ – необходимые условия
ОГМ – обобщенный градиентный метод
ОДР – область допустимых решений
П-задача – прямая задача
ПСМ – последовательный симплекс-метод
СМ – симплекс-метод
СП – сепарабельное программирование
СТ – симплекс-таблица
ТЗ – транспортная задача
ТМ – транспортная модель
ТТ – транспортная таблица
ТАУ – теория автоматического управления
ТОУ – теория оптимального управления
ЦП – целочисленное программирование
ЦФ – целевая функция
ШФ – штрафная функция