ЮЖНЫЙ
ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ

Л.А. Гладков
Н.В. Гладкова

Методы оптимизации

Часть 2

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего профессионального образования
«Южный федеральный университет»

Л.А. Гладков, Н.В. Гладкова

Методы оптимизации

КОНСПЕКТ ЛЕКЦИЙ

ЧАСТЬ 2

Таганрог 2013

УДК: 519.8(075)

Рецензенты:

зав. кафедрой компьютерных образовательных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики, доктор технических наук, профессор **Лисицина Л.С.**;

кандидат технических наук, доцент кафедры САиТ ЮФУ **Лебедев В.Б.**

Гладков Л.А., Гладкова Н.В. Методы оптимизации: Конспект лекций. Ч. 2. – Таганрог: Изд-во ЮФУ, 2013. – 105 с.

Рассмотрены основные определения классической теории оптимизации, перечислены основные типы задач оптимизации. Приведены методы и алгоритмы решения задач оптимизации с ограничениями и без ограничений. Описаны методы решения задач нелинейного программирования. Приведено описание различных классов задач нелинейного программирования и методов их решения. Отдельное внимание уделено методам прямого поиска решения задач оптимизации, таким как метод пассивного поиска, метод дихотомии, метод «золотого сечения», метод Фибоначчи. Дано большое количество примеров решения различных задач оптимизации. Конспект лекций предназначен для студентов и магистрантов вузов, обучающихся по направлениям «Информатика и вычислительная техника» и «Информационные системы». Также может быть полезным для специалистов, занятых решением различных задач оптимизации и проектирования, поддержки и принятия решений, внедрения новых информационных технологий в науке, технике, образовании и экономике.

Ил. 29. Библиогр.: 9 назв.

1. КЛАССИЧЕСКАЯ ТЕОРИЯ ОПТИМИЗАЦИИ

1.1. Введение

В классической теории оптимизации для нахождения точек максимума и минимума (экстремальных точек) функций в условиях как отсутствия, так и наличия ограничений на переменные широко используется аппарат дифференциального исчисления. Получаемые при этом методы не всегда оказываются удобными при их численной реализации. Однако соответствующие теоретические результаты лежат в основе большинства алгоритмов решения задач нелинейного программирования (см. разд. 3).

В этом разделе изложены необходимые и достаточные условия существования экстремумов функций при отсутствии ограничений на переменные задачи, методы *Якоби* и *Лагранжа* для решения задач с ограничениями на переменные в форме равенств, а также условия *Куна–Таккера* для задач с ограничениями в виде неравенств.

1.2. Экстремальные задачи без ограничений

Экстремальная точка функции $f(X)$ определяет либо ее максимальное, либо минимальное значение. С математической точки зрения точка $X_0 = (x_1, ..., x_j, ..., x_n)$ является точкой максимума функции $f(X)$, если неравенство

$$f(X_0 + h) \le f(X_0)$$

выполняется для всех $h = (h_1, ..., h_j, ..., h_n)$ таких, что $|h_j|$ достаточно малы при всех j. Другими словами, точка X_0 является точкой максимума, если значения функции f в окрестности точки X_0 не превышают $f(X_0)$. Аналогично, точка X_0 является точкой минимума функции $f(X)$, если для определенного выше вектора h имеет место неравенство

$$f(X_0 + h) \ge f(X_0)\,.$$

На рис. 1.1 показаны точки максимума и минимума функции одной переменной $f(x)$ на интервале $[a, b]$. Точки x_1, x_2, x_3, x_4 и x_6 составляют множество экстремальных точек функции $f(x)$. Здесь точки x_1, x_3 и x_6 являются точками максимума, а точки x_2 и x_4 – точками минимума функции $f(x)$. Поскольку

$$f(x_6) = \max\{f(x_1), f(x_3), f(x_6)\},$$

значение $f(x_6)$ называется глобальным или абсолютным максимумом, а значения $f(x_1)$ и $f(x_3)$ – **локальными** или **относительными** максимумами. Подобным образом значение $f(x_4)$ является локальным, а $f(x_2)$ – глобальным минимумом функции $f(x)$.

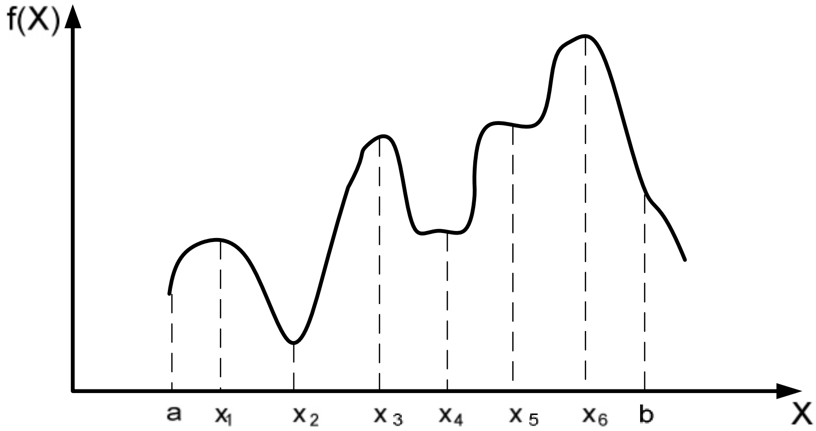


Рис. 1.1

Заметим, что хотя точка x_1 является точкой максимума функции $f(x)$ (рис. 1.1), она отличается от остальных локальных максимумов $f(x)$ тем, что по крайней мере в одной точке ее окрестности значение функции $f(x)$ совпадает с $f(x_1)$. Точка x_1 по этой причине называется **нестрогим (слабым) максимумом** функции $f(x)$, в отличие, например, от точки x_3, которая является **строгим максимумом** $f(x_1)$. Нестрогий максимум, следовательно, подразумевает наличие (бесконечного количества) различных точек, которым соответствует одно и то же максимальное значение функции. Аналогичные результаты имеют место в точке x_4, где функция $f(x)$ имеет нестрогий минимум. В общем случае X_0 является точкой нестрогого максимума функции $f(x)$, если $f(X_0+h) \le f(X_0)$, и точкой ее строгого максимума, если $f(X_0+h) < f(X_0)$, где h – вектор, определенный выше.

На рис. 1.1 легко заметить, что первая производная функции f (тангенс угла наклона касательной к графику функции) равна 0 во всех ее экстремальных точках. Однако это условие выполняется и в точках **перегиба** и **седловых** точках функции f таких, как точка x_5. Если точка, в которой угол наклона касательной к графику функции (градиент функции) равен нулю, не является в то же время точкой экстремума (максимума или минимума), то она автоматически должна быть точкой перегиба или седловой точкой.

1.2.1. Необходимые и достаточные условия существования экстремума

В этом подразделе излагаются необходимые и достаточные условия существования экстремумов функции n переменных $f(X)$. При этом предполагается, что первые и вторые частные производные функции $f(X)$ непрерывны в каждой точке X.

Теорема 1. *Необходимым условием того, что точка* X_0 *является экстремальной точкой функции* $f(X)$, *является равенство*

$$\nabla f(X_0) = 0.$$

Доказательство. Из теоремы Тейлора следует, что при $0 < \theta < 1$ имеет место разложение функции $f(X)$

$$f(X_0 + h) - f(X_0) = \nabla f(X_0)h + \frac{1}{2}h^T Hh|_{X_0+\theta h},$$

где h – вектор, определенный выше.

Для достаточно малых значений $|h_j|$остаточный член $\frac{1}{2}h^T Hh|$ является величиной порядка h_j^2. Следовательно,

$$f(X_0 + h) - f(X_0) = \nabla f(X_0)h + O(h_j^2) \approx \nabla f(X_0)h\ .$$

Пусть X_0 – точка минимума функции $f(X)$. Докажем от противного, что градиент $\nabla f(X_0)$ функции $f(X)$ в точке минимума X_0 равен нулю. Пусть это условие не выполняется; тогда для некоторого j должно выполняться условие

$$\frac{\partial f(X_0)}{\partial x_j} < 0 \text{ или } \frac{\partial f(X_0)}{\partial x_j} > 0.$$

Знак h_j всегда можно выбрать таким образом, чтобы

$$h_j \frac{\partial f(X_0)}{\partial x_j} < 0.$$

Полагая остальные h_j равными нулю, из разложения Тейлора получаем неравенство

$$f(X_0 + h) < f(X_0)\ .$$

Этот результат противоречит предположению, что X_0 – точка минимума. Следовательно, величина $\nabla f(X_0)$ должна равняться нулю. Доказательство для точки максимума проводится аналогично.

Так как необходимое условие выполняется также в точках перегиба и седловых точках, точки, удовлетворяющие уравнению $\nabla f(X_0) = 0$, называют стационарными. Следующая теорема устанавливает достаточные условия того, что стационарная точка X_0 является экстремальной.

Теорема 2. *Для того чтобы стационарная точка* X_0 *была экстремальной, достаточно, чтобы матрица Гессе* H *в точке* X_0 *была*:

1) *положительно определенной (тогда* X_0 *– точка минимума);*

2) *отрицательно определенной (тогда* X_0 *– точка максимума).*

Доказательство. В силу теоремы Тейлора при $0 < \theta < 1$ имеем

$$f(X_0+h) - f(X_0) = \nabla f(X_0)h + \frac{1}{2}h^T Hh|_{X_0+\theta h},$$

Поскольку X_0 – стационарная точка, в силу теоремы 1 $\nabla f(X_0)=0$. Таким образом,

$$f(X_0+h) - f(X_0) = \frac{1}{2}h^T Hh|_{X_0+\theta h}.$$

Если X_0 – точка минимума, то

$$f(X_0+h) > f(X_0)$$

для всех ненулевых векторов h. Следовательно, в точке минимума X_0 должно выполняться неравенство

$$\frac{1}{2}h^T Hh|_{X_0+\theta h} > 0.$$

Непрерывность вторых частных производных функции $f(X)$ гарантирует, что величина $\frac{1}{2}h^T Hh$ имеет один и тот же знак как в точке X_0, так и $X_0 + \theta h$. Так как $h^T Hh|_{X_0}$ представляет собой квадратичную форму, ее значение (и, следовательно, $h^T Hh|_{X_0+\theta h}$) положительно тогда и только тогда, когда $H|_{X_0}$ – положительно определенная матрица. Это означает, что положительная определенность матрицы Гессе в стационарной точке X_0 является достаточным условием существования в этой точке минимума. Путем аналогичных рассуждений доказывается, что стационарная точка является точкой максимума, если матрица Гессе в этой точке отрицательно определена.

Пример 1.1

Рассмотрим функцию

$$f(x_1, x_2, x_3) = x_1 + 2x_3 + x_2 x_3 - x_1^2 - x_2^2 - x_3^2.$$

Необходимое условие экстремума $\nabla f(X_0)=0$ здесь принимает следующий вид:

$$\frac{\partial f}{\partial x_1} = 1 - 2x_1 = 0,$$
$$\frac{\partial f}{\partial x_2} = x_3 - 2x_2 = 0,$$
$$\frac{\partial f}{\partial x_3} = 2 + x_2 - 2x_3 = 0.$$

Решением этой системы уравнений является точка $\mathbf{X}_0$ = (1/2, 2/3, 4/3). Для проверки выполнения условия достаточности вычислим

$$H|_{X_0} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_2} & \frac{\partial^2 f}{\partial x_3^2} \end{bmatrix} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -2 & 1 \\ 0 & 1 & -2 \end{bmatrix}.$$

Угловые миноры матрицы $H|_{X_0}$ равны –2, 4 и –6 соответственно. В этом случае $H|_{X_0}$ является отрицательно определенной матрицей, откуда следует, что точка X_0 = (1/2, 2/3, 4/3) является точкой максимума.

В общем случае, когда матрица $H|_{X_0}$ является неопределенной, точка X_0 должна быть седловой. Если же матрица $H|_{X_0}$ оказывается полуопределенной, то соответствующая точка X_0 может как быть, так и не быть экстремальной. При этом формулировка достаточного условия существования экстремума значительно усложняется, ибо для этого необходимо учитывать члены более высоких порядков в разложении Тейлора. (Иллюстрацией вышесказанного в случае функции одной переменной служит теорема 2.) В некоторых случаях такие сложные процедуры не являются необходимыми, поскольку диагонализация матрицы H позволяет сделать вывод о наличии или отсутствии экстремума. Проиллюстрируем это следующим примером.

Пример 1.2

Рассмотрим функцию

$$f(x_1, x_2) = 8x_1 x_2 + 3x_2^2,$$

необходимые условия экстремума для которой принимают вид

$$\nabla f(x_1, x_2) = (8x_2, 8x_1 + 6x_2) = (0, 0).$$

Отсюда следует, что стационарной является точка X_0 = (0, 0). При этом матрица Гессе в точке X_0 имеет вид

$$H = \begin{bmatrix} 0 & 8 \\ 8 & 6 \end{bmatrix}$$

и не дает информации о наличии или отсутствии экстремума в рассматриваемой точке. Используя один из методов диагонализации матриц, получаем преобразованную матрицу Гессе

$$H_t = \begin{bmatrix} -\dfrac{64}{6} & 0 \\ 0 & 6 \end{bmatrix},$$

проверка угловых миноров которой свидетельствует, что матрица H_t (и, следовательно, H) является неопределенной. Если матрица H приведена к диагональному виду путем преобразования подобия (а это здесь подразумевается), то нет необходимости проверять угловые миноры матрицы H_t, так как в этом случае диагональные элементы матрицы H_t являются собственными значениями матрицы H. Данная матрица будет неопределенной, если диагональные элементы матрицы H_t имеют разные знаки. Следовательно, X_0 – седловая точка.

Применим достаточные условия, полученные в теореме 2, к функции одной переменной. Пусть y_0 – стационарная точка функции f, тогда:

1) неравенство $f''(y_0) < 0$ является достаточным условием существования максимума в точке y_0;

2) неравенство $f''(y_0) < 0$ является достаточным условием существования минимума в точке y_0.

Если же для функции одной переменной $f''(y_0) < 0$, то необходимо исследовать производные высших порядков в соответствии со следующей теоремой.

Теорема 3. *Если в стационарной точке y_0 функции $f(y)$ первые (n – 1) ее производных равны нулю и* $f^{(n)}(y_0) \neq 0$. *то в точке $y = y_0$ функция $f(y)$ имеет*:

1) *точку перегиба, если n – нечетное;*

2) *экстремальную точку, если n – четное. Эта экстремальная точка является точкой максимума при* $f^{(n)}(y_0) < 0$ *и точкой минимума при* $f^{(n)}(y_0) > 0$.

Пример 1.3

Рассмотрим функции $f(y) = y^4$ и $g(y) = y^3$. Для функции $f(y) = y^4$ имеем

$$f'(y) = 4y^3 = 0,$$

откуда получаем стационарную точку $y_0 = 0$. Далее находим

$$f'(0) = f''(0) = f^{(3)}(0) = 0.$$

Так как $f^{(4)}(0) = 24 > 0$, то $y_0 = 0$ является точкой минимума (рис. 1.2). Для функции $g(y) = y^3$ имеем

$$g'(y) = 3y^2 = 0.$$

Следовательно, точка $y_0 = 0$ является стационарной точкой. Поскольку $g^{(n)}(0)$ не обращается в нуль при $n = 3$, точка $y_0 = 0$ является точкой перегиба.

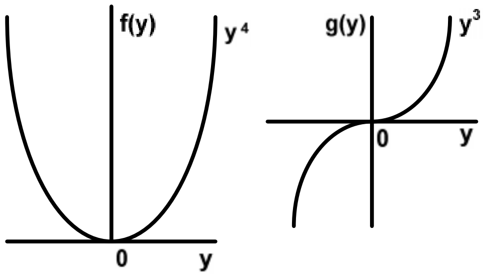


Рис. 1.2

1.2.2. Метод Ньютона–Рафсона

В общем случае использование необходимого условия экстремума $\nabla f(X_0) = 0$ для нахождения стационарных точек функции $f(X)$ может быть сопряжено с трудностями, возникающими при численном решении соответствующей системы уравнений. Метод Ньютона–Рафсона предлагает итерационную процедуру решения системы нелинейных уравнений. Несмотря на то, что данный метод рассматривается в этом разделе именно в указанном контексте, на самом деле он относится к числу градиентных методов численного поиска экстремумов функций при отсутствии ограничений (см. подразд. 1.1.2).

Рассмотрим систему уравнений

$$f_i(X) = 0, i = 1, 2, \ldots, m.$$

Пусть X^k – некоторая фиксированная точка. Используя разложение Тейлора, имеем

$$f_i(X) \approx f_i(X^k) + \nabla f_i(X^k)(X - X^k), i = 1, 2, \ldots, m..$$

Следовательно, исходная система уравнений приближенно представима в виде

$$f_i(X^k) + \nabla f_i(X^k)(X - X^k) = 0, i = 1, 2, \ldots, m.$$

Эти уравнения можно записать в матричной форме:

$$A_k + B_k(X - X^k) = 0.$$

Предположим, что векторы $f_i(X)$ независимы, тогда матрица B_k является невырожденной. В результате из предыдущего уравнения получим

$$X = X^k - B_k^{-1} A_k .$$

Идея метода Ньютона–Рафсона состоит в следующем. На первом шаге выбирается начальная точка X^0. С помощью полученного уравнения по известной точке X^k можно вычислить координаты новой точки X^{k+1}. Процесс вычислений завершается в точке X^m, которая считается приближенным решением исходной системы, если $X^m \approx X^{m-1}$.

Геометрическая интерпретация данного метода для функции одной переменной $f(x)$ приведена на рис. 1.3. Связь между точками x^k и x^{k+1} в этом случае выражается формулой

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

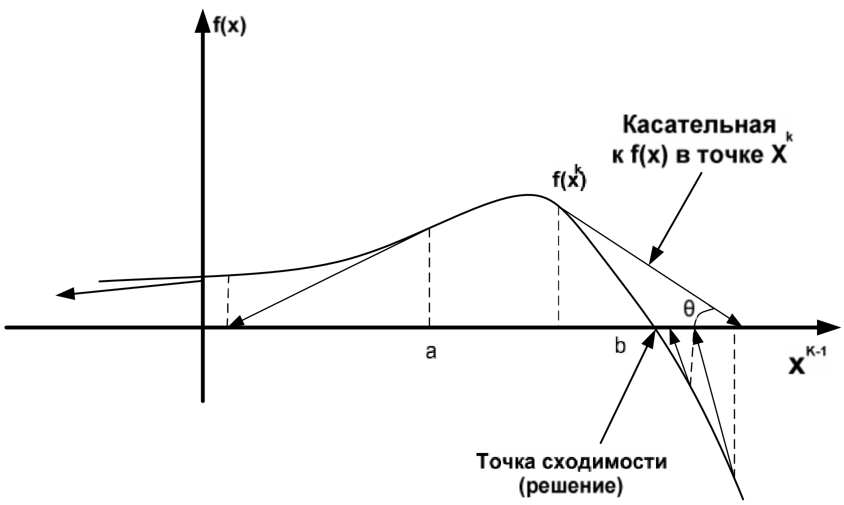


Рис. 1.3

На рис. 1.3 видно, что положение точки x^{k+1} определяется углом θ наклона касательной к графику функции $f(x)$ в точке x^k, где $tg\theta = f'(x^k)$.

Одним из недостатков изложенного метода является то, что для функций общего вида не всегда гарантируется его сходимость. На рис. 1.3 видно, что при выборе в качестве x^0 точки a итерационный процесс расходится. Простых рецептов для выбора "хорошего" начального приближения не существует.

1.3. Задачи на экстремум при наличии ограничений

В настоящем подразделе рассматриваются задачи поиска экстремумов непрерывных функций при наличии ограничений на переменные. В подразд. 1.3.1 рассматривается случай, когда дополнительные ограничения имеют вид равенств, а в подразд. 1.3.2 – вид неравенств.

1.3.1. Ограничения в виде равенств

В данном подразделе рассматриваются два метода решения оптимизационных задач при наличии ограничений в виде равенств: **метод Якоби** и **метод Лагранжа**. Метод Лагранжа можно логически получить из метода Якоби. Эта связь позволяет дать интересную экономическую интерпретацию метода множителей Лагранжа.

МЕТОД ПРИВЕДЕННОГО ГРАДИЕНТА (МЕТОД ЯКОБИ)

Рассмотрим задачу.

Минимизировать $z = f(X)$

при ограничениях

$$g(X) = 0,$$

где

$$X = (x_1, x_2, \ldots, x_n),$$
$$g = (g_1, g_2, \ldots, g_n).$$

Функции $f(X)$ и $g_i(X)$, $i = 1, 2, \ldots, m$, предполагаются дважды непрерывно дифференцируемыми.

Идея использования приведенного градиента заключается в том, чтобы найти замкнутое аналитическое выражение для первых частных производных функции $f(X)$ во всех точках, удовлетворяющих ограничениям $g(X) = 0$. Соответствующие стационарные точки определяются из условия равенства нулю указанных частных производных. Затем можно использовать достаточные условия, сформулированные в подразд. 1.2, для классификации найденных стационарных точек.

Для пояснения изложенной идеи рассмотрим функцию $f(x_1, x_2)$, график которой представлен на рис. 1.4. Предположим, что эту функцию необходимо минимизировать при ограничении

$$g(x_1, x_2) = x_2 - b = 0,$$

где b – некоторая константа. На рис. 1.4 видно, что кривая, которая проходит через точки A, B и C, состоит из значений функции $f(x_1, x_2)$, для которых заданное ограничение выполнено. В соответствии с рассматриваемым методом определяются компоненты приведенного градиента функции $f(x_1, x_2)$ в каждой точке кривой ABC. Точка B, в которой приведенная производная обращается в нуль, является стационарной для рассматриваемой задачи с ограничением.

Теперь рассмотрим общую математическую формулировку метода. Из теоремы Тейлора следует, что для точек $X+\Delta X$ из окрестности точки X имеем

$$f(X+\Delta X)-f(X)=\nabla f(X)\Delta X+O(\Delta x_j^2),\ \ g(X+\Delta X)-g(X)=\nabla g(X)\Delta X+O(\Delta x_j^2).$$

При $\Delta x_j \to 0$ эти уравнения принимают вид

$$\partial f(X)=\nabla f(X)\partial X,\ \ \partial g(X)=\nabla g(X)\partial X.$$

Поскольку $g(X) = 0$, $\partial g(X) = 0$ в допустимой области. Отсюда следует, что

$$\partial f(X)-\nabla f(X)\partial X=0,\ \ \nabla g(X)\partial X=0.$$

Как видим, задача сводится к решению $m + 1$ уравнений с $n + 1$ неизвестными, которыми являются $\partial f(X)$ и ∂X. Неизвестную величину $\partial f(X)$ можно определить, как только будет найден вектор ∂X. Это означает, что, по существу, имеется m уравнений с n неизвестными.

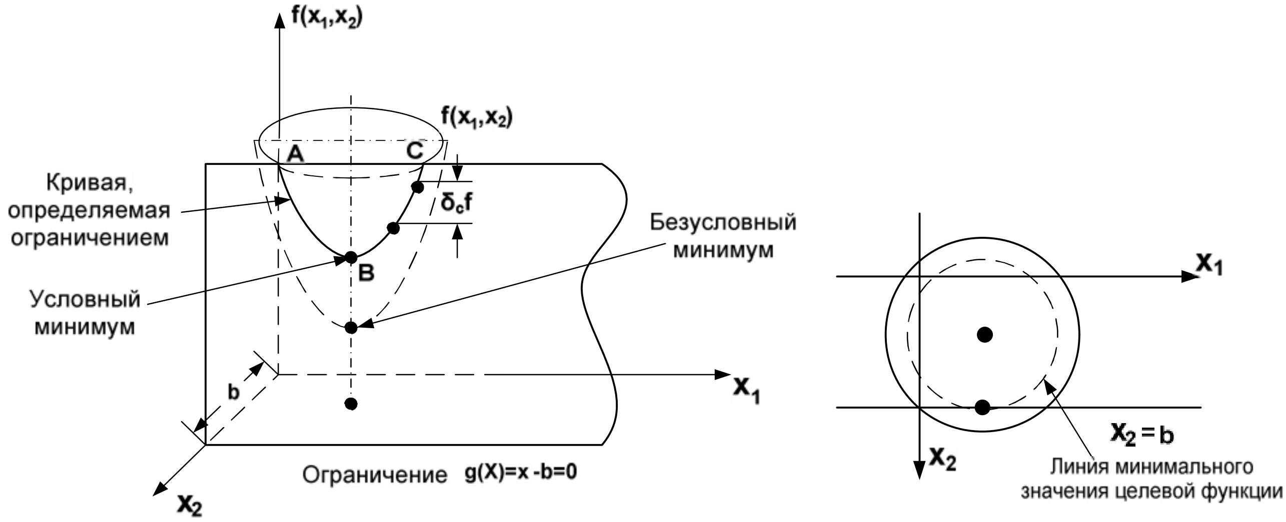


Рис. 1.4

При $m > n$ по меньшей мере $m - n$ уравнений системы являются избыточными. После устранения избыточности количество независимых уравнений в системе становится равным m ($\le n$). Если $m = n$, решением является $\partial X=0$. При этом точка X не имеет допустимой окрестности, и, следовательно, пространство решений задачи состоит из единственной точки. Такая ситуация является тривиальной. Ситуацию, когда $m < n$, рассмотрим подробно.

Пусть

$$X=(Y,Z),$$

где

$$Y=(y_1,y_2,\ldots,y_m) \text{ и } Z=(z_1,z_2,\ldots,z_{n-m})$$

называются *зависимыми* и *независимыми* переменными соответственно.

Переписывая градиенты функций f и g в новых обозначениях, получим

$$\nabla f(Y,Z) = (\nabla_Y f, \nabla_Z f), \;\; \nabla g(Y,Z) = (\nabla_Y g, \nabla_Z g).$$

Введем в рассмотрение матрицы

$$J = \nabla_Y g = \begin{bmatrix} \nabla_Y g_1 \\ \vdots \\ \nabla_Y g_m \end{bmatrix}, \; C = \nabla_Z g = \begin{bmatrix} \nabla_Z g_1 \\ \vdots \\ \nabla_Z g_m \end{bmatrix}.$$

Матрица $J_{m\times m}$ называется **матрицей Якоби,** а $C_{m\times(n-m)}$ – **матрицей управления.** Матрица Якоби J предполагается невырожденной. Это всегда можно обеспечить, поскольку рассматриваемые m уравнений являются независимыми по определению. Поэтому компоненты вектора Y можно выбрать среди компонентов вектора X таким образом, что матрица J окажется невырожденной.

Исходную систему уравнений с неизвестными $\partial f(X)$ и ∂X можно переписать в следующем виде:

$$\partial f(Y,Z) = \nabla_Y f \partial Y + \nabla_Z f \partial Z, \;\; J\partial Y = -C\partial Z.$$

Так как матрица J невырожденная, существует обратная матрица J^{-1}. Следовательно,

$$\partial Y = -J^{-1} C \partial Z .$$

Подставляя это выражение ∂Y в уравнение для $\partial f(Y,Z)$, можно выразить ∂f через ∂Z в следующем виде:

$$\partial f(Y,Z) = (\nabla_Z f - \nabla_Y f J^{-1} C)\partial Z.$$

Из этого уравнения получаем формулу для производных функции f по вектору независимых переменных Z

$$\nabla_c f = \frac{\partial_c f(Y,Z)}{\partial_c Z} = \nabla_Z f - \nabla_Y f J^{-1} C,$$

где $\nabla_c f$ представляет вектор **приведенного градиента** функции f по Z. Следовательно, вектор $\nabla_c f(Y,Z)$ должен обращаться в нуль в стационарных точках.

Достаточные условия экстремума в стационарной точке аналогичны изложенным в подразд. 1.2. В этом случае элементы матрицы Гессе будут соответствовать компонентам вектора независимых переменных Z. Между тем элементы матрицы Гессе должны быть *приведенными* вторыми производными. Чтобы показать, как они вычисляются, положим

$$\nabla_c f = \nabla_Z f - WC .$$

Отсюда следует, что i-й строкой приведенной матрицы Гессе является вектор $\partial\nabla_c f/\partial z_i$. Заметим, что *W* – функция от *Y*, а *Y*, в свою очередь, – функция от *Z*. Следовательно, при вычислении частной производной $\nabla_c f$ по z_i следует применять правило дифференцирования сложной функции, а именно

$$\frac{\partial w_j}{\partial z_i} = \frac{\partial w_j}{\partial y_j}\frac{\partial y_j}{\partial z_i}.$$

Пример 1.4

Рассмотрим следующую задачу:

$$f(X) = x_1^2 + 3x_2^2 + 5x_1x_3^2,$$
$$g_1(X) = x_1x_3 + 2x_2 + x_2^2 + 11 = 0,$$
$$g_2(X) = x_1^2 + 2x_1x_2 + x_3^2 + 14 = 0.$$

Известна допустимая точка X^0 = (1, 2, 3). Требуется оценить приращение функции $f(=\partial_c f)$ в допустимой окрестности точки X^0. Пусть $Y=(x_1,x_3)$ и $Z = x_2$. Тогда имеем

$$\nabla_Y f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_3}\right) = (2x_1 + 5x_3^2, 10x_1x_3),\ \ \nabla_Z f = \frac{\partial f}{\partial x_2} = 6x_2,$$

$$J = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_3} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_3} \end{bmatrix} = \begin{bmatrix} x_3 & x_1 \\ 2x_1 + 2x_2 & 2x_3 \end{bmatrix},\ \ C = \begin{bmatrix} \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_2 + 2 \\ 2x_1 \end{bmatrix}.$$

Оценка приращения $\partial_c f$ окрестности допустимой точки X^0 = (1, 2, 3), вызванного малым изменением ∂x_2 = 0,01, получается следующим образом:

$$J^{-1}C = \begin{bmatrix} 3 & 1 \\ 6 & 6 \end{bmatrix}^{-1} \begin{bmatrix} 6 \\ 2 \end{bmatrix} = \begin{bmatrix} \frac{6}{12} & -\frac{1}{12} \\ -\frac{6}{12} & \frac{3}{12} \end{bmatrix} \begin{bmatrix} 6 \\ 2 \end{bmatrix} \approx \begin{bmatrix} 2,83 \\ -2,50 \end{bmatrix}.$$

Следовательно, приращение функции *f* с учетом ограничений равно

$$\partial_c f = (\nabla_Z f - \nabla_Y f J^{-1}C)\partial Z = \left(6\times 2 - (47,30)\begin{bmatrix} 2.83 \\ -2.50 \end{bmatrix}\right)\partial x_2 \approx -46\partial x_2 = -0,46.$$

Если указана величина изменения ∂x_2 *независимой* переменной x_2, то допустимые значения ∂x_1 и ∂x_3 зависимых переменных x_1 и x_3 определяются по формуле

$$\partial Y = -J^{-1}C\partial Z.$$

При ∂x_2 = 0,01 получаем

$$\begin{bmatrix} \partial x_1 \\ \partial x_3 \end{bmatrix} = J^{-1}C\partial x_2 = \begin{bmatrix} -0,0283 \\ 0,0250 \end{bmatrix}.$$

Чтобы проверить точность полученной выше оценки для $\partial_c f$, можно вычислить значения функции *f* в точках X^0 и $X^0 + \partial X$. Получаем

$$X^0 + \partial X = (1 - 0{,}0283; 2 + 0{,}01; 3 + 0{,}025) = (0{,}9717; 2{,}01; 3{,}025).$$

Отсюда следует, что

$$f(X^0) = 58\, f(X^0 + \partial X) = 57{,}523 \text{ или } \partial_c f = f(X^0 + \partial X) - f(X^0) = -0{,}477.$$

Полученный результат свидетельствует об уменьшении значения функции *f* по сравнению с вычисленным выше в соответствии с формулой для $\partial_c f$. Это различие между двумя полученными результатами (–0,477 и –0,46) является следствием линейной аппроксимации в окрестности точки X_0. Поэтому приведенная выше формула дает хорошие результаты лишь тогда, когда отклонения от точки X_0 малы.

Пример 1.5

Данный пример иллюстрирует процедуру использования метода приведенного градиента. Рассмотрим задачу.

Минимизировать $f(X) = x_1^2 + x_2^2 + x_3^2$

при ограничениях

$$g_1(X) = x_1 + x_2 + 3x_3 - 2 = 0,$$
$$g_2(X) = 5x_1 + 2x_2 + x_3 - 5 = 0.$$

Определяем экстремальные точки целевой функции при наличии ограничений следующим образом. Пусть $Y = (x_1, x_2)$ и $Z = x_3$. Тогда

$$\nabla_Y f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right) = (2x_1, 2x_2),\ \nabla_Z f = \frac{\partial f}{\partial x_3} = 2x_3,\ J = \begin{bmatrix} 1 & 1 \\ 5 & 2 \end{bmatrix},\ J^{-1} = \begin{bmatrix} -\frac{2}{3} & \frac{1}{3} \\ \frac{5}{3} & -\frac{1}{3} \end{bmatrix},\ C = \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Следовательно,

$$\nabla_c f = \frac{\partial_c f}{\partial_c x_3} = 2x_3 - (2x_1, 2x_2)\begin{bmatrix} -\frac{2}{3} & \frac{1}{3} \\ \frac{5}{3} & -\frac{1}{3} \end{bmatrix}\begin{bmatrix} 3 \\ 1 \end{bmatrix} = \frac{10}{3}x_1 - \frac{28}{3}x_2 + 2x_3.$$

В стационарной точке выполняется равенство $\nabla_c f = 0$, которое вместе с ограничениями $g_1(X) = 0$ и $g_2(X) = 0$ определяет искомую стационарную точку (или точки). В данном случае система уравнений

$$\begin{bmatrix} 10 & -28 & 6 \\ 1 & 1 & 3 \\ 5 & 2 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 5 \end{bmatrix}$$

имеет решение $X^0 \approx$ (0,81; 0,35; 0,28).

Далее устанавливаем тип полученной стационарной точки путем проверки выполнения достаточных условий экстремума. Так как x_3 – независимая переменная, из равенства $\nabla_c f = 0$ следует, что

$$\frac{\partial_c^2 f}{\partial_c x_3^2} = \frac{10}{3}\left(\frac{dx_1}{dx_2}\right) - \frac{28}{3}\left(\frac{dx_2}{dx_3}\right) + 2 = \left(\frac{10}{3}, -\frac{28}{3}\right)\begin{bmatrix} \dfrac{dx_1}{dx_2} \\ \dfrac{dx_2}{dx_3} \end{bmatrix} + 2.$$

В соответствии с методом Якоби получаем

$$\begin{bmatrix} \dfrac{dx_1}{dx_2} \\ \dfrac{dx_2}{dx_3} \end{bmatrix} = -J^{-1}C = \begin{bmatrix} \dfrac{5}{3} \\ -\dfrac{14}{3} \end{bmatrix}.$$

Теперь путем подстановки находим, что $\frac{\partial_c^2 f}{\partial_c x_3^2} = \frac{460}{9} > 0$. Следовательно, X_0 – точка минимума.

Использование описанного метода Якоби в общем случае может быть затруднено вычислением обратной матрицы J^{-1}, если количество ограничений задачи велико. Однако этого можно избежать, воспользовавшись правилом Крамера, которое позволяет выразить ∂f через ∂Z. Если z_j представляет собой j-ю компоненту вектора Z, и y_i – i-я компонента вектора Y, то можно показать, что

$$\frac{\partial_c f}{\partial_c z_j} = \frac{\dfrac{\partial(f, g_1, g_2, \dots, g_m)}{\partial(z_j, y_1, y_2, \dots, y_m)}}{\dfrac{\partial(g_1, \dots, g_m)}{\partial(y_1, \dots, y_m)}},$$

где

$$\frac{\partial(f, g_1, \dots, g_m)}{\partial(z_j, y_1, \dots, y_m)} = \begin{vmatrix} \dfrac{\partial f}{\partial z_j} & \dfrac{\partial f}{\partial y_1} & \dots & \dfrac{\partial f}{\partial y_m} \\ \dfrac{\partial g_1}{\partial z_j} & \dfrac{\partial g_1}{\partial y_1} & \dots & \dfrac{\partial g_1}{\partial y_m} \\ \dots & \dots & \dots & \dots \\ \dfrac{\partial g_m}{\partial z_j} & \dfrac{\partial g_m}{\partial y_1} & \dots & \dfrac{\partial g_m}{\partial y_m} \end{vmatrix} \text{ и } \frac{\partial(g_1, \dots, g_m)}{\partial(y_1, \dots, y_m)} = \begin{vmatrix} \dfrac{\partial g_1}{\partial y_1} & \dots & \dfrac{\partial g_1}{\partial y_m} \\ \dots & \dots & \dots \\ \dfrac{\partial g_m}{\partial y_1} & \dots & \dfrac{\partial g_m}{\partial y_m} \end{vmatrix} = |J|.$$

Следовательно, необходимые условия экстремума принимают вид

$$\frac{\partial_c f}{\partial_c z_j} = 0, \; j = 1, 2, \dots, n - m.$$

Аналогично в матричном равенстве

$$\frac{\partial Y}{\partial Z} = -J^{-1}C$$

(i, j)-й элемент вычисляется по формуле

$$\frac{\partial y_i}{\partial z_j} = -\frac{\dfrac{\partial(g_1,\ldots,g_m)}{\partial(y_1,\ldots,y_{i-1},z_j,y_{i+1},\ldots,y_m)}}{\dfrac{\partial(g_1,\ldots,g_m)}{\partial(y_1,\ldots,y_m)}},$$

которая представляет собой скорость изменения зависимой переменной y_i, вызванного изменением независимой переменной z_j.

Для получения достаточного условия наличия экстремума i-й элемент вектора $W \equiv \nabla_Y f J^{-1}$ выразим в следующем виде:

$$w_i = \frac{\dfrac{\partial(g_1,\ldots,g_{i-1},f,g_{i+1},\ldots,g_m)}{\partial(y_1,\ldots,y_m)}}{\dfrac{\partial(g_1,\ldots,g_m)}{\partial(y_1,\ldots,y_m)}}.$$

В качестве иллюстрации использования описанного выше метода установим необходимое условие экстремума в задаче из примера 5. В этом случае имеем

$$\frac{\partial_c f}{\partial_c x_3} = \frac{\begin{vmatrix} 2x_3 & 2x_1 & 2x_2 \\ 3 & 1 & 1 \\ 1 & 5 & 2 \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ 5 & 2 \end{vmatrix}} = \frac{10}{3}x_1 - \frac{28}{3}x_2 + 2x_3.$$

Анализ чувствительности с помощью метода Якоби. Метод Якоби можно использовать для анализа чувствительности оптимального значения целевой функции f к малым изменениям правых частей ограничений оптимизационной задачи. В частности, можно определить, как повлияет на оптимальное значение функции f замена ограничения $g_i(X) = 0$ на $g_i(X) = \partial g_i$. Исследование такого типа именуется анализом **чувствительности** и в некотором смысле аналогично соответствующей процедуре, которая была рассмотрена в линейном программировании. Следует, однако, заметить, что анализ чувствительности в нелинейном программировании приводит к результатам, справедливым лишь в малой окрестности экстремальной точки. Знакомство с такими процедурами будет полезно при изучении метода множителей Лагранжа.

Выше было показано, что

$$\partial f(Y,Z) = \nabla_Y f \partial Y + \nabla_Z f \partial Z,$$
$$\partial g = J\partial Y + C\partial Z.$$

Предположим что $\partial g \neq 0$, тогда

$$\partial Y = J^{-1}\partial g - J^{-1}C\partial Z.$$

Подставляя последнее выражение в уравнение для $\partial f(Y,Z)$, получаем

$$\partial f(Y,Z) = \nabla_{Y_0} fJ^{-1}\partial g + \nabla_c f \partial Z,$$

как было определено ранее. Полученное выражение для $\partial f(Y,Z)$ можно использовать при анализе изменений целевой функции f в малой окрестности допустимой точки X_0, обусловленных малыми изменениями ∂g и ∂Z.

В экстремальной (фактически любой стационарной) точке $X_0 = (Y_0, Z_0)$ приведенный градиент $\nabla_c f$ должен быть равен нулю. Следовательно, в точке X_0 имеем выражение

$$\partial f(Y_0,Z_0) = \nabla_{Y_0} fJ^{-1}\partial g(Y_0,Z_0) \text{ или } \frac{\partial f}{\partial g} = \nabla_{Y_0} fJ^{-1},$$

вычисленное в точке X_0. Это значит, что влияние малых изменений ∂g на *оптимальное* значение функции f можно оценить через скорость изменения функции f по отношению к изменениям g. Эти величины принято называть коэффициентами чувствительности.

Пример 1.6

Рассмотрим задачу из примера 5. Оптимальной точкой является $X_0 = (x_1^0, x_2^0, x_3^0)$ = (0,81; 0,35; 0,28). Так как $Y = (x_1^0, x_2^0)$, то

$$\nabla_{Y_0} f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right) = (2_1^0, 2x_2^0) = (1,62; 0,70).$$

Следовательно,

$$\left(\frac{\partial f}{\partial g_1}, \frac{\partial f}{\partial g_2}\right) = \nabla_{Y_0} fJ^{-1} = (1,62; 0,70)\begin{bmatrix} -\frac{2}{3} & \frac{1}{3} \\ \frac{5}{3} & -\frac{1}{3} \end{bmatrix} = (0,0876; 0,3067).$$

Это свидетельствует о том, что изменение ∂g_1 = 1 *приводит к увеличению значения* целевой функции f *приблизительно* на 0,0867. Аналогично при ∂g_2 = 1 имеет место увеличение значения f *приблизительно* на 0,3067.

Использование метода Якоби в задачах линейного программирования. Рассмотрим задачу линейного программирования.

Максимизировать $z = 2x_1 + 3x_2$

при ограничениях

$$x_1 + x_2 + x_3 = 5,$$
$$x_1 - x_2 + x_4 = 3,$$
$$x_1, x_2, x_3, x_4 \geq 0.$$

Для каждого условия неотрицательности $x_j \geq 0$ введем соответствующую (неотрицательную) дополнительную переменную w_j^2. Таким образом, $x_j - w_j^2 = 0$ или $x_j = w_j^2$. При такой замене условия неотрицательности становятся неявными, и исходная задача принимает вид

максимизировать $z = 2w_1^2 + 3w_2^2$

при ограничениях

$$w_1^2 + w_2^2 + w_3^2 = 5,$$
$$w_1^2 - w_2^2 + w_4^2 = 3.$$

Для метода Якоби положим $Y = (w_1, w_2)$ и $Z = (w_3, w_4)$. (В терминологии линейного программирования векторы Y и Z образованы базисными и небазисными переменными соответственно.) Следовательно,

$$J = \begin{bmatrix} 2w_1 & 2w_2 \\ 2w_1 & -2w_2 \end{bmatrix},\ C = \begin{bmatrix} 2w_3 & 0 \\ 0 & 2w_4 \end{bmatrix},\ \nabla_Y f = (4w_1, 6w_2), \nabla_Z f = (0, 0),$$

$$J^{-1} = \begin{bmatrix} \dfrac{1}{4w_1} & \dfrac{1}{4w_1} \\ \dfrac{1}{4w_2} & \dfrac{-1}{4w_2} \end{bmatrix}, w_1 \quad w_2 \neq$$

так что

$$\nabla_c f = (0, 0) - (4w_1, 6w_2)\begin{bmatrix} \dfrac{1}{4w_1} & \dfrac{1}{4w_1} \\ \dfrac{1}{4w_2} & \dfrac{-1}{4w_2} \end{bmatrix}\begin{bmatrix} 2w_3 & 0 \\ 0 & 2w_4 \end{bmatrix} = (-5w_3, w_4).$$

Решение системы уравнений, состоящей из уравнения $\nabla_c f$ и ограничений задачи, позволяет определить стационарную точку $w_1 = 2$, $w_2 = 1$, $w_3 = 0$, $w_4 = 0$. При этом матрица Гессе имеет вид

$$H_c = \begin{bmatrix} \dfrac{\partial_c^2 f}{\partial_c w_3^2} & \dfrac{\partial_c^2 f}{\partial_c w_3 \partial_c w_4} \\ \dfrac{\partial_c^2 f}{\partial_c w_3 \partial_c w_4} & \dfrac{\partial_c^2 f}{\partial_c w_4^2} \end{bmatrix} = \begin{bmatrix} -5 & 0 \\ 0 & 1 \end{bmatrix}.$$

Поскольку H_c является неопределенной матрицей, стационарная точка не будет точкой максимума.

На самом деле полученный результат не является неожиданным, поскольку (небазисные) переменные w_3 и w_4 (и, следовательно, x_3 и x_4) равняются нулю, как и предусмотрено теорией линейного программирования. Следовательно, решение, полученное с помощью метода Якоби, определяет ту или иную крайнюю точку пространства допустимых решений рассматриваемой задачи, которая определяется конкретным выбором векторов Y и Z. При этом найденное решение может как быть, так и не быть оптимальным. Однако метод Якоби позволяет распознать оптимальное решение задачи путем использования достаточных условий экстремума.

Приведенные рассуждения свидетельствуют о том, что для нахождения оптимального решения задачи линейного программирования необходимо менять конкретный выбор Y и Z, пока не будут выполнены достаточные условия. Пусть, например, $Y = (w_2, w_4)$ и $Z = (w_1, w_3)$. Тогда соответствующий приведенный градиент принимает вид

$$\nabla_c f = (4w_1, 0) - (6w_2, 0)\begin{bmatrix} \dfrac{1}{2w_2} & 0 \\ \dfrac{1}{2w_4} & \dfrac{1}{2w_4} \end{bmatrix}\begin{bmatrix} 2w_1 & 2w_3 \\ 2w_1 & 0 \end{bmatrix} = (-2w_1, 6w_3).$$

Соответствующая стационарная точка определяется значениями $w_1 = 0$, $w_2 = \sqrt{5}$, $w_3 = 0$, $w_4 = \sqrt{8}$. Так как матрица

$$H_c = \begin{bmatrix} -2 & 0 \\ 0 & -6 \end{bmatrix}$$

является отрицательно определенной, то указанная точка – точка максимума.

Полученный результат можно проверить графически, используя рис. 1.5. Первое из найденных решений ($x_1 = 4$, $x_2 = 1$) не является оптимальным, в то время как второе ($x_1 = 0$, $x_2 = 5$) оказывается таковым. Читатель имеет возможность проверить, что две оставшиеся крайние точки пространства допустимых решений не являются точками максимума. Кроме того, с использованием достаточных условий экстремума можно показать, что в точке $x_1 = 0$, $x_2 = 0$ имеет место минимум целевой функции рассматриваемой задачи.

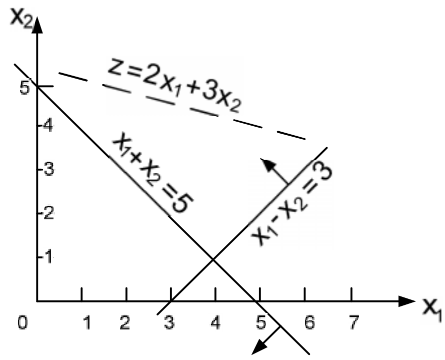


Рис. 1.5

Заметим, что введенные ранее коэффициенты чувствительности $\nabla_{Y_0} fJ^{-1}$ в случае задачи линейного программирования являются переменными двойственной задачи. Чтобы проиллюстрировать это на рассматриваемом численном примере, обозначим через u_1 и u_2 соответствующие переменные двойственной задачи.

В оптимальной точке $w_1 = 0$, $w_2 = \sqrt{5}$, $w_3 = 0$, $w_4 = \sqrt{8}$ переменные двойственной задачи вычисляются по формуле

$$(u_1, u_2) = \nabla_{Y_0} fJ^{-1} = (6w_2, 0)\begin{bmatrix} \frac{1}{2w_2} & 0 \\ \frac{1}{2w_4} & \frac{1}{2w_4} \end{bmatrix} = (3, 0).$$

В точке (3, 0) целевая функция двойственной задачи равна $5u_1 + 3u_2 = 15$ и совпадает с оптимальным значением целевой функции прямой задачи. Полученное решение также удовлетворяет ограничениям двойственной задачи, т.е. является допустимым и, следовательно, оптимальным. Это значит, что коэффициенты чувствительности совпадают с переменными двойственной задачи линейного программирования.

Из процедуры применения метода Якоби к решению задач линейного программирования можно сделать несколько общих выводов. Рассмотренный численный пример показывает, что в силу необходимых условий экстремума независимые переменные задачи равны нулю. Достаточные же условия указывают, что матрица Гессе является диагональной. Следовательно, все диагональные элементы этой матрицы должны быть положительными, если в рассматриваемой точке имеет место минимум, и отрицательными – в случае максимума.

Из вышесказанного следует, что необходимое условие экстремума равносильно указанию на то, что оптимальное решение задачи следует искать только среди ее базисных решений. В этом случае в качестве небазисных переменных задачи линейного программирования выступают независимые переменные. Достаточное же условие приводит к выводу, что между диагональными элементами матрицы Гессе и разностями $z_j - c_j$, задачи линейного программирования, получаемыми с помощью симплекс-метода, существует строгое соответствие.

МЕТОД МНОЖИТЕЛЕЙ ЛАГРАНЖА. Коэффициенты чувствительности метода Якоби

$$\frac{\partial f}{\partial g} = \nabla_{Y_0} f J^{-1}$$

можно использовать для решения задач с ограничениями в виде равенств. Пусть

$$\lambda = \nabla_{Y_0} f J^{-1} = \frac{\partial f}{\partial g}.$$

Следовательно,

$$\partial f - \lambda \partial g = 0.$$

Это уравнение соответствует необходимым условиям стационарности точек, так как формула для $\frac{\partial f}{\partial g}$ получена с учетом того, что $\nabla_c f = 0$. Представленное уравнение можно записать в более удобной форме, если перейти к частным производным по всем переменным x_j, что приводит к системе уравнений

$$\frac{\partial}{\partial x_j}(f - \lambda g) = 0, \; j = 1, 2, \ldots, n.$$

Полученные уравнения вместе с ограничениями задачи $g = 0$ формируют систему, которая позволяет определить допустимые векторы X и λ, удовлетворяющие *необходимым* условиям стационарности.

Описанная процедура составляет основу *метода множителей Лагранжа,* который позволяет определять стационарные точки задачи оптимизации с ограничениями в виде *равенств.* Формально схема этого метода представима в следующем виде. Пусть

$$L(X, \lambda) = f(X) - \lambda \partial g(X).$$

Функция L называется функцией Лагранжа, а параметры X – множителями Лагранжа. Как следует из определения, эти множители имеют тот же смысл, что и коэффициенты чувствительности, фигурирующие в методе Якоби.

Уравнения

$$\frac{\partial L}{\partial \lambda}=0 \text{ и } \frac{\partial L}{\partial X}=0$$

дают необходимые условия для определения стационарных точек функции $f(X)$ при ограничениях $g(X) = 0$. Достаточные условия, используемые в методе множителей Лагранжа, будут сформулированы без доказательства. Определим матрицу

$$H^B=\begin{bmatrix} 0 & | & P \\ - & - & - \\ P^T & | & Q \end{bmatrix}_{(m+n)\times(m+n)},$$

где

$$P=\begin{bmatrix} \nabla g_1(X) \\ \vdots \\ \nabla g_m(X) \end{bmatrix}_{m\times n} \text{ и } Q=\left\| \frac{\partial^2 L(X,\lambda)}{\partial x_i \partial x_j} \right\|_{m\times n} \text{ для всех } i \text{ и } j.$$

Матрица H^B называется **окаймленной матрицей Гессе.**

Пусть имеется стационарная точка (X_0, λ_0) функции Лагранжа $L(X, \lambda)$ и окаймленная матрица Гессе H^B вычислена в точке (X_0, λ_0). Тогда X_0 является:

1) точкой максимума, если, начиная с углового минора порядка $2m+1$, *последующие* $n - m$ угловых миноров окаймленной матрицы Гессе H^B образуют знакопеременный числовой ряд, в котором знак первого члена определяется множителем $(-1)^{m+1}$;

2) точкой минимума, если, начиная с углового минора порядка $2m+1$, *последующие* $n - m$ угловых миноров матрицы H^B имеют знаки, определяемые множителем $(-1)^{m}$.

Эти условия являются достаточными для определения экстремальной точки. Другими словами, экстремальной может оказаться стационарная точка, не удовлетворяющая этим условиям.

Существуют другие условия определения экстремальных точек задачи, которые являются как необходимыми, так и достаточными. Однако их практическое использование часто связано со значительными вычислительными трудностями. Определим матрицу

$$\Delta=\begin{bmatrix} 0 & P \\ P^T & Q-\mu I \end{bmatrix},$$

вычисленную в стационарной точке (X_0, λ_0), где μ – неизвестный параметр. Пусть $|\Delta|$ – определитель матрицы Δ, тогда все $n - m$ действительных корней u_i полинома

$$|\Delta| = 0$$

должны быть:

1) отрицательными, если X_0 – точка максимума;

2) положительными, если X_0 – точка минимума.

Пример 1.7

Рассмотрим задачу из примера 1.5. Функция Лагранжа имеет вид

$$L(X,\lambda) = x_1^2 + x_2^2 + x_3^2 - \lambda_1(x_1 + x_2 + 3x_3 - 2) - \lambda_2(5x_1 + 2x_2 + x_3 - 5).$$

Необходимые условия экстремума записываются в виде следующей системы уравнений:

$$\frac{\partial L}{\partial x_1} = 2x_1 - \lambda_1 - 5\lambda_2 = 0,$$

$$\frac{\partial L}{\partial x_2} = 2x_2 - \lambda_1 - 2\lambda_2 = 0,$$

$$\frac{\partial L}{\partial x_3} = 2x_3 - 3\lambda_1 - \lambda_2 = 0,$$

$$\frac{\partial L}{\partial \lambda_1} = -(x_1 + x_2 + 3x_3 - 2) = 0,$$

$$\frac{\partial L}{\partial \lambda_2} = -(5x_1 + 2x_2 + x_3 - 5) = 0.$$

Решением этой системы являются векторы

$$X_0 = (x_1, x_2, x_3) = (0,81; 0,35; 0,28),$$

$$\lambda = (\lambda_1, \lambda_2) = (0,0867; 0,3067).$$

В этом решении объединены результаты примеров 1.5 и 1.6. Как видим, значения множителей Лагранжа λ совпадают со значениями коэффициентов чувствительности, полученными в примере 1.6. Это указывает на то, что эти коэффициенты не зависят от выбора вектора зависимых переменных Y при реализации метода Якоби.

Чтобы показать, что найденная точка является точкой минимума, рассмотрим матрицу

$$H^B = \begin{bmatrix} 0 & 0 & 1 & 1 & 3 \\ 0 & 0 & 5 & 2 & 1 \\ 1 & 5 & 2 & 0 & 0 \\ 1 & 2 & 0 & 2 & 0 \\ 3 & 1 & 0 & 0 & 2 \end{bmatrix}.$$

Здесь $n = 3$, $m = 2$ и $n - m = 1$. Следовательно, необходимо проверить только определитель матрицы H^B, знак которого должен определяться множителем $(-1)^2$ в точке минимума. Так как $\det H^B = 460 > 0$, то X_0 является точкой минимума.

При решении систем уравнений, выражающих необходимые условия экстремума, иногда удобно применять метод, который заключается в последовательном выборе числовых значений *X*, после чего данная система решается относительно *X*. Процедура повторяется до тех пор, пока вектор *X*, соответствующий некоторому значению λ, не будет удовлетворять всем ограничениям. В вычислительном плане эта процедура является громоздкой. Можно использовать и другие численные методы решения систем уравнений, например метод Ньютона–Рафсона (подразд. 1.2.2).

Пример 1.8

Рассмотрим задачу.

Минимизировать $z = x_1^2 + x_2^2 + x_3^2$

при ограничении

$$4x_1 + x_2^2 + 2x_3 - 14 = 0.$$

Функция Лагранжа имеет вид

$$L(X, \lambda) = x_1^2 + x_2^2 + x_3^2 - \lambda(4x_1 + x_2^2 + 2x_3 - 14).$$

Отсюда получаем необходимые условия экстремума в виде системы

$$\frac{\partial L}{\partial x_1} = 2x_1 - 4\lambda = 0,$$
$$\frac{\partial L}{\partial x_2} = 2x_2 - 2\lambda x_2 = 0,$$
$$\frac{\partial L}{\partial x_3} = 2x_3 - 2\lambda = 0,$$
$$\frac{\partial L}{\partial \lambda} = -(4x_1 + x_2^2 + 2x_3 - 14) = 0,$$

решениями которой являются векторы

$$(X_0, \lambda_0)_1 = (2; 2; 1; 1),$$
$$(X_0, \lambda_0)_2 = (2; -2; 1; 1),$$
$$(X_0, \lambda_0)_3 = (2{,}8; 0; 1{,}4; 1{,}4).$$

Используя достаточные условия, вычислим элементы матрицы

$$H^B = \begin{bmatrix} 0 & 4 & 2x_2 & 2 \\ 4 & 2 & 0 & 0 \\ 2x_2 & 0 & 2-2\lambda & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix}.$$

Так как m = 1 и n = 3, то для того, чтобы стационарная точка была точкой минимума, знак последних 3 – 1 = 2 угловых миноров должен определяться знаком множителя $(-1)^m = -1$. Таким образом, в точке

$(X_0, \lambda_0)_1 = (2; 2; 1; 1)$ имеем

$$\begin{vmatrix} 0 & 4 & 4 \\ 4 & 2 & 0 \\ 4 & 0 & 0 \end{vmatrix} = -32 < 0 \qquad \begin{vmatrix} 0 & 4 & 4 & 2 \\ 4 & 2 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 \end{vmatrix} = - \quad <$$

В точке $(X_0, \lambda_0)_2 = (2; -2; 1; 1)$ имеем

$$\begin{vmatrix} 0 & 4 & -4 \\ 4 & 2 & 0 \\ -4 & 0 & 0 \end{vmatrix} = -32 < 0 \qquad \begin{vmatrix} 0 & 4 & -4 & 2 \\ 4 & 2 & 0 & 0 \\ -4 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 \end{vmatrix} = - \quad <$$

Наконец, в точке $(X_0, \lambda_0)_3 = (2,8; 0; 1,4; 1,4)$ имеем

$$\begin{vmatrix} 0 & 4 & 0 \\ 4 & 2 & 0 \\ 0 & 0 & -0,8 \end{vmatrix} = 12,8 > 0 \qquad \begin{vmatrix} 0 & 4 & 0 & 2 \\ 4 & 2 & 0 & 0 \\ 0 & 0 & -0,8 & 0 \\ 2 & 0 & 0 & 2 \end{vmatrix} = \quad >$$

Отсюда следует, что $(X_0)_1$ и $(X_0)_2$ – точки минимума. Тот факт, что в точке $(X_0)_3$ не удовлетворяются достаточные условия максимума или минимума, не означает, что эта точка не является экстремальной. Это следствие того, что используемые условия являются лишь достаточными.

Для иллюстрации схемы проверки достаточных условий экстремума, которая использует корни полинома, рассмотрим матрицу

$$\Delta = \begin{bmatrix} 0 & 4 & 2x_2 & 2 \\ 4 & 2-\mu & 0 & 0 \\ 2x_2 & 0 & 2-2\lambda-\mu & 0 \\ 2 & 0 & 0 & 2-\mu \end{bmatrix}.$$

В точке $(X_0, \lambda_0)_1 = (2; 2; 1; 1)$ имеем

$$|\Delta| = 9\mu^2 - 26\mu + 16 = 0,$$

откуда получаем $\mu = 2$ и $\mu = 8/9$. Так как все $\mu > 0$, то $(X_0)_1 = (2; 2; 1)$ – точка минимума. В точке $(X_0, \lambda_0)_2 = (2; -2; 1; 1)$ имеем

$$|\Delta| = 9\mu^2 - 26\mu + 16 = 0,$$

что совпадает с рассмотренным выше случаем. Значит, $(X_0)_2 = (2; -2; 1)$ является точкой минимума. Наконец, в точке $(X_0, \lambda_0)_3 = (2,8; 0; 1,4; 1,4)$ имеем

$$|\Delta| = 5\mu^2 - 6\mu + 8 = 0,$$

откуда получаем $\mu = 2$ и $\mu = -0,8$, что свидетельствует о том, что точка $(X_0)_3 = (2,8; 0; 1,4)$ не является экстремальной.

1.3.2. Ограничения в виде неравенств

В данном подразделе показано, как метод множителей Лагранжа можно обобщить на случай задачи с ограничениями в виде неравенств. Основную часть подраздела составляет вывод условий Куна-Таккера, которые играют важную роль в нелинейном программировании.

ОБОБЩЕННЫЙ МЕТОД МНОЖИТЕЛЕЙ ЛАГРАНЖА

Предположим, что дана задача, в которой требуется максимизировать функцию $z = f(X)$
при ограничениях

$$g_i(X) \le 0, i = 1, 2, \dots, m.$$

Если в рассматриваемой задаче имеются условия неотрицательности переменных $X \ge 0$, то предполагается, что они включены в указанные m ограничений.

Основная идея обобщенного метода множителей Лагранжа состоит в том, что если точка *безусловного* оптимума функции $f(X)$ не удовлетворяет всем ограничениям задачи, то оптимальное решение задачи с ограничениями должно достигаться в граничной точке области допустимых решений. Следовательно, одно или несколько из m ограничений задачи должны выполняться как равенства. Поэтому вычислительная процедура обобщенного метода множителей Лагранжа включает следующие шаги.

Шаг 1. Решается задача без учета ограничений

Максимизировать $z = f(X)$.

Если полученное оптимальное решение удовлетворяет всем ограничениям задачи, вычисления заканчиваются, так как все ограничения являются избыточными. Иначе следует положить $k = 1$ и перейти к шагу 2.

Шаг 2. Активизируются любые k ограничений задачи (т.е. преобразовываются в равенства) и оптимизируется функция $f(X)$ при наличии k ограничений методом множителей Лагранжа. Если оптимальное решение этой задачи является допустимым по отношению к остальным ограничениям исходной задачи, вычисления заканчиваются: полученная точка является *локальным* оптимумом. *Локальный* оптимум определяется из множества оптимумов, полученных в результате оптимизации целевой функции $f(X)$ при учете всех комбинаций из k ограничений-равенств, $k = 1, 2, ..., m$. Иначе нужно сделать активными другие k ограничений исходной задачи и повторить данный шаг. Если *все* подмножества, состоящие из k активных ограничений, не приводят к допустимому решению, следует перейти к шагу 3.

Шаг 3. Если $k = m$, вычисления прекращаются: задача не имеет допустимых решений. Иначе необходимо положить $k = k + 1$ и перейти к шагу 2.

В описанной вычислительной процедуре часто игнорируется то важное обстоятельство, что она *не гарантирует* получения глобального оптимума даже в тех случаях, когда задача имеет *единственное* оптимальное решение. Другим важным моментом является ошибочность интуитивного представления, что если $p < q$, то оптимальное значение целевой функции $f(X)$ при p ограничениях-равенствах всегда лучше, чем при q ограничениях-равенствах. В общем случае это имеет место лишь в том случае, когда набор из q ограничений является подмножеством набора из p ограничений. Рассматриваемый ниже пример служит иллюстрацией сказанного.

Пример 1.9

Максимизировать $z = -(2x_1 - 5)^2 - (2x_2 - 1)^2$

при ограничениях

$$x_1 + 2x_2 \leq 2,$$
$$x_1, x_2 \geq 0.$$

Графическое представление данной задачи (рис. 1.6) призвано помочь в понимании аналитических выкладок. Как видим, целевая функция задачи является вогнутой, а область допустимых решений выпуклой. Отсюда следует, что эффективный алгоритм должен гарантировать нахождение глобального оптимума. Однако обобщенный метод множителей Лагранжа, как будет показано далее, позволяет найти лишь точку локального максимума.

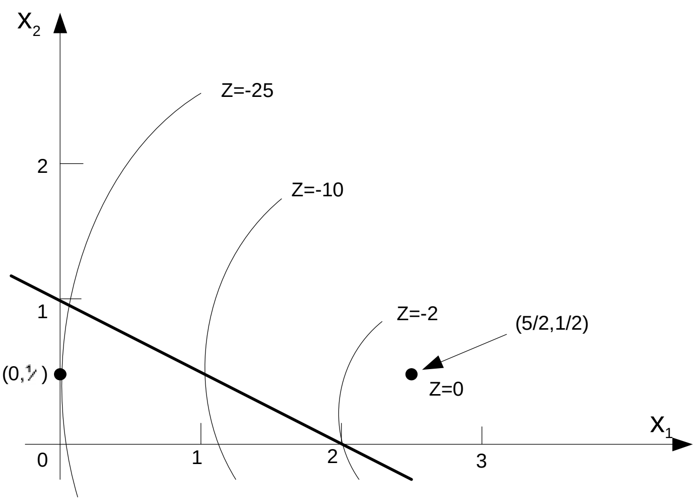


Рис. 1.6

Точка безусловного экстремума находится как решение уравнений

$$\frac{dz}{dx_1} = -4(2x_1 - 5) = 0,$$

$$\frac{dz}{dx_2} = -4(2x_2 - 1) = 0.$$

Отсюда имеем (x_1, x_2) = (5/2, 1/2). Так как это решение не удовлетворяет условию $x_1 + 2x_2 \le 2$, ограничения поочередно активизируются. Рассмотрим ограничение $x_1 = 0$. Функция Лагранжа в этом случае примет вид

$$L(x_1, x_2, \lambda) = -(2x_1 - 5)^2 - (2x_2 - 1)^2 - \lambda x_1.$$

Следовательно,

$$\frac{dL}{dx_1} = -4(2x_1 - 5) - \lambda = 0,$$

$$\frac{dL}{dx_2} = -4(2x_2 - 1) = 0,$$

$$\frac{dL}{d\lambda} = -x_1 = 0.$$

Решением этой системы будет точка (x_1, x_2) = (0, 1/2), которая является точкой максимума, в чем можно убедиться, используя достаточное условие. Поскольку эта точка удовлетворяет остальным ограничениям исходной задачи, вычислительная процедура завершается: (x_1, x_2) = (0, 1/2) – точка локального максимума задачи. (Заметим, что если поочередно активизировать оставшиеся ограничения задачи $x_2 \ge 0$ и $x_1 + 2x_2 \le 2$, то получаются недопустимые решения.) Оптимальное значение целевой функции $z = -25$.

Как видно из рис. 1.6, допустимому решению (x_1, x_2) = (2, 0) рассматриваемой задачи, которое определяется точкой пересечения *двух* прямых $x_2 = 0$ и $x_1 + 2x_2 = 2$, соответствует значение целевой функции $z = -2$. Оно лучше значения функции z, которое получено с учетом одного активного ограничения.

Как следует из описанной вычислительной процедуры, при реализации обобщенного метода множителей Лагранжа на большее, чем получение приемлемого допустимого решения задачи, не следует рассчитывать. Особенно это проявляется тогда, когда целевая функция задачи не является одновершинной. Если в оптимизационной задаче имеется единственный (глобальный) условный экстремум (как в примере 9), то для его нахождения можно использовать откорректированный обобщенный метод множителей Лагранжа. Для этого необходимо провести сравнение безусловного оптимума и условных экстремумов, полученных с учетом *всех* возможных наборов активных ограничений, т.е. когда сначала отдельные ограничения делаются активными поочередно, затем рассматриваются пары активных ограничений и так до тех пор, пока все *m* ограничений рассматриваемой задачи не станут активными. Наилучший из *всех* таких *допустимых* экстремумов является глобальным.

Если эту процедуру применить к решению задачи из примера 9, то для определения глобального экстремума необходимо решить семь задач. Это свидетельствует о том, что возможности использования обобщенного метода множителей Лагранжа для решения практических задач ограничены.

УСЛОВИЯ КУНА–ТАККЕРА. В данном разделе рассмотрены *необходимые* условия Куна–Таккера, которые позволяют определять стационарные точки в задаче нелинейного программирования с ограничениями в виде неравенств. В основу изложения положен метод множителей Лагранжа. Эти условия являются также и достаточными, если выполняются определенные правила, которые описаны ниже.

Рассмотрим следующую задачу.

Максимизировать $z = f(X)$

при ограничениях

$$g(X) \le 0.$$

Ограничения-неравенства можно преобразовать в равенства с помощью *неотрицательных* дополнительных переменных. Пусть S_i^2 (≥ 0) – дополнительная переменная, которая прибавляется к левой части *i*-го ограничения $g_i(X) \le 0$, и пусть

$$S = (S_1, S_2, \dots, S_m)^T \quad S^2 = (S_1^2, S_2^2, \dots, S_m^2)^T$$

где *m* – общее количество ограничений-неравенств. Следовательно, функция Лагранжа записывается в виде

$$L(X, S, \lambda) = f(X) - \lambda\left[g(X) + S^2\right].$$

При ограничениях $g(X) \le 0$ необходимым условием оптимальности в задаче максимизации (минимизации) является неотрицательность (соответственно, неположительность) множителей λ. Приведем обоснование этого условия. Рассмотрим задачу максимизации. Так как множители λ выражают скорость изменения целевой функции f по отношению к изменениям g, т.е.

$$\lambda = \frac{\delta f}{\delta g},$$

то как только правая часть ограничения $g(X) \le 0$ увеличивается и становится больше нуля, область допустимых решений задачи расширяется и, следовательно, оптимальное значение целевой функции не может уменьшиться. Это значит, что $\lambda \ge 0$. Аналогично в случае задачи минимизации при увеличении правой части ограничения оптимальное значение функции f не может увеличиться, откуда следует, что $\lambda \le 0$. Если же ограничения задачи имеют вид равенств, т.е. $g(X) = 0$, то компоненты вектора λ по знаку не ограничены.

Указанные выше ограничения на вектор λ должны рассматриваться как часть необходимых условий Куна–Таккера. Теперь получим остальные условия.

Вычисляя частные производные функции L по X, S и λ и приравнивая их к нулю, получаем

$$\frac{\delta L}{\delta X} = \nabla f(X) - \lambda \nabla g(X) = 0,$$

$$\frac{\delta L}{\delta S_i} = -2\lambda_i S_i = 0, i = 1, 2, \dots, m,$$

$$\frac{\delta L}{\delta \lambda} = -(g(X) + S^2) = 0.$$

Из второй группы уравнений следуют такие результаты.

1. Если λ_i не равняется нулю, то $S_i^2 = 0$. Это означает, что соответствующий этому ограничению ресурс является дефицитным и, следовательно, полностью исчерпан (ограничение имеет вид равенства).

2. Если $S_i^2 > 0$, то $\lambda_i = 0$. Это значит, что i-й ресурс дефицитным не является и, следовательно, не влияет на значение целевой функции f (т.е. $\lambda_i = \frac{\delta f}{\delta g_i} = 0$).

Из второй и третьей групп уравнений следует, что

$$\lambda_i g_i(X) = 0, i = 1, 2, \dots, m.$$

Полученные условия фактически подтверждают предыдущий результат, ибо если $\lambda_i > 0$, то $g_i(X) = 0$ или $S_1^2 = 0$. Аналогично при $g_i(X) < 0$ $S_1^2 > 0$ и, следовательно, $\lambda_i = 0$.

Теперь для задачи максимизации можно сформулировать условия Куна–Таккера, необходимые для того, чтобы векторы X и λ определяли стационарную точку:

$$\begin{aligned}
&\lambda \ge 0,\\
&\nabla f(X) - \lambda \nabla g(X) = 0,\\
&\lambda_i g_i(X) = 0, i = 1, 2, \dots, m,\\
&g(X) \le 0.
\end{aligned}$$

Читатель имеет возможность самостоятельно убедиться, что эти условия применимы также к задаче минимизации, за тем лишь исключением, что вектор λ должен быть неположительным, как это было установлено ранее. Как в случае задачи максимизации, так и задачи минимизации множители Лагранжа, соответствующие ограничениям в виде равенств, по знаку не ограничены.

Достаточность условий Куна–Таккера, Если целевая функция и область допустимых решений рассматриваемой задачи обладают определенными свойствами, связанными с выпуклостью и вогнутостью, то необходимые условия Куна–Таккера являются также достаточными. Упомянутые свойства перечислены в табл. 1.1.

Таблица 1.1

Тип оптимизации	Требуемые свойства	
	Целевая функция	Область допустимых решений
Максимизация Минимизация	Вогнутая Выпуклая	Выпуклое множество Выпуклое множество

Процедура проверки выпуклости или вогнутости некоторой функции является более простой, чем доказательство выпуклости множества допустимых решений экстремальной задачи. Так как выпуклость допустимого множества может быть установлена путем непосредственной проверки выпуклости или вогнутости функций ограничений, по этой причине мы приводим перечень требований, которые легче использовать на практике. Чтобы установить эти требования, рассмотрим задачу нелинейного программирования в общей постановке.

Максимизировать и минимизировать $z = f(x)$
при ограничениях

$$\begin{aligned}
&g_i(x) \le 0, i = 1, 2, \dots, r,\\
&g_i(x) \ge 0, i = r+1, \dots, p,\\
&g_i(x) = 0, i = p+1, \dots, m.
\end{aligned}$$

При этом функция Лагранжа имеет вид

$$L(X,S,\lambda) = f(X) - \sum_{i=1}^{r} \lambda_i \left[g_i(X) + S_i^2 \right] - \sum_{i=r+1}^{p} \lambda_i \left[g_i(X) - S_i^2 \right] - \sum_{i=p+1}^{m} \lambda_i g_i(X),$$

где λ_i – множитель Лагранжа, соответствующий i-му ограничению. Требования, которые устанавливают достаточность условий Куна–Таккера, приведены в табл. 1.2.

В табл. 1.2 представлена лишь часть условий, упомянутых в табл. 1.1. Это связано с тем, что область допустимых решений задачи может быть выпуклой и в то же время функции ограничений, которые ее формируют, не соответствуют требованиям, перечисленным в табл. 1.2.

Таблица 1.2

Тип оптимизации	Требуемые свойства			
	$f(X)$	$g_i(X)$	λ_i	
Максимизация	Вогнутая	Выпуклая	≥0	$(1 \le i \le r)$
		Вогнутая	≤0	$(r+1 \le i \le p)$
		Линейная	Нет ограничений	$(p+1 \le i \le m)$
Минимизация	Выпуклая	Выпуклая	≥0	$(1 \le i \le r)$
		Вогнутая	≤0	$(r+1 \le i \le p)$
		Линейная	Нет ограничений	$(p+1 \le i \le m)$

Законность положений табл. 1.2 основана на том, что при сформулированных условиях функция Лагранжа $L(X, S, \lambda)$ является вогнутой в случае задачи максимизации и выпуклой – в случае задачи минимизации. Это подтверждается тем, что если $g_i(X)$ – выпуклая функция, то функция $\lambda_i g_i(X)$ оказывается выпуклой при $\lambda_i \ge 0$ и вогнутой – при $\lambda_i \le 0$. Аналогичные рассуждения подтверждают все остальные условия. Заметим также, что линейная функция является как выпуклой, так и вогнутой. Наконец, если функция f вогнутая, то функция $-f$ является выпуклой, и наоборот.

Пример 1.10

Рассмотрим следующую задачу.

Минимизировать $f(X) = x_1^2 + x_2^2 + x_3^2$

при ограничениях

$$g_1(X) = 2x_1 + x_2 - 5 \le 0,$$

$$g_2(X) = x_1 + x_3 - 2 \le 0,$$

$$g_3(X) = 1 - x_1 \le 0,$$

$$g_4(X) = 2 - x_2 \le 0,$$

$$g_5(X) = -x_3 \le 0.$$

Поскольку рассматривается задача минимизации, то $\lambda \le 0$. Поэтому условия Куна–Таккера записываются в следующем виде:

$$(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) \le 0,$$

$$(2x_1, 2x_2, 2x_3) - (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} = 0,$$

$$\lambda_1 g_1 = \lambda_2 g_2 = \ldots = \lambda_5 g_5 = 0,$$

$$g(X) \le 0.$$

Эти условия можно упростить и привести к виду

$$\begin{aligned}
&\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 \le 0,\\
&2x_1 - 2\lambda_1 - \lambda_2 + \lambda_3 = 0,\\
&2x_2 - \lambda_1 + \lambda_4 = 0,\\
&2x_3 - \lambda_2 + \lambda_5 = 0,\\
&\lambda_1(2x_1 + x_2 - 5) = 0,\\
&\lambda_2(x_1 + x_3 - 2) = 0,\\
&\lambda_3(1 - x_1) = 0,\\
&\lambda_4(2 - x_2) = 0,\\
&\lambda_5 x_3 = 0,\\
&2x_1 + x_2 \le 5,\\
&x_1 + x_3 \le 2,\\
&x_1 \ge 1, x_2 \ge 2, x_3 \ge 0.
\end{aligned}$$

Отсюда получаем решение: $x_1 = 1, x_2 = 2, x_3 = 0, \lambda_1 = \lambda_2 = \lambda_5 = 0, \lambda_3 = -2, \lambda_4 = -4.$ Поскольку и целевая функция $f(X)$, и область допустимых решений $g(X) \le 0$ являются выпуклыми, функция $L(X, S, \lambda)$ также должна быть выпуклой, и полученная стационарная точка определяет глобальный минимум задачи. Рассмотренный пример показывает также, что решение системы, порожденной условиями Куна–Таккера, в явной форме может быть затруднительным. Следовательно, для численных расчетов описанная процедура не подходит. Тем не менее условия Куна–Таккера играют основополагающую роль при рассмотрении алгоритмов решения задач нелинейного программирования в разд. 3.

1.4. Заключение

В этом разделе изложена классическая теория нахождения максимума и минимума в задачах нелинейного программирования как с ограничениями на переменные, так и без них. Как правило, классическая теория мало подходит для практического применения. Из этого правила существует несколько исключений, когда теоретические положения Куна–Таккера служат основой для создания эффективных вычислительных алгоритмов. В этом смысле квадратичное программирование, которое рассматривается в разд. 3, является примером использования необходимых условий Куна–Таккера.

Для задач нелинейного программирования с ограничениями в виде неравенств нет достаточных условий существования экстремума (подобных условиям для задач без ограничений и задач с ограничениями в виде равенств). Следовательно, не существует какого-либо приемлемого способа проверить, к какому экстремуму сходится тот или иной алгоритм нелинейного программирования – к локальному или глобальному, за исключением тех случаев, которые перечислены в табл. 1.1 и 1.2, когда такие условия можно установить заранее.

2. МЕТОДЫ ОДНОМЕРНОЙ МИНИМИЗАЦИИ

2.1. Предварительные замечания

В этом разделе представлены методы решения задач нелинейного программирования без ограничений на переменные, которые обобщенно называют методами *прямого поиска*. Эти методы построены на реализации прямого поиска оптимальной точки в заданной области. Методы прямого поиска применяются главным образом для нахождения экстремумов функций одной переменной. Несмотря на то, что такие задачи могут показаться тривиальными, методы прямого поиска находят применение и при решении многомерных задач.

Идея методов прямого поиска проста. Сначала определяется интервал неопределенности, который содержит искомую точку оптимума. Затем длина интервала последовательно уменьшается до тех пор, пока не будет найдена точка оптимума. Процедура строится таким образом, что длину интервала, содержащего оптимальное решение, можно сделать сколь угодно малой.

В некоторых случаях *ограничения* в *задаче оптимизации* позволяют через один из *параметров оптимизации* выразить остальные и исключить их из *целевой функции*. В результате задача будет сведена к поиску наибольшего или наименьшего (в зависимости от цели оптимизации) значения скалярной действительной функции $f(x)$, $x \in D(f) \subset \mathrm{R}$, выражающей *критерий оптимальности*. Выбирая тот или иной знак перед этой функцией, всегда можно ограничиться лишь поиском ее наименьшего значения в области определения $D(f)$, заданной с учетом ограничений на параметр оптимизации x. Поэтому далее в этом разделе будем рассматривать задачу

$$f(x) \to min,\ x \in D(f) \subset \mathrm{R}, \qquad (2.1)$$

поиска наименьшего значения $f = f(x_*)$ функции $f(x)$ и точки $x_* \in D(f)$, в которой $f(x)$ принимает это значение. Для краткости будем говорить об *одномерной минимизации*, имея в виду нахождение наименьшего значения функции $f(x)$ на множестве $D(f)$ и точек, в которых это значение достигается.

Изучение методов одномерной минимизации важно не только для решения задачи (2.1), имеющей самостоятельное значение. Эти методы являются также существенной составной частью методов многомерной минимизации, при помощи которых находят наименьшее значение действительных функций многих переменных.

Пусть область определения $D(f)$ функции $f(x)$ есть промежуток числовой прямой. Напомним, что если $D(f)$ – отрезок и $f(x)$ непрерывна на нем, то она имеет на этом отрезке наименьшее значение. Но при наличии на отрезке точек разрыва функции она может не иметь на нем наименьшего значения. Оно может не существовать и в том случае, когда $D(f)$ является интервалом или полуинтервалом.

Если функция $f(x)$ не имеет на множестве $D(f)$ наименьшего значения, то (2.1) следует заменить формулировкой задачи в виде

$$f(x) \to inf,\ x \in D(f) \subset \mathrm{R}. \qquad (2.2)$$

Тогда под решением задачи минимизации такой функции на $D(f)$ следует понимать построение последовательности $\{x_n\}$ точек из $D(f)$, для которой существует предел

$$\lim_{n\to\infty} f(x_n) = \inf_{x_n \in D(f)} f(x) = \tilde{f}_*, \qquad (2.3)$$

и нахождение этого предела. Если функция $f(x)$ достигает на множестве $D(f)$ своего наименьшего значения f_*, то $\tilde{f}_* = f_*$.

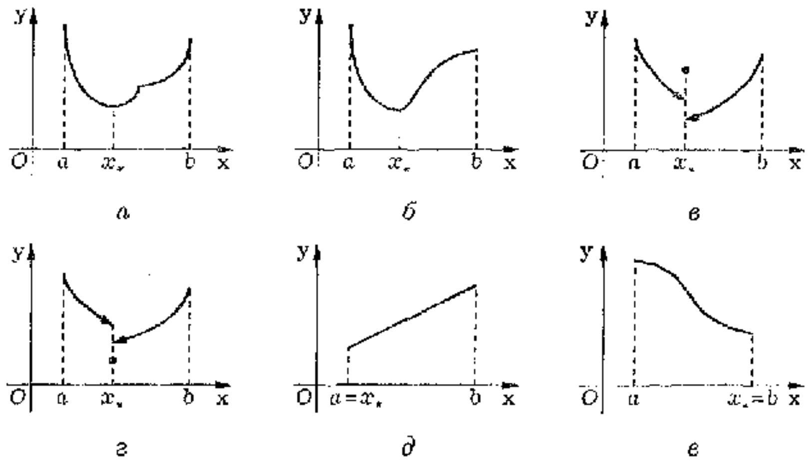
Например, функция $f(x) = 1 / x$ на множестве $D(f) = [1, 2)$ не достигает наименьшего значения, хотя и ограничена снизу. Точная нижняя грань $\tilde{f}_*$ функции в данном случае равна 1/2. В качестве последовательности $\{x_n\}$ точек в полуинтервале [1. 2), для которой справедливо (2.3), можно выбрать $\{2 – 1/n\}$. Тогда

$$f(x_n) = \frac{1}{x_n} = \frac{1}{2 - 1/n} = \frac{n}{2n-1},$$

и последовательность $\{f(x_n)\}$ сходится к числу ½ = $\tilde{f}_*$.

Функция может достигать наименьшего значения как в единственной точке, так и на некотором множестве точек, конечном, счетном или несчетном. Например, функция $f(x) = x^4$ определена на всей числовой прямой и достигает своего наименьшего значения $f_* = 0$ в единственной точке $x_* = 0$ которая является ее точкой минимума. Функция $f(x) = x^4 – 2x^2 + 2$ также определена на всей числовой прямой и достигает наименьшего значения $f_* = 1$ в точках $x_* = \pm 1$. Функция $f(x) = cos\, x$ достигает наименьшего значения на счетном множестве $D_* = \{x \in R: x = \pi + 2\pi k, k \in Z\}$, а функция $f(x) = |x + 1| +$ + $|x – 1|$ – на несчетном множестве $D_* = [–1; 1]$.

Функцию $f(x)$ называют ***унимодальной функцией*** на отрезке $[a, b]$, если существует такая точка $x_* \in [a, b]$, что функция $f(x)$ в полуинтервале $[a, x_*)$ убывает, а в полуинтервале $(x_*, b]$ возрастает. Примеры графиков унимодальных функций приведены на рис. 2.1.



Точка x_* может быть внутренней точкой отрезка $[a, b]$ (т.е. $a < x_* < b$, рис. 2.1 *а–г*) или совпадать с одним из его концов ($x_* = a$ или $x_* = b$, рис. 2.1, *д*, *е*). Унимодальная функция не обязательно непрерывна на отрезке $[a, b]$ (рис. 2.1, *в*, *г*).

Функцию $f(x)$, достигающую на отрезке $[a, b]$ наименьшего значения в единственной точке $x_* \in [a, b]$, убывающую при $x \in [a, x_*]$ и возрастающую при $x \in [x_*, b]$, будем называть ***строго унимодальной*** на отрезке $[a, b]$ (на рисунке 2.1 строго унимодальными являются все функции, кроме функции на рис. 2.1, *г*).

Область определения $D(f)$ минимизируемой функции $f(x)$ может состоять из нескольких промежутков, не имеющих граничных точек. В этом случае, чтобы найти значение функции на множестве $D(f)$, достаточно определить наименьшее значение функции в каждом из промежутков, составляющих $D(f)$, а затем, сравнивая, выбрать среди этих значений минимальное.

Если функция дифференцируема в промежутке, то возможно использование необходимого и достаточного условий локального минимума. Однако в прикладных задачах нередки ситуации, когда трудно вычислить производные функции (например, если функция не задана в аналитическом виде). Более того, не исключено, что значения функции известны или могут быть вычислены только в отдельных точках. В таких ситуациях использование необходимого и достаточного условий локального минимума невозможно и следует применять другие методы решения задачи оптимизации. Методы минимизации функции одного переменного, в которых используют значения функции в точках рассматриваемого промежутка и не используют значения ее производных, называют ***методами прямого поиска.***

2.2. Пассивный и последовательный поиск

Пусть требуется найти наименьшее значение или точную нижнюю грань f_* скалярной действительной функции $f(x)$ одного переменного на отрезке $[a, b]$. Предположим, что задан алгоритм вычисления значения функции для любой точки $x \in [a, b]$. Можно выделить две группы *методов прямого поиска*, соответствующие двум принципиально различным ситуациям:

1) все N точек x_k, $k = 1, 2, \ldots, N$, в которых будут вычислены значения функции, выбирают заранее (до вычисления функции в этих точках);

2) точки x_k выбирают последовательно (для выбора последующей точки используют значения функции, вычисленные в предыдущих точках).

В первом случае поиск значения f_* называют ***пассивным***, а во втором – ***последовательным***. Естественно ожидать, что последовательный поиск лучше пассивного. В этом можно убедиться, вспомнив детскую игру, в которой надо найти спрятанную вещь, задавая вопросы и получая на них ответы „да” или „нет". Задавая вопросы последовательно с учетом предыдущих ответов, можно найти спрятанную вещь за меньшее число вопросов (итераций), чем, задав определенное количество заранее подготовленных вопросов сразу.

Так как в прикладных задачах вычисление каждого значения функции может быть достаточно трудоемким, то целесообразно выбрать такую стратегию поиска, чтобы значение f_* с заданной точностью было найдено наиболее экономным путем. Будем считать, что стратегия поиска определена, если:

- определен алгоритм выбора точек x_k, $k = 1, 2, \ldots, N$;
- определено условие прекращения поиска, т.е. условие, при выполнении которого значение f_* считают найденным с заданной точностью.

Для методов пассивного поиска алгоритм выбора точек x_k, $k = 1, 2, \ldots, N$ – это правило, по которому заранее определяют все N точек x_k, $k = 1, 2, \ldots, N$, в которых затем будут вычислены значения функции $f(x)$. Для методов последовательного поиска алгоритм выбора точек x_k – это правило, по которому последовательно определяют каждую следующую точку x_k по информации о расположении точек x_i, i = 1, 2, …, k–1 и о вычисленных значениях $f(x_i)$ функции $f(x)$ в этих точках. Выбор очередной точки x_k и вычисление значения $f(x_k)$ называют ***шагом*** последовательного *поиска*.

В методах последовательного поиска количество точек x_k обычно не задают заранее. Однако объективное сравнение различных методов прямого поиска нужно проводить при одинаковом количестве n вычисленных значений функции $f(x)$. После n вычислений обычно указывают интервал (или отрезок) длиной l_n, называемый ***интервалом неопределенности***, в котором гарантированно находится точка x_*, соответствующая значению f_*. Условие прекращения вычислений в случае пассивного или последовательного поиска примем одинаковым – выполнение неравенства $l_n \le \varepsilon_*$, где ε_* – заданная наибольшая допустимая длина интервала неопределенности.

Длина l_n зависит как от самого метода прямого поиска p, так и от минимизируемой функции $f(x)$, т.е. $l_n = l_n(p, f)$. Зависимость l_n от n дает оценку скорости сходимости конкретного метода прямого поиска p к искомому значению f_* заданной функции $f(x)$. Различные методы из некоторого множества P методов прямого поиска сравнивают обычно при выбранном фиксированном значении $n = N$ на некотором достаточно широком классе функций. В качестве такого класса можно выбрать множество F *унимодальных функций*, определенных на фиксированном отрезке $x \subset$ R. Для метода прямого поиска $p \in P$ примем наихудшую оценку

$$l_n(p) = \max_{f \in F} l_N(p, f)\,.$$

Если „наихудшей” унимодальной функции не найдется, то оценку принимаем в виде

$$l_n(p) = \sup_{f \in F} l_N(p, f)\,.$$

Значение $l_N(P)$ представляет собой оценку сверху погрешности вычисления точки $x_* \in X$ соответствующей искомому значению f_* произвольной функции $f \in F$, которая получена методом прямого поиска $p \in P$ по N вычисленным значениям этой функции. Метод прямого поиска P^* считаем наилучшим, если

$$l_N(p^*) = \min_{p \in P} \max_{f \in F} l_N(p, f) \quad \text{или} \quad l_N(p^*) = \min_{p \in P} \sup_{f \in F} l_N(p, f)\,.$$

Этот критерий сравнения методов поиска определяет ***минимаксный метод поиска***. Такой метод является наилучшим для всего множества F унимодальных функций на отрезке $x \subset R$ в том смысле, что он дает наименьшую погрешность вычисления точки x_*, соответствующей значению f_* любой из рассматриваемых функций $f \in F$. Хотя вполне возможно, что существует некоторый конкретный метод, который для определенной специально подобранной унимодальной функции из множества F, обеспечит еще меньшую погрешность.

Все методы прямого поиска можно строить и сравнивать между собой на отрезке $X = [0, 1]$. Полученные результаты при необходимости нетрудно перенести на случай произвольного отрезка $[a, b]$, так как любую точку отрезка $[0,1]$ можно перевести в соответствующую ей точку отрезка $[a, b]$ растяжением в $b - a$ раз и сдвигом на a.

Если минимизируемая функция $f(x)$ не является унимодальной на отрезке $[a, b]$ (такую функцию называют ***мультимодальной функцией*** на этом отрезке), то даже если она непрерывна на $[a, b]$, при поиске наименьшего значения f_* функции на отрезке может возникнуть ошибка: будет найдена точка локального минимума, в которой значение функции не f_*, а другое, большее. Чтобы избежать такой ошибки, в процесс минимизации включают предварительный этап, на котором отрезок минимизации разделяют на несколько отрезков, на каждом из которых минимизируемая функция унимодальна. Сравнительный анализ наименьших значений функции на этих отрезках позволяет найти искомое наименьшее значение f_* на всем отрезке минимизации.

Пример 2.1

Рассмотрим один из возможных подходов к выделению из промежутка X в области определения $D(f)$ минимизируемой функции $f(x)$ отрезка $[a, b]$, на котором эта функция является унимодальной. Пусть известна такая точка $x_0 \in X$, что при $x \geq x_0$ функция $f(x)$ сначала убывает, а затем, начиная с пока неизвестного значения $x = x_* \in X$ возрастает, хотя далее в промежутке X могут быть расположены и другие участки немонотонного поведения этой функции. Выберем начальное значение $h > 0$ приращения аргумента x функции $f(x)$, в несколько раз меньшее предполагаемого расстояния между точками x_0 и x_*, и вычислим значения $f(x)$ и $f(x_1)$, где $x_1 = x_0 + h$.

Может оказаться, что $f(x_0) \leq f(x_1)$. Тогда за искомый отрезок $[a, b]$

можно сразу принять отрезок $[x_0, x_1]$. Но можно продолжить вычисления и, используя последовательный поиск, определять значения $f(x_k)$ в точках $x_k = x_0 + h / 2^{k-1}$, $k = 2, 3, \ldots$, до тех пор, пока не будет выполнено неравенство $f(x_k) \le f(x_0)$. Тогда следует принять $[a, b] = [x_0, x_{k-1}]$ (на рис. 2.2 $[a, b] = [x_0, x_2]$, поскольку точка x_* должна быть либо на отрезке $[x_0, x_3]$, либо на отрезке $[x_3, x_2]$). Однако при этом можно „не заметить” по крайней мере, еще один отрезок, на котором функция унимодальна (штриховая линия на рис. 2.2).

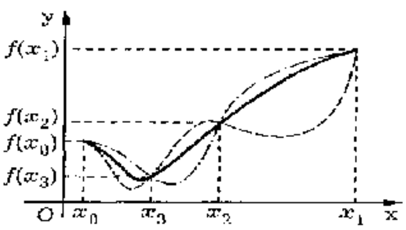


Рис. 2.2

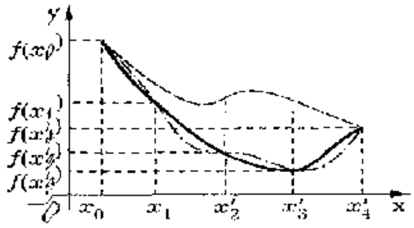


Рис.2.3

Если $f(x_0) > f(x_1)$, то используя последовательный поиск, вычисляем значения $f(x_k')$, где $x_k' = x_0 + (k - 1)h$, $k = 2, 3, \ldots$, пока не будет выполнено неравенство $f(x_{k-1}') \le f(x_k')$, что позволяет принять $[a, b] = [x_{k-2}', x_k']$ (на рис. 2.3 $[a, b] = [x_2', x_4']$, так как точка x_* должна быть либо на отрезке $[x_2', x_3']$, либо на отрезке $[x_3', x_4']$).

Отметим, что описанный подход не гарантирует нахождения отрезка унимодальности функции. Например, на рис. 2.3 штриховой линией показан график функции, для которой этот подход не позволяет обнаружить искомый отрезок.

2.3. Оптимальный пассивный поиск

Пусть требуется путем *пассивного поиска* найти точку $x_* \in [0, 1]$, в которой *унимодальная* на отрезке [0, 1] *функция* $f(x)$ достигает наименьшего значения $f_* = f(x_*)$. *Минимаксный метод поиска*, в котором информация о значениях функции, вычисленных в предшествующих точках, не может быть использована, называют ***оптимальным пассивным поиском***. Рассмотрим алгоритм такого поиска при различном числе N точек, выбираемых на отрезке [0, 1].

Если $n = 1$, то единственную точку целесообразно выбрать в середине отрезка, т.е. принять $x_1 = 1/2$ (рис. 2.4). В этом случае вследствие унимодальности функции $f(x)$ имеем $f_* \le f(1/2)$. Поэтому наименьшая возможная длина *интервала неопределенности* равна $l_1^* = 1$ и можно гарантировать, что выбор в качестве точки $x_* \in [0, 1]$ точки $x_1 = 1/2$ приведет к погрешности не более $\Delta_1^* = l_1^*/2 = 1/2$. При любом ином положении точки x_1 погрешность при выборе $x_* = x_1$ будет $\Delta_1 \geq \Delta_1^*$, так как в действительности точка x_* может лежать на большей части отрезка [0, 1].

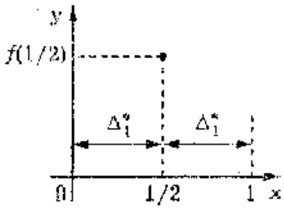


Рис. 2.4

Если при $n = 2$ (рис. 2.5) две точки расположить на отрезке [0,1] так, чтобы они делили его на равные части, т.е. выбрать $x_1 = 1/3$ и $x_2 = 2/3$, то точка $x_* \in [0, 1]$ будет найдена с точностью $\Delta_2^* = 1/3$, а наименьшая длина интервала неопределенности составит $l_2^* = 2\Delta_2^* = 2/3$. В самом деле, если $f(1/3) < f(2/3)$ (рис. 2.5, *а*), то в силу унимодальности функции $f(x)$ отрезок [2/3, 1] можно исключить и считать, что $x_* \in [0, 2/3]$. Тогда при выборе $x_* = 1/3$ наибольшая погрешность равна $\Delta_2 = 1/3$ и $f_* \approx f(1/3)$. Если же окажется, что $f(1/3) > f(2/3)$ (рис. 2.5, *б*), то можно исключить отрезок [0,1/3] и считать, что $x_* \in [1/3, 1]$. И в этом случае выбор $x_* = 1/3$ приведет к погрешности не более $\Delta_2 = 1/3$ и $f_* \approx f(1/3)$. Заметим, что при $f(1/3) = f(2/3)$ можно исключить любой из указанных отрезков, гарантируя ту же точность нахождения точки $x_* \in [0, 1]$. При ином делении отрезка [0, 1] на части двумя точками длина какой-то из его частей будет больше 1/3 и в действительности точка x_* может принадлежать именно этой части, так что получим погрешность $\Delta_2 > \Delta_2^* = 1/3$.

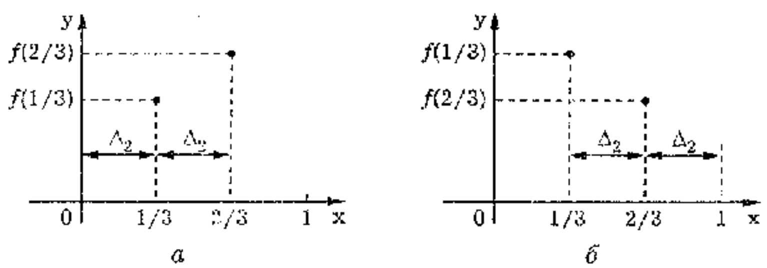


Рис. 2.5

Рассуждая аналогично, можно заключить, что при $n = 3$ нужно также выбирать точки равномерно на отрезке [0,1]: $x_1 = 1/4$, $x_2 = 2/4$, $x_3 = 3/4$, обеспечив точность $\Delta_3^* = 1/4$ нахождения точки $x_* \in [0, 1]$ и наименьшую длину $l_3^* = 1/2$ интервала неопределенности. В случае произвольного $n \in N$ по тем же соображениям надо выбирать точки

$$x_k = \frac{k}{N+1} \in [0,1], \quad k = \overline{1, N}, \tag{2.4}$$

обеспечивая точность $\Delta_N^* = 1/(N + 1)$ нахождения точки x_* и наименьшую возможную длину

$$\Delta_N^* = \frac{2}{N+1} \tag{2.5}$$

интервала неопределенности. Таким образом, оптимальный пассивный поиск состоит в выборе точек, равномерно расположенных на отрезке. При этом (2.5) дает оценку скорости сходимости пассивного поиска с ростом числа N точек, так как скорость сходимости любого метода прямого поиска можно характеризовать скоростью уменьшения интервала неопределенности с возрастанием N.

Пример 2.2

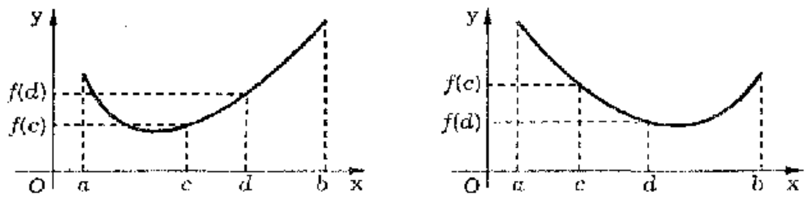
При заданной наибольшей допустимой длине $\varepsilon_* = 0,2$ интервала неопределенности, используя оптимальный пассивный поиск, найдем точку $x_* \in [0, 1]$, в которой унимодальная на отрезке [0, 1] функция $f(x) = x^3 - x + e^{-x}$ достигает наименьшего на этом отрезке значения. Из (2.5) следует, что для этого необходимо принять $N = 9$ и в соответствии с (2.4) вычислить значения функции $f(x)$ в точках $x_k = k / 10$, $k = \overline{1,9}$:

x	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
$f(x)$	0,81	0,63	0,47	0,33	0,23	0,17	0,14	0,16	0,18

Из представленных результатов вычислений можно сделать вывод, что интервалом неопределенности является интервал (0,6, 0,8), а $x_* = 0,7 \pm 0,1$. Отметим, что при $\varepsilon_* = 0,01$ потребуется принять $N = 199$.

Рассуждения, проведенные выше, попутно обосновывают ***процедуру исключения отрезка***, которую используют во всех методах прямого поиска точки минимума унимодальной функции одного переменного. Опишем эту процедуру. Пусть на отрезке $[a, b]$ числовой прямой расположены две точки c и d, $a < c < d < b$ и известны (или вычислены) значения $f(c)$ и $f(d)$ унимодальной на $[a, b]$ функции $f(x)$. Если $f(c) < f(d)$ (рис. 2.6, *а*), то в силу унимодальности функции $f(x)$ имеем $x_* \in [a, d]$, а отрезок $[d, b]$ можно исключить из дальнейшего рассмотрения. Наоборот, если $f(c) \geq f(d)$ (рис. 2.6, *б*), то $x_* \in [c, b]$, а отрезок $[a, c]$ далее можно не рассматривать.

Таким образом, в результате применения процедуры исключения отрезка получаем новый отрезок, вложенный в рассматриваемый и заведомо содержащий искомую точку x_*.



а *в*

Рис. 2.6

В методах пассивного поиска применение этой процедуры позволяет оценить наибольшую возможную погрешность нахождения точки x_*. Все рассмотренные далее методы *последовательного поиска* используют процедуру исключения отрезка для выбора нового отрезка на каждом очередном *шаге* такого поиска.

2.4. Методы последовательного поиска

Метод дихотомии. Рассмотрим последовательный поиск точки $x_* \in [0, 1]$, в которой унимодальная на отрезке [0, 1] *функция* $f(x)$ достигает наименьшего значения $f_* = f(x_*)$. Метод прямого поиска, основанный на делении пополам отрезка, на котором находится точка x_*, называют ***методом дихотомии***. Опишем алгоритм этого метода.

Пусть известно, что на k-м шаге последовательного поиска $x_* \in [a_k, b_k] \subset [0, 1]$ (на первом шаге при $k = 1$ имеем $a_1 = 0$ и $b_1 = 1$). На отрезке $[a_k, b_k]$ длиной l_k выберем две точки $x_{k1} = (a_k + b_k)/2 - \delta$ и $x_{k2} = (a_k + b_k)/2 + \delta$ (рис. 2.7), где $\delta > 0$ – некоторое достаточно малое число. Вычислим значения $f(x_{k1})$ и $f(x_{k2})$ функции $f(x)$ в этих точках и выполним процедуру исключения отрезка. В результате получим новый отрезок $[a_{k+1}, b_{k+1}] \subset [a_k, b_k]$. Если длина $l_k + 1$ нового отрезка больше заданной наибольшей допустимой длины ε_* интервала неопределенности, то алгоритм метода дихотомии переходит к $(k + 1)$–му шагу, повторяя все описанные для k-го шага действия. Если же $l_{k+1} \le \varepsilon_*$, то вычисления прекращают и полагают $x_* = (a_{k+1} + b_{k+1}) / 2$.

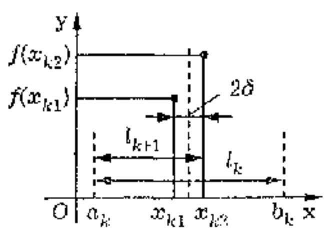


Рис. 2.7

Так как $l_{k+1} = l_k / 2 + \delta$, или $l_{k+1} - 2\delta = (l_k - 2\delta) / 2$, то имеем

$$l_k - 2\delta = \frac{l_1 - 2\delta}{2^{k-1}}.$$

Из этого равенства выводим следующую формулу длины l_k отрезка $[a_k, b_k]$, получаемого на k-м шаге метода дихотомии:

$$l_k = \frac{l_1 - 2\delta}{2^{k-1}} + 2\delta. \qquad (2.6)$$

Из (2.6) следует, что $l_k \to 2\delta$ при $k \to \infty$, но при этом $l_k > 2\delta$. Поэтому выполнение неравенства $l_{k+1} < \varepsilon_*$, означающее достижение заданной точности нахождения точки x_*, возможно лишь при условии выбора $2\delta < \varepsilon_*$. Кроме того, нужно учитывать неизбежную погрешность, возникающую при вычислении приближенных значений $\tilde{f}(x)$ функции $f(x)$. Это приводит к дополнительной погрешности Δ_* при нахождении точки x_* (рис. 2.7). Поэтому выбор значения δ ограничен и снизу, т.е.

$$\Delta_* < 2\delta < \varepsilon_*. \qquad (2.7)$$

Если эти неравенства нарушаются, то знак разности $\tilde{f}(x_{k1}) - \tilde{f}(x_{k2})$ может не совпадать со знаком разности $f(x_{k1}) - f(x_{k2})$, что приводит к ошибочному выполнению процедуры исключения отрезка.

Итак, метод дихотомии – это последовательное построение на каждом k-м шаге поиска точек $x_{k1} = (a_k + b_k)/2 - \delta$ и $x_{k2} = (a_k + b_k)/2 + \delta$, симметричных относительно середины отрезка $[a_k, b_k]$ длиной l_k. После выполнения k-го шага будет выделен отрезок $[a_{k+1}, b_{k+1}]$ длиной l_{k+1} и вычислено $N = 2k$ значений функции. Используя формулу (2.6) для длины отрезка (интервала неопределенности) и полагая $l_1 = 1$, получаем

$$l_N^d = l_{k+1} = \frac{1 - 2\delta}{2^{k+1}} + 2\delta = \frac{1 - 2\delta}{2^{N/2}} + 2\delta. \qquad (2.8)$$

Сравнивая (2.8) с (2.5), видим, что скорость сходимости метода дихотомии значительно выше скорости сходимости оптимального пассивного поиска.

Отметим, что после исключения отрезка на k-м шаге описанного алгоритма точки x_{k1} и x_{k2} принадлежат новому отрезку $[a_{k+1}, b_{k+1}]$, причем одна из них является внутренней для этого отрезка. Но вычисленное в этой точке значение функции $f(x)$ в методе дихотомии не используют для исключения отрезка на следующем шаге, а проводят вычисления в двух новых точках. Существуют методы последовательного поиска, в которых на каждом k-м шаге, начиная с $k = 2$, вычисляют лишь одно новое значение функции в точке, принадлежащей отрезку $[a_{k+1}, b_{k+1}]$. Это значение вместе с уже вычисленным на предыдущем шаге значением функции во внутренней точке отрезка [a_k, b_k] используют при выполнении процедуры исключения отрезка на следующем шаге последовательного поиска.

Метод золотого сечения. Как известно, ***золотым сечением отрезка*** называют такое его деление на две неравные части, при котором отношение длины всего отрезка к длине его большей части равно отношению длины большей части к длине меньшей.

Термин „золотое сечение” ввел Леонардо да Винчи. Золотое сечение широко применяли при композиционном построении многих произведений мирового искусства, в том числе в античной архитектуре и в эпоху Возрождения.

Рассмотрим k-й шаг поиска. Чтобы выполнить процедуру исключения отрезка на этом шаге, отрезок [a_k, b_k] необходимо двумя внутренними точками x_{k1}, $x_{k2,}$ $x_{k1} < x_{k2}$ разделить на три части. Эти точки выберем симметрично относительно середины отрезка [a_k, b_k] (рис. 2.8) и так, чтобы каждая из них производила золотое сечение отрезка [$a_к$, b_k]. В этом случае отрезок [a_{k+1}, b_{k+1}] внутри будет содержать одну из точек x_{k1}, x_{k2} (другая будет одним из концов отрезка), причем эта точка будет производить золотое сечение отрезка [a_{k+1}, b_{k+1}]. Это вытекает из равенства длин отрезков [a_k, x_{k1}] и [x_{k2}, b_k]. Таким образом, на (k + 1)–м шаге в одной из точек $x_{k+1,1}$ и $x_{k+1,2}$ значение функции вычислять не нужно. При этом отношение l_k / l_{k+1} длин отрезков сохраняется от шага к шагу, т.е.

$$\frac{l_k}{l_{k+1}} = \frac{l_{k+1}}{l_{k+2}} = r = const\,. \qquad (2.9)$$

Число r называют ***отношением золотого сечения***.

Последовательный поиск, в котором на k-м шаге каждая из симметрично выбранных на отрезке [a_k, b_k] точек $x_{к1}$, x_{k2} осуществляет золотое сечение этого отрезка, называют ***методом золотого сечения***. В этом методе каждое исключение отрезка уменьшает оставшийся отрезок в r раз.

Выясним, чему равно отношение золотого сечения. Так как точки x_{k1} и x_{k2}, $x_{k1} < x_{k2}$, выбраны симметрично относительно середины отрезка [a_k, b_k], то

$$b_k – x_{k2} = x_{k1} – a_k = l_k – l_{k\text{-}1.}$$

(рис. 2.8). Будем считать, что на k–м шаге выбран отрезок [a_k, x_{k2}]. Тогда на (k + 1)–м шаге одной из точек деления (а именно правой) будет точка x_{k1}.

Значит, длина l_{k+2} отрезка, выбираемого на (k + 1)–м шаге, совпадает с длиной отрезка [a_k, x_{k1}] и верно равенство $l_{k+2} = l_k - l_{k-1}$.Подставляя найденное выражение для l_{k+2} в уравнение (2.9), получаем

$$\frac{l_k}{l_{k+1}} = \frac{l_{k+1}}{l_k - l_{k+1}},$$

или $r = 1/(r - 1)$. Преобразуя это соотношение, приходим к квадратному уравнениюn $r^2 - r - 1 = 0$, имеющему единственное положительное решение $r = \frac{1+\sqrt{5}}{2} \approx 1{,}618034$.

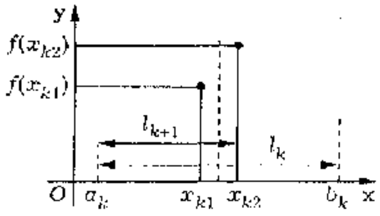


Рис. 2.8

Предположим, что отрезком минимизации унимодальной функции $f(x)$ является [0, 1], т.е. $a_1 = 0$, $b_1 = 1$ и $l_1 = 1$. На первом шаге последовательного поиска ($k = 1$) на отрезке [0, 1] выбираем две точки $x_{11} = a_1 + (1 - 1/r)b_1 = 1 -$ $- 1/r$ и $x_{12} = a_1 + b_1/r = 1/r$, осуществляющие золотое сечение отрезка [0, 1]. Вычисляем значения минимизируемой функции в этих точках и выполняем процедуру исключения отрезка. Если $f(x_{11}) < f(x_{12})$, то выбираем отрезок [a_1, x_{12}], т.е. полагаем $a_2 = a_1 = 0$, $b_2 = x_{12}$; в противном случае выбираем отрезок [x_{11}, b_1], т.е. полагаем $a_2 = x_{11}$, $b_2 = b_1$=1. Кроме того, в первом случае принимаем $\tilde{x}_2 = x_{11}$, а во втором случае $\tilde{x}_2 = x_{12}$. Точка $\tilde{x}_2$ – одна из точек, осуществляющих золотое сечение отрезка [a_2, b_2], меньшая в первом случае и большая во втором. Если длина вновь полученного отрезка больше заданной допустимой длины ε_* интервала неопределенности, то следует перейти ко второму шагу алгоритма, на котором одна из точек x_{21}, x_{22} есть точка $\tilde{x}_2$, а вторую можно найти, например, по формуле $a_2 + b_2 - \tilde{x}_2$. На втором шаге алгоритма вычисляем лишь одно значение функции в точке, симметричной $\tilde{x}_2$ относительно середины отрезка [a_2, b_2]. Если же длина l_2 отрезка [a_2, b_2], полученного после первого шага алгоритма, оказалась меньше ε_*, то поиск прекращают и полагают $x_* \approx (a_2 + b_2) / 2$.

Пусть на k-м шаге, $k \geq 2$, последовательного поиска по методу золотого сечения выбран отрезок [a_k, b_k] и в нем точка $\tilde{x}_k$, осуществляющая золотое сечение этого отрезка. Значение $f(\tilde{x}_k)$ функции в этой точке уже вычислено на предыдущем шаге. Находим вторую точку $\hat{x}_k$ золотого сечения по формуле

$$\hat{x}_k = a_k + b_k - \tilde{x}_k$$

и вычисляем в ней значение функции. Если $\hat{x}_k < \tilde{x}_k$, то $x_{k1} = \hat{x}_k$ и $x_{k2} = \tilde{x}_k$, иначе $x_{k1} = \tilde{x}_k$ и $x_{k2} = \hat{x}_k$.

Пусть для определенности $\hat{x}_k < \tilde{x}_k$, (рис. 2.8) и $x_{k1} = \hat{x}_k$, $x_{k2} = \tilde{x}_k$. Если $f(x_{k1}) < f(x_{k2})$, то выбираем отрезок $[a_k, x_{k2}]$, т.е. полагаем $a_{k+1} = a_k$, $b_{k+1} = x_{k2}$, $\tilde{x}_{k+1} = x_{k1}$, иначе выбираем отрезок $[x_{k1}, b_k]$, т.е. полагаем $a_{k+1} = x_{k1}$, $b_{k+1} = b_k$, $\tilde{x}_{k+1} = x_{k2}$. Длину l_{k+1} нового отрезка $[a_{k+1}, b_{k+1}]$ сравниваем с ε_* и принимаем решение, продолжать поиск (при $l_{k+1} \geq \varepsilon_*$) или нет (при $l_{k+1} < \varepsilon_*$). В случае прекращения поиска полагаем $x_* \approx a_k + b_k / 2$.

Согласно описанию алгоритма, на первом шаге значение функции вычисляют в двух точках, а на каждом из последующих шагов вычисляют лишь одно значение функции. Поэтому после k шагов алгоритма значение функции будет вычислено в $N = k + 1$ точках. Поскольку после каждого шага интервал неопределенности уменьшается в r раз, то для длины l_{k+1} отрезка $[a_{k+1}, b_{k+1}]$ получаем $l_{k+1} = l_1 / r^k = 1 / r^k$, а зависимость l_N^z длины интервала неопределенности от количества N вычисленных значений функции выражается формулой

$$l_N^z = l_{k+1} = \frac{1}{r^k} = \frac{1}{r^{N-1}}. \qquad (2.10)$$

Алгоритмы методов золотого сечения и дихотомии аналогичны. Различие состоит лишь в том, что в методе дихотомии расстояние 2δ между внутренними точками x_{k1}, x_{k2} отрезка $[a_k, b_k]$ на каждом k-м шаге остается неизменным, а в методе золотого сечения оно зависит от номера шага поиска и уменьшается с уменьшением длины l_k отрезка по мере возрастания номера шага. Так в методе золотого сечения на k-м шаге поиска внутренними точками отрезка $[a_k, b_k]$ будут $x_{k1} = a_k + (1 + 1/r)l_k$ и $x_{k2} = a_k + l_k/r$, а расстояние между ними равно $x_{k2} - x_{k1} = (2/r - 1)l_k = (\sqrt{5} - 2)\ l_k \approx 0{,}236068 l_k$.

Метод Фибоначчи. Пусть при поиске точки $x_* \in [0, 1]$, в которой унимодальная на отрезке $[0, 1]$ функция $f(x)$ принимает наименьшее на этом отрезке значение, можно вычислить ее значения только в двух точках. Тогда предпочтение следует отдать методу дихотомии при $\delta << 1$, так как он позволит уменьшить интервал неопределенности почти вдвое, а метод золотого сечения – лишь в $r \approx 1{,}618$ раз. Сравнение (2.8) и (2.10) показывает, что при количестве вычисляемых значений функции $N \geq 4$ эффективность метода золотого сечения становится выше, чем метода дихотомии.

Однако при любом заданном общем числе $N > 2$ вычисляемых значений функции можно построить еще более эффективный метод, состоящий из $N - 1$ шагов. Он сочетает преимущество симметричного расположения внутренних точек x_{k1}, x_{k2} на отрезке $[a_k, b_k]$ относительно его середины, реализованное в методах дихотомии и золотого сечения, с возможностью на каждом шаге изменять отношение l_k/l_{k+1} длин сокращаемого и нового отрезков. Как показано при обсуждении метода золотого сечения, в случае выбора внутренних точек симметрично относительно середины отрезка для трех последовательных шагов этого метода выполняется соотношение

$$l_{k-1} = l_k + l_{k+1},\ k = 2, 3, \ldots \tag{2.11}$$

Построение алгоритма такого метода удобнее начать с последнего шага, но предварительно уточним задачу. Располагая возможностью вычислить в N точках $x_k \in [0, 1]$, $k = \overline{1, N}$, значения унимодальной на отрезке функции $f(x)$, необходимо как можно точнее, т.е. с наименее возможной длиной интервала неопределенности, отыскать точку x_* наименьшего значения этой функции на отрезке [0, 1].

При выполнении процедуры исключения отрезка на последнем, $(N - 1)$-м шаге имеем отрезок $[a_{N-1}, b_{N-1}]$ длиной l_{N-1} с двумя внутренними точками x_{N-1} и x_N, симметрично расположенными относительно середины отрезка на достаточно малом расстоянии 2δ друг от друга (рис. 2.9). В этих точках вычислены значения $f(x_{N-1})$ и $f(x_N)$ функции $f(x)$. Примем, что $f(x_N) < f(x_{N-1})$, тогда для нового отрезка $[a_N, b_N]$ длиной $l_N = l_{N-1}/2 + \delta$ внутренней будет точка x_N, а точка x_{N-1} совпадет с одним из его концов.

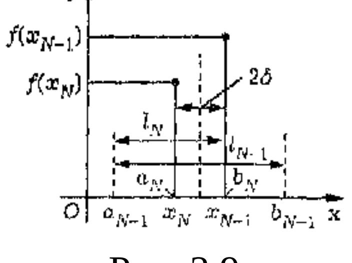


Рис. 2.9

В такой ситуации при выборе $x_* = x_N$ длина интервала неопределенности равна пока неизвестной длине l_N отрезка $[a_N, b_N]$. Через l_N

можно выразить длину $l_{N-1} = 2l_N - 2\delta$ отрезка $[a_{N-1}, b_{N-1}]$. Далее в соответствии с (2.11) получаем

$$l_{N-2} = l_{N-1} + l_N = 3l_N - 2\delta,\ l_{N-3} = l_{N-2} + l_{N-1} = 5l_N - 4\delta$$

$$l_{N-4} = l_{N-3} + l_{N-2} = 8l_N - 6\delta,\ l_{N-5} = l_{N-4} + l_{N-3} = 13l_N - 10\delta$$

и в общем виде

$$l_{N-K} = F_{K+2}l_N - 2F_K\delta,\ K = \overline{0, N-1}, \tag{2.12}$$

где коэффициенты F_m определены рекуррентным соотношением

$$F_m = F_{m-1} + F_{m-2},\ m = \overline{3, N-1},\ F_1 = F_2 = 1. \tag{2.13}$$

Так как при $K = N - 1$ длина $l_{N-K} = l_1 = 1$ отрезка [0, 1] известна, то из (2.12) можно найти длину интервала неопределенности

$$l_N^f = \frac{l_1}{F_{N+1}} + 2\delta\frac{F_{N-1}}{F_{N+1}}. \tag{2.14}$$

Существует алгоритм метода прямого поиска, удовлетворяющий соотношению (2.14).

Все коэффициенты F_m принадлежат множеству N натуральных чисел, и их называют ***числами Фибоначчи****. В табл. 2.1 представлены эти числа до номера $m = 25$.

Таблица 2.1

m	F_m	m	F_m	m	F_m	m	F_m	m	F_m
1	1	6	8	11	89	16	987	21	10946
2	1	7	13	12	144	17	1597	22	17711
3	2	8	21	13	233	18	2584	23	28657
4	3	9	34	14	377	19	4181	24	46368
5	5	10	55	15	610	20	6765	25	75025

Метод, использующий числа Фибоначчи для выбора длин отрезков l_k, а значит, и точек $x_k \in [0, 1]$, $k = \overline{1, N}$, в которых вычисляют значения минимизируемой функции, называют ***методом Фибоначчи*** (иногда – оптимальным последовательным поиском). Если на первом шаге поиска ($k = 1$, $K = N - 1$) интервал неопределенности имеет длину l_1, то в соответствии с (2.12) и (2.14) длина l_2 нового отрезка $[a_1, b_2]$ равна

$$l_2 = F_N l_N - 2\delta F_{N-2} = \frac{F_N}{F_{N+1}}l_1 + 2\delta\frac{F_N F_{N-1} - F_N F_{N-2}}{F_{N+1}} = \frac{F_N}{F_{N+1}}l_1 + (-1)^{N+1}\frac{2\delta}{F_{N+1}}.$$

Опишем алгоритм метода, пренебрегая малой величиной δ, т.е. принимая

$$\frac{l_1}{l_2} = \frac{F_{N+1}}{F_N}. \qquad (2.15)$$

Несложно проверить, что в этом случае выполнение процедуры исключения отрезка на последнем, $(N - 1)$-м шаге поиска приводит к совпадению внутренних точек x_{N-1} и x_N (рис. 2.9).

Отметим, что уже при $N = 11$ имеем $F_{12}/F_{11} = 144/89 \approx 1{,}617978$, а при $N = 21$ получаем $F_{22}/F_{21} = 17711/10946 \approx 1{,}618034$, что совпадает с отношением r золотого сечения с точностью до 10^{-6}. Таким образом, на первом шаге длина исходного отрезка уменьшается практически так же, как и в методе золотого сечения.

При $l_1 = 1$ из (2.15) находим $l_2 = F_N/F_{N+1}$. Таким образом, учитывая (2.13), заключаем, что на первом шаге выбор точек, симметричных относительно середины отрезка [0, 1], можно определить по формулам

$$x_1 = l_2 = \frac{F_N}{F_{N+1}}, \; x_2 = 1 - l_2 = 1 - \frac{F_N}{F_{N+1}} = \frac{F_{N-1}}{F_{N+1}}, \; x_2 < x_1,$$

причем расстояние между ними будет равно

$$d_1 = x_1 - x_2 = \frac{F_N}{F_{N+1}} - \frac{F_{N-1}}{F_{N+1}} = \frac{F_{N-2}}{F_{N+1}}.$$

После выполнения на этом шаге процедуры исключения отрезка одна из точек x_1, x_2 будет граничной точкой нового отрезка $[a_2, a_1]$, а другая – его внутренней точкой, которую обозначим x_k'. Вторая внутренняя точка на этом отрезке должна быть выбрана симметрично точке x_k' относительно его середины. Аналогично происходит выбор второй внутренней точки нового отрезка на всех последующих шагах поиска.

На k-м шаге в соответствии с равенством (2.12), в котором принимаем $K = N - k$, и равенством (2.14) длина отрезка $[a_k, b_k]$ равна $l_k = F_{N+2-k} / F_{N+1}$ и происходит ее уменьшение в $l_k / l_{k+1} = F_{N+2-k} / F_{N+1-k}$ раз. Если внутренние точки на этом отрезке обозначить α_k и β_k, то проведенные рассуждения позволяют написать следующие выражения:

$$\alpha_k = \alpha_k + \frac{F_{N-k}}{F_{N+1}}, \; \beta_k = \beta_k + \frac{F_{N+1-k}}{F_{N+1}}, \; \alpha_k < \beta_k, \; k = \overline{1, N-1}.$$

Подчеркнем, что реализация метода Фибоначчи предполагает априорное задание требуемого количества N вычисляемых значений функции (или количества шагов поиска). Этот параметр необходим для реализации первого шага алгоритма при выборе точек x_{11} и x_{12} деления отрезка $[a_1, b_1]$. Если параметр N по каким-либо причинам не может быть задан заранее, следует использовать другие методы, например дихотомии или золотого сечения.

2.5. Сравнение методов последовательного поиска

В качестве оценки скорости сходимости методов прямого поиска можно использовать скорость убывания длины интервала неопределенности в зависимости от числа n вычисленных значений минимизируемой функции в различных методах.

Из полученных результатов видно, что скорость сходимости метода Фибоначчи при больших значениях n немного выше, чем скорость сходимости метода золотого сечения. Метод золотого сечения качественно “лучше” метода дихотомии. Но скорость сходимости метода дихотомии при больших значениях n выше, чем скорость сходимости метода оптимального пассивного поиска.

Итак, метод золотого сечения уступает по скорости сходимости лучшему методу – методу Фибоначчи – примерно в 1,17 раза, но является более гибким, поскольку не требует выбора заранее определенного числа точек, в которых предстоит вычислить значения минимизируемой функции.

3. АЛГОРИТМЫ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

3.1. Алгоритмы решения задач без ограничений

В этом разделе представлен *градиентный* метод решения задач нелинейного программирования без ограничений на переменные. При этом для нахождения экстремальной точки используется градиент целевой функции.

3.1.1. Градиентный метод

В этом подразделе рассматривается метод решения задач нелинейного программирования с дважды непрерывно дифференцируемыми функциями. Основная идея метода заключается в построении последовательности точек, с учетом направления градиента исследуемой функции.

Одним из градиентных методов является изложенный ранее метод Ньютона – Рафсона, который ориентирован на решение систем уравнений. Здесь рассматривается еще один метод, который называется методом *наискорейшего подъема.* В русской математической литературе описываемый метод называется *методом наискорейшего спуска* и ориентирован на решение задач безусловной минимизации.

Согласно градиентному методу, вычисления завершаются при нахождении точки, в которой градиент функции равен нулю. Но это лишь необходимое условие для того, чтобы в данной точке находился оптимум. Чтобы удостовериться, что найдена именно точка оптимума, обычно привлекаются свойства вогнутости или выпуклости функции $f(x)$.

Рассмотрим задачу максимизации функции $f(x)$. Пусть X^0 – начальная точка, с которой начинается реализация метода, $\nabla f(X^k)$ – градиент функции f в k-й точке X^k. Идея метода сводится к определению в данной точке направления p, вдоль которого производная по направлению df/dp достигает своего максимума. Это достигается в том случае, когда последовательные точки X^k и X^{k+1} связаны соотношением

$$X^{k+1} = X^k + r^k \nabla f(X^k),$$

где r^k – параметр, называемый **длиной шага**.

Обычно значение параметра r^k выбирается из условия, чтобы в точке X^{k+1} наблюдалось максимальное увеличение значения целевой функции f. Другими словами, если функция $h(r)$ определяется соотношением

$$h(r) = f(X^k + r^k \nabla f(X^k)),$$

то r^k равен значению r, на котором функция $h(r)$ достигает максимума. Поскольку $h(r)$ является функцией одной переменной, для нахождения ее максимума можно применить метод прямого поиска экстремума, если, конечно, функция $h(r)$ строго одновершинная.

Описанная процедура заканчивается, когда две последовательные точки X^k и X^{k+1} близки. Это эквивалентно тому, что $r^k\nabla f(X^k) \approx 0$. Поскольку $r^k \neq 0$, последнее соотношение означает, что в точке X^k выполняется необходимое условие экстремума $\nabla f(X^k) = 0$.

Пример 3.1

Рассмотрим задачу максимизации функции

$$f(x_1, x_2) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2.$$

Функция $f(x_1, x_2)$ является квадратичной и нам известно, что ее абсолютный максимум достигается в точке $(x_1^*, x_2^*) = (1/3; 4/3)$. Решим эту задачу методом наискорейшего подъема. На рис. 3.1 изображена последовательность получаемых точек. Градиенты в двух последовательных итерационных точках с необходимостью являются ортогональными (перпендикулярными) друг к другу. Для данной функции градиент вычисляется по формуле

$$\nabla f(X) = (4 - 4x_1 - 2x_2, 6 - 2x_1 - 4x_2).$$

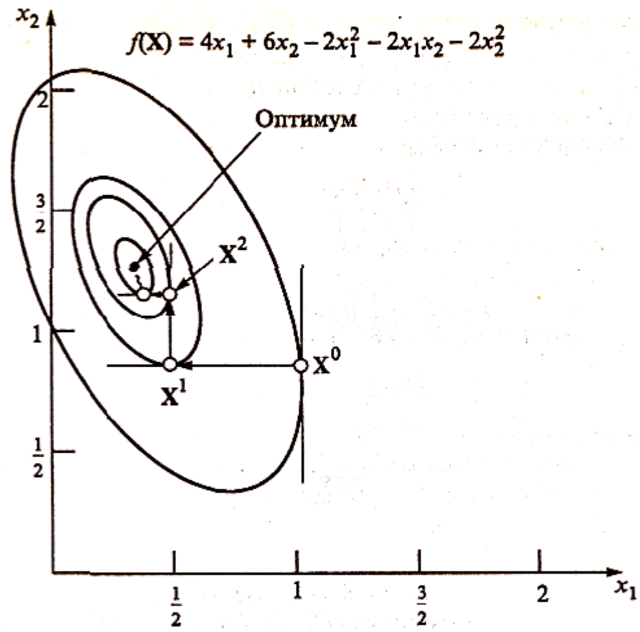


Рис. 3.1

Пусть начальной точкой является $X^0 = (1, 1)$.

Первая итерация

$$\nabla f(X^0) = (-2, 0).$$

Следующая точка X^1 находится на основании формулы

$$X = (1, 1) + r(-2, 0) = (1 - 2r, 1).$$

Отсюда получаем выражение для функции $h(r)$:

$$h(r) = f(1 - 2r, 1) = -2(1 - 2r)^2 + 2(1 - 2r) + 4.$$

Оптимальная длина шага r, при которой функция $h(r)$ достигает максимума, равна 1/4. Таким образом, мы нашли, что $X^1 = (1/2, 1)$.

Вторая итерация

$$\nabla f(X^1) = (0, 1).$$

Теперь следующая точка X^2 определяется выражением

$$X = \left(\tfrac{1}{2}, 1\right) + r(0,1) = \left(\tfrac{1}{2}, 1 + r\right).$$

Следовательно,

$$h(r) = -2(1 + r)^2 + 5(1 + r) + 3/2.$$

Отсюда получаем $r = 1/4$ и $X^2 = (1/2, 5/4)$.

Третья итерация

$$\nabla f(X^2) = (-1/2, 0).$$

Точка X^3 определяется выражением

$$X = \left(\frac{1}{2}, \frac{5}{4}\right) + r\left(-\frac{1}{2}, 0\right) = \left(\frac{1-r}{2}, \frac{5}{4}\right).$$

Следовательно,

$$h(r) = -\frac{1}{2}(1-r)^2 + \frac{3}{4}(1-r) + \frac{35}{8}.$$

Отсюда получаем $r = 1/4$ и $X^3 = (3/8, 5/4)$.

Четвертая итерация

$$\nabla f(X^3) = (0, 1/4).$$

Точка X^4 определяется выражением

$$X = \left(\frac{3}{8}, \frac{5}{4}\right) + r\left(0, \frac{1}{4}\right) = \left(\frac{3}{8}, \frac{5+r}{4}\right).$$

Поэтому

$$h(r) = -\left(\frac{1}{8}\right)(5+r)^2 + \left(\frac{21}{16}\right)(5+r) + \frac{39}{32}.$$

Отсюда имеем $r = 1/4$ и $X^4 = (3/8, 21/16)$.

Пятая итерация

$$\nabla f(X^4) = (-1/8, 0).$$

Точка X^5 определяется выражением

$$X = \left(\frac{3}{8}, \frac{21}{16}\right) + r\left(-\frac{1}{8}, 0\right) = \left(\frac{3-r}{8}, \frac{21}{16}\right).$$

Следовательно,

$$h(r) = -\left(\frac{1}{32}\right)(3-r)^2 + \left(\frac{11}{64}\right)(3-r) + \frac{567}{128}.$$

Отсюда получаем $r = 1/4$ и $X^5 = (11/32, 21/16)$.

Шестая итерация

$$\nabla f(X^5) = (0, 1/16).$$

Так как $\nabla f(X^5) \approx 0$, вычисления можно закончить. Получена *приближенная* точка максимума $X^5 = (0{,}3437;\ 1{,}3125)$. Точный максимум достигается в точке $X^* = (0{,}3333;\ 1{,}3333)$.

3.2. Алгоритмы решения задач с ограничениями

Общая задача нелинейного программирования с ограничениями записывается в следующем виде:

максимизировать (или минимизировать) $z = f(X)$

при ограничениях

$$g(X) \le 0.$$

Условия неотрицательности переменных $X \ge 0$ составляют часть заданных ограничений общего вида. Предполагается также, что, по крайней мере, одна из функций $f(x)$ или $g(X)$ является нелинейной и все функции непрерывно дифференцируемы.

Универсальных алгоритмов решения задач нелинейного программирования не существует, и связано это, главным образом, с разнообразием нелинейных функций. Возможно, наиболее общим результатом, имеющим отношение к рассматриваемым задачам, являются условия Куна – Таккера. Как показано ранее, эти условия являются лишь необходимыми для существования экстремума, за исключением, когда функции $f(x)$ и $g(X)$ являются выпуклыми или вогнутыми.

Существует ряд алгоритмов решения задач нелинейного программирования, которые условно можно разделить на *непрямые* и *прямые*. В непрямых методах решение задачи нелинейного программирования сводится к решению одной или нескольких *линейных* задач, порожденных исходной. Прямые методы непосредственно имеют дело с исходной нелинейной задачей и позволяют построить последовательность точек, сходящуюся к точке экстремума.

Непрямые методы представлены алгоритмами сепарабельного, квадратичного, геометрического и стохастического программирования. К числу прямых методов относятся метод линейных комбинаций и метод последовательной безусловной максимизации, изложенный далее кратко.

3.2.1. Сепарабельное программирование

Функция $f(x_1, x_2, \ldots, x_n)$ называется **сепарабельной** (разделимой), если она представляется в виде суммы и функций одной переменной $f_1(x_1)$, $f_2(x_2)$, …, $f_n(x_n)$, т.е.

$$f(x_1, x_2, \ldots, x_n) = f(x_1) + f(x_2) + \ldots + f(x_n).$$

К примеру, линейная функция

$$h(x_1, x_2, \ldots, x_n) = a_1x_1 + a_2x_2 + \ldots + a_nx_n$$

(здесь a_i, $i = 1, 2, \ldots, n$ – константы) является сепарабельной. Функция же

$$h(x_1, x_2, x_3) = x_1^2 + x_1\sin(x_2 + x_3) + x_2 e^{x_3}$$

таковой не является.

Некоторые нелинейные функции сепарабельными непосредственно не являются, однако могут быть приведены к такому виду путем соответствующих подстановок. Рассмотрим, к примеру, задачу максимизации функции $z = x_1x_2$. Если ввести обозначение $y = x_1x_2$, то $ln\, y = ln\, x_1 + ln\, x_2$, и задача принимает следующий вид:

Максимизировать $z = y$

при ограничении

$$ln\, y = ln\, x_1 + ln\, x_2,$$

т.е. она является сепарабельной. При такой замене предполагается, что переменные x_1 и x_2 принимают положительные значения, иначе логарифмическая функция не определена.

Если переменные x_1 и x_2 принимают нулевые значения (т.е. $x_1, x_2 \geq 0$), можно поступить следующим образом. Пусть δ_1 и δ_2 – положительные константы, введем новые переменные $w_1 = x_1 + \delta_1$ и $w_2 = x_2 + \delta_2$.

Эти переменные принимают только положительные значения. Теперь имеем

$$x_1x_2 = w_1w_2 - \delta_2w_1 - \delta_1w_2 + \delta_1\delta_2.$$

Пусть $y = w_1w_2$, тогда исходная задача эквивалентна следующей.

Максимизировать $z = y - \delta_2w_1 - \delta_1w_2 + \delta_1\delta_2$

при ограничениях

$$ln\, y = ln\, w_1 + ln\, w_2,\ w_1 \geq \delta_1,\ w_2 \geq \delta_2.$$

В этой задаче все функции сепарабельные.

Примерами других функций, которые в результате замены переменных приводятся к сепарабельным, могут служить $e^{x_1+x_2}$ и $x_1^{x_2}$. В этих случаях для реализации свойства сепарабельности следует применить соответствующий вариант описанной выше процедуры.

В сепарабельном программировании рассматриваются задачи нелинейного программирования, в которых как целевая функция, так и функции ограничений являются сепарабельными. В этом подразделе рассматриваются методы приближенного решения задачи сепарабельного программирования, основанные на линейной аппроксимации функций и на симплекс-методе линейного программирования.

Функцию одной переменной $f(x)$ можно аппроксимировать кусочно-линейной функцией с помощью методов частично-целочисленного программирования. Пусть требуется аппроксимировать функцию $f(x)$ на интервале $[a, b]$. Обозначим через a_k, $k = 1, 2, \ldots, K$, k-ю точку разбиения интервала $[a, b]$, причем $a = a_1 < a_2 < \ldots < a_k = b$. Тогда функция $f(x)$ аппроксимируется следующей кусочно-линейной функцией:

$$f(x) \approx \sum_{k=1}^{k} f(a_k)t_k\,,\ \ x = \sum_{k=1}^{k} a_k t_k\,,$$

где t_k – неотрицательный весовой коэффициент, связанный с k-й точкой разбиения интервала. Весовые коэффициенты удовлетворяют условию

$$\sum_{k=1}^{k} t_k = 1.$$

Методы частично-целочисленного программирования гарантируют корректность такой аппроксимации. В частности, аппроксимация является обоснованной в следующих случаях.

1. Если не больше двух весовых коэффициентов t_k имеют положительные значения.

2. Если значение t_k больше нуля, то положительным может оказаться лишь один из смежных весовых коэффициентов t_{k+1} или t_{k-1}.

Чтобы показать, как выполняются перечисленные условия, рассмотрим общую задачу сепарабельного программирования.

Максимизировать (минимизировать) $z = \sum\limits_{i=1}^{n} f_i(x_i)$

при ограничениях

$$\sum_{i=1}^{n} g_i^j(x_i) \le b_j,\ j = 1, 2, \ldots, m.$$

Эту задачу можно привести к задаче частично-целочисленного программирования следующим образом. Обозначим через K_i число точек разбиения для i-й переменной x_i, а через a_i^k – k-ю точку разбиения. Пусть t_i^k – весовой коэффициент, ассоциируемый с k-й точкой разбиения для i-й переменной. Тогда эквивалентная задача частично-целочисленного программирования имеет следующий вид:

Максимизировать (или минимизировать) $z = \sum_{i=1}^{n} \sum_{k=1}^{K_i} f_i(a_i^k) t_i^k$

при ограничениях

$$\sum_{i=1}^{n} \sum_{k=1}^{K_i} g_i^j(a_i^k) t_i^k \le b_j,\ j = 1, 2, \ldots, m,$$

$$0 \le t_i^1 \le y_i^1,\ 0 \le t_i^k \le y_i^{k-1},\ k = 2, 3, \ldots, K_i - 1,$$

$$0 \le t_i^{Ki} \le y_i^{Ki-1},$$

$$\sum_{k=1}^{K_i - 1} y_i^k = 1,$$

$$\sum_{k=1}^{K_i} t_i^k = 1,$$

$$y_i^k = 0 \text{ или } 1,\ k = 1, 2, \ldots, K_i,\ i = 1, 2, \ldots, n.$$

В аппроксимирующей задаче переменными являются t_i^k и y_i^k.

Данное представление свидетельствует о том, что в принципе любая задача сепарабельного программирования может быть решена методами частично-целочисленного программирования. Трудность такого подхода к решению задачи связана с быстрым ростом числа ограничений при увеличении количества точек разбиения. В частности, проблематичной является вычислительная реализация процедуры, поскольку нет эффективных компьютерных программ для решения задач частично-целочисленного программирования большой размерности.

Для решения аппроксимирующей задачи можно использовать обычный симплекс-метод, дополненный правилом ограниченного ввода в базис. В этом случае игнорируются вспомогательные ограничения, содержащие y_i^k. Согласно данному правилу, в базисе может находиться не более двух положительных весовых коэффициентов t_i^k. Более того, два коэффициента t_i^k могут быть положительными лишь тогда, когда они являются смежными.

Таким образом, строгое условие оптимальности симплекс-метода только тогда используется для выбора переменной t_i^k, подлежащей введению в число базисных, когда она удовлетворяет указанным выше требованиям. В противном случае анализ разностей $(z_i^k - c_i^k)$ позволяет выбрать следующую переменную t_i^k, которая может быть сделана базисной. Процесс повторяется до тех пор, пока не будет выполнен критерий оптимальности или же установлена невозможность введения в базис новой переменной t_i^k без нарушения правила ограниченного ввода. В обоих случаях последняя симплексная таблица дает приближенное оптимальное решение исходной задачи.

Метод частично-целочисленного программирования позволяет найти глобальный экстремум аппроксимирующей задачи, тогда как симплекс-метод с учетом правила ограниченного ввода в базис может гарантировать нахождение лишь локального оптимума. Кроме того, приближенное решение, полученное любым из двух упомянутых методов, может быть недопустимым для исходной задачи, поскольку при решении аппроксимирующей задачи могут быть обнаружены дополнительные экстремальные точки, которые для исходной задачи экстремальными не являются. Это зависит от точности линейной аппроксимации исходной задачи.

Пример 3.2

Рассмотрим следующую задачу.

Максимизировать $z = x_1 + x_2^4$

при ограничениях

$$3x_1 + 2x_2^2 \le 9,$$

$$x_1, x_2 \ge 0.$$

Опишем применение симплекс-метода совместно с правилом ограниченного ввода в базис.

Точное оптимальное решение этой задачи находится непосредственной проверкой: $x_1 = 0$, $x_2 = 0$ и $z^* = 20{,}25$. Чтобы продемонстрировать использование метода аппроксимации, рассмотрим отдельные функции:

$$f_1(x_1) = x_1,$$

$$f_2(x_2) = x_2^4,$$

$$g_1^1(x_1) = 3x_1,$$

$$g_1^2(x_2) = 2x_2^2.$$

Функции $f_1(x_1)$ и $g_1^1(x_1)$ остаются в исходном виде, поскольку они уже являются линейными. В этом случае x_1 рассматривается как одна из переменных. Для функций $f_2(x_2)$ и $g_1^2(x_2)$ полагаем, что количество точек разбиения равно четырем ($K_2 = 4$). Так как значение x_2 не может превышать 3, отсюда вытекают данные, приведенные в следующей таблице.

k	a_2^k	$f_2(a_2^k)$	$g_1^2(a_2^k)$
1	0	0	0
2	1	1	2
3	2	16	8
4	3	81	18

Отсюда получаем

$$f_2(x_2) \approx t_2^1\, f_2(a_2^1) + t_2^2\, f_2(a_2^2) + t_2^3\, f_2(a_2^3) + t_2^4\, f_2(a_2^4) = 0 \times t_2^1 + 1 \times t_2^2 + 16 \times t_2^3 + 81 \times t_2^4 = t_2^2 + 16t_2^3 + 81t_2^4.$$

Аналогично

$$g_1^2(x_2) \approx 2t_2^2 + 8t_2^3 + 18t_2^4.$$

Следовательно, аппроксимирующая задача принимает вид

Максимизировать $z = x_1 + t_2^2 + 16t_2^3 + 81t_2^4$

при ограничениях

$$3x_1 + 2t_2^2 + 8t_2^3 + 18t_2^4 \le 9,$$
$$t_2^1 + t_2^2 + t_2^3 + t_2^4 = 1,$$
$$t_2^k \ge 0,\ k = 1, 2, 3, 4.$$
$$x_1 \ge 0.$$

Исходная симплекс-таблица (в которой осуществлена перестановка столбцов для получения начального решения) выглядит следующим образом.

Базис	x_1	t_2^2	t_2^3	t_2^4	S_1	t_2^1	Решение
z	–1	–1	–16	–81	0	0	0
S_1	3	2	8	18	1	0	9
t_2^1	0	1	1	1	0	1	1

Здесь S_1 (≥ 0) – дополнительная (остаточная) переменная. (В этой задаче начальное решение является очевидным. В общем же случае для его получения могут понадобиться искусственные переменные.)

Коэффициенты z-строки указывают на то, что в число базисных необходимо ввести переменную t_2^4. Поскольку при этом переменная t_2^1 уже входит в базисные, в силу правила ограниченного ввода в базис, она должна быть выведена из числа базисных перед тем, как базисной станет переменная t_2^4. Вместе с тем, согласно условию допустимости, при введении в базисные переменной t_2^4 переменная S_1 должна выводиться из базиса. Это значит, что переменную t_2^4 нельзя сделать базисной. Следующей подходящей для введения в базисные является переменная t_2^3. Снова при этом переменная t_2^1 должна быть выведена из числа базисных. Условие допустимости на этот раз позволяет вывести из числа базисных переменную t_2^1, что и требуется. Таким образом, получаем новую симплекс-таблицу.

Базис	x_1	t_2^2	t_2^3	t_2^4	S_1	t_2^1	Решение
z	–1	15	0	–65	0	16	16
S_1	3	–6	0	10	1	–8	1
t_2^3	0	1	1	1	0	1	1

Очевидно, что базисной следует сделать переменную t_2^4. То, что переменная t_2^3 уже является базисной, не противоречит правилу ограниченного ввода в базис. В соответствии с симплекс-методом переменная S_1 исключается из числа базисных. Получаем следующую симплекс-таблицу.

Базис	x_1	t_2^2	t_2^3	t_2^4	S_1	t_2^1	Решение
z	37/2	–24	0	0	13/2	–36	45/2
t_2^4	3/10	–6/10	0	1	1/10	–8/10	1/10
t_2^3	–3/10	16/10	1	0	–1/10	18/10	9/10

Из этой таблицы следует, что в базис могут вводиться переменные t_2^1 и t_2^2. Переменная t_2^1 не может быть базисной, поскольку не является смежной с переменными t_2^3 и t_2^4, которые уже находятся среди базисных. Далее, переменную t_2^2 также нельзя сделать базисной, поскольку t_2^4 нельзя исключить из числа базисных.

Поэтому процедура решения задачи на этом завершается, и полученное решение является наилучшим допустимым решением аппроксимирующей задачи.

Воспользуемся найденными значениями $t_2^3 = 9/10$ и $t_2^4 = 1/10$, чтобы выразить полученное решение через переменные x_1 и x_2. Получаем

$$x_2 \approx 2t_2^3 + 3t_2^4 = 2\left(\frac{9}{10}\right) + 3\left(\frac{1}{10}\right) = 2{,}1$$

и $x_1 = 0$, $z = 22{,}5$. Полученное приближенное оптимальное значение для $x_2 = 2{,}1$ мало отличается от истинного оптимального значения, равного 2,12.

Сепарабельное выпуклое программирование. Задача выпуклого сепарабельного программирования имеет место в том случае, когда функции $g_i^j(x_i)$ являются выпуклыми, обеспечивая, таким образом, выпуклость области допустимых решений. Кроме того, считаем, что функции $f_i(x_i)$ являются выпуклыми в задаче минимизации и вогнутыми в задаче максимизации. При этих условиях можно использовать следующий упрощенный метод аппроксимации.

Рассмотрим задачу минимизации. Пусть функция $f_i(x_i)$ имеет такой вид, как показано на рис. 3.2. Обозначим точки разбиения для функции $f_i(x_i)$ через $x_i = a_{ki}$, $k = 0, 1, \ldots, K_i$.

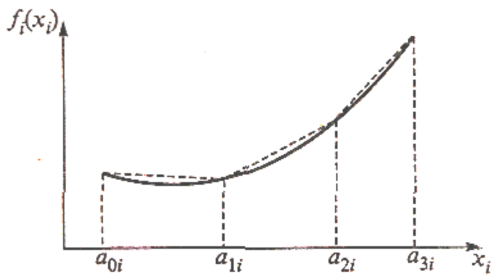


Рис. 3.2

Пусть x_{ki} – величина изменения переменной x_i на интервале (a_{k-1}, a_{ki}), $k = 1, 2, \ldots, K_i$, а ρ_{ki} – тангенс угла наклона линейного сегмента на том же интервале.

Тогда имеем

$$f_i(x_i) \approx \sum_{k=1}^{K_i} \rho_{ki} x_{ki} + f_i(a_{0i}), \ x_i = \sum_{k=1}^{K_i} x_{ki} .$$

Здесь предполагается, что $0 \le x_{ki} \le a_{ki} - a_{k-1}$, $k = 1, 2, \ldots, K_i$.

Так как функция $f_i(x_i)$ выпуклая, то $\rho_{1i} < \rho_{2i} < \rho_{k,i}$. Это означает, что в задаче минимизации при $p < q$ большее влияние на значение целевой функции оказывает переменная x_{pi} чем x_{qi}. Следовательно, переменная x_{pi} всегда будет входить в решение до того, как в него войдет x_{qi}. При этом значения каждой переменной x_{ki} должны быть ограничены сверху величиной $(a_{ki} - a_{k-1,\, i})$.

Выпуклые функции ограничений $g_i^j(x_i)$ аппроксимируются аналогичным образом.

Пусть $\rho_{k_i}^j$ – тангенс угла наклона k-го линейного сегмента, соответствующего функции $g_i^j(x_i)$. Тогда функция $g_i^j(x_i)$ аппроксимируется формулой

$$g_i^j(x_i) \approx \sum_{k=1}^{K_i} \rho_{ki}^j x_{ki} + g_i^j(a_{0i}).$$

Следовательно, рассматриваемая задача принимает вид

Минимизировать $z = \sum_{i=1}^{n} \left(\sum_{k=1}^{K_i} \rho_{ki}^j x_{ki} + g_i^j(a_{0i}) \right)$,

при ограничениях

$$\sum_{i=1}^{n} \left(\sum_{k=1}^{K_i} \rho_{ki}^j x_{ki} + g_i^j(a_{0i}) \right) \le b_j,\ j = 1, 2, \ldots, m,$$

$$0 \le x_{ki} \le a_{ki} - a_{k-1,i},\ k = 1, 2, \ldots, K_i,\ i = 1, 2, \ldots, n,$$

где

$$\rho_{ki} = \frac{f_i(a_{ki}) - f_i(a_{k-1,i})}{a_{ki} - a_{k-1,i}},$$

$$\rho_{ki}^j = \frac{g_i^j(a_{ki}) - g_i^j(a_{k-1,i})}{a_{ki} - a_{k-1,i}}.$$

Задача максимизации преобразуется аналогично. В этом случае $\rho_{1i} > \rho_{2i} > \rho_{k,i}$, откуда следует, что при $p < q$ переменная x_{pi} всегда будет входить в решение раньше x_{qi}.

Полученную задачу можно решить симплекс-методом, предназначенным для решения задач с ограниченными сверху переменными. При этом исчезает необходимость в соблюдении правила ограниченного ввода в базис, поскольку выпуклость (вогнутость) функций гарантирует надлежащий выбор переменных.

Пример 3.3

Рассмотрим задачу.

Минимизировать $z = x_1^2 + x_2^2 + 5$

при ограничениях

$$3x_1^4 + x_2 \le 243,$$

$$x_1 + 2x_2^2 \le 32,$$

$$x_1, x_2 \ge 0.$$

Отдельными (*сепарабельными*) функциями этой задачи являются

$$f_1(x_1) = x_1^2,\ f_2(x_2) = x_2^2 + 5,$$

$$g_1^1(x_1) = 3x_1^4,\ g_2^1(x_2) = x_2,$$

$$g_1^2(x_1) = x_1,\ g_2^2(x_2) = 2x_2^2.$$

Эти функции выпуклые, что и требуется в задачах минимизации.

Область допустимых значений переменных x_1 и x_2, как следует из ограничений задачи, определяется неравенствами $0 \le x_1 \le 3$ и $0 \le x_2 \le 4$. Разбиваем эти интервалы изменения переменных x_1 и x_2. Например, при $K_1 = 3$ и $K_2 = 4$. Значения $a_{01} = a_{02} = 0$ тангенсов углов наклона, соответствующих отдельным функциям рассматриваемой задачи, приведены в следующих таблицах.

Для $i = 1$ имеем

k	a_{k1}	ρ_{k1}	ρ^1_{k1}	ρ^2_{k1}	x_{k1}
0	0	–	–	–	–
1	1	1	3	1	x_{11}
2	2	3	45	1	x_{21}
3	3	5	195	1	x_{31}

Для $i = 2$ имеем

k	a_{k2}	ρ_{k2}	ρ^1_{k2}	ρ^2_{k2}	x_{k2}
0	0	–	–	–	–
1	1	1	1	2	x_{12}
2	2	3	1	6	x_{22}
3	3	5	1	10	x_{32}
4	4	7	1	14	x_{42}

Теперь задача принимает следующий вид.

Минимизировать $z \approx x_{11} + 3x_{21} + 5x_{31} + x_{12} + 3x_{22} + 5x_{32} + 7x_{42} + 5$
при ограничениях

$$3x_{11} + 45x_{21} + 195x_{31} + x_{12} + x_{22} + x_{32} + x_{42} \le 243,$$

$$x_{11} + x_{21} + x_{31} + 2x_{12} + 6x_{22} + 10x_{32} + 14x_{42} \le 32,$$

$$0 \le x_{k1} \le 1,\ k = 1, 2, 3,$$

$$0 \le x_{k2} \le 1,\ k = 1, 2, 3, 4.$$

Пусть x^*_{k1} и x^*_{k2} – оптимальные значения соответствующих переменных, полученные с помощью описанного выше варианта симплекс-метода. Тогда оптимальные значения переменных x_1 и x_2 вычисляются по формулам

$$x^*_1 = \sum_{k=1}^{3} x^*_{k1},\ x^*_2 = \sum_{k=1}^{4} x^*_{k2}.$$

3.2.2. Квадратичное программирование

Задача квадратичного программирования имеет следующий вид.

Максимизировать (или минимизировать)

$$z = CX + X^T DX$$

при ограничениях

$$AX \leq \mathrm{b},\ X \geq 0,$$

где

$$X = (x_1, x_2, \ldots, x_n)^T,$$

$$C = (c_1, c_2, \ldots, c_n),$$

$$b = (b_1, b_2, \ldots, b_m)^T,$$

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \cdots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix},$$

$$D = \begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \cdots & \vdots \\ d_{n1} & \cdots & d_{nn} \end{bmatrix}.$$

Функция $X^T DX$, где D – симметрическая матрица, является квадратичной формой. Матрица D будет отрицательно определенной в задаче максимизации и положительно определенной – в задаче минимизации. Это означает, что функция z является строго выпуклой по переменным X в случае задачи минимизации и строго вогнутой – в задаче максимизации. Ограничения в этой задаче предполагаются линейными, что гарантирует выпуклость области допустимых решений.

Решение сформулированной задачи основано на использовании необходимых условий Куна–Таккера. Так как целевая функция z строго выпуклая (или вогнутая) и область допустимых решений задачи является выпуклым множеством, эти условия также достаточны для установления наличия глобального оптимума.

Задачу квадратичного программирования рассмотрим для случая, когда целевая функция подлежит максимизации. Изменение формулировки задачи при минимизации целевой функции является тривиальным. Общая постановка задачи имеет следующий вид.

Максимизировать $z = CX + X^T DX$

при ограничениях

$$\mathrm{G(X)} = \begin{bmatrix} A \\ -I \end{bmatrix} X - \begin{bmatrix} b \\ 0 \end{bmatrix} \le 0.$$

Обозначим через $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T$ и $U = (\mu_1, \mu_2, \ldots, \mu_n)^T$ множители Лагранжа, соответствующие ограничениям $AX - b \le 0$ и $-X \le 0$. Применяя условия Куна–Таккера, получаем

$$\lambda \ge 0,\ U \ge 0,$$

$$\nabla z - (\lambda^T, U^T)\ \nabla \mathrm{G(X)} = 0,$$

$$\lambda\left(b_i - \sum_{j=1}^{n} a_{ij} x_j\right) = 0,\ i = 1, 2, \ldots, m,$$

$$\mu_j x_j = 0,\ j = 1, 2, \ldots, n,$$

$$AX \le b,\ -X \le 0.$$

Отсюда имеем

$$\nabla z = C + 2X^T D,$$

$$\nabla \mathrm{G(X)} = \begin{bmatrix} A \\ -I \end{bmatrix}.$$

Обозначим через $S = b - AX \ge 0$ вектор дополнительных (остаточных) переменных. Тогда приведенные выше условия принимают следующий вид:

$$-2X^T D + \lambda^T A - U^T = C,$$

$$AX + S = b,$$

$$\mu_j x_j = 0 = \lambda_i S_i,\ \text{для всех } i \text{ и } j,$$

$$\lambda, U, X, S \ge 0.$$

Так как $D^T = D$, в результате транспонирования первой системы уравнений получим

$$-2DX + A^T \lambda - U = C^T.$$

Следовательно, необходимые условия могут быть записаны в виде

$$\begin{bmatrix} -2D & A^T & -I & 0 \\ A & 0 & 0 & I \end{bmatrix} \begin{bmatrix} X \\ \lambda \\ U \\ S \end{bmatrix} = \begin{bmatrix} C^T \\ b \end{bmatrix},$$

$$\mu_j x_j = 0 = \lambda_i S_i,\ \text{для всех } i \text{ и } j,$$

$$\lambda, U, X, S \ge 0.$$

Все уравнения, за исключением $\mu_j x_j = 0 = \lambda_i S_i$, являются линейными относительно переменных X, λ, U и S. Следовательно, исходная задача сводится к нахождению решения системы линейных уравнений, удовлетворяющего дополнительным условиям $\mu_j x_j = 0 = \lambda_i S_i$. Поскольку функция z строго вогнутая, а область допустимых решений представляет собой выпуклое множество, допустимое решение, удовлетворяющее всем этим условиям, должно быть единственным и оптимальным.

Решение рассматриваемой системы находится путем реализации этапа I двухэтапного метода. При этом единственным ограничением является необходимость удовлетворения условий $\mu_j x_j = 0 = \lambda_i S_i$. Выполнение этого условия означает, что если переменная λ_i в базисном решении принимает *положительное значение*, то переменная S_i не может быть базисной и принимать положительное значение. Аналогично переменные μ_j и x_j не могут одновременно принимать положительные значения. Это обстоятельство подобно правилу *ограниченного ввода в базис*, которое было использовано в предыдущем подразделе. Этап 1 завершается обращением в нуль всех искусственных переменных только в том случае, когда исходная задача имеет непустое множество допустимых решений.

Пример 3.4

Рассмотрим следующую задачу.

Максимизировать $z = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$

при ограничениях

$$x_1 + 2x_2 \le 2,$$

$$x_1, x_2 \ge 0.$$

Эту задачу можно записать в матричной форме.

Максимизировать $z = (4,6)\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + (x_1, x_2)\begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

при ограничениях

$$(1, 2)\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le 2,$$

$$x_1, x_2 \ge 0.$$

Условия Куна–Таккера принимают следующий вид:

$$\begin{bmatrix} 4 & 2 & 1 & -1 & 0 & 0 \\ 2 & 4 & 2 & 0 & -1 & 0 \\ 1 & 2 & 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \lambda_1 \\ \mu_1 \\ \mu_2 \\ S_1 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 2 \end{bmatrix}.$$

Начальная симплекс-таблица для этапа I строится путем введения искусственных переменных R_1 и R_2. Имеем

Базис	x_1	x_2	λ_1	μ_1	μ_2	R_1	R_2	S_1	Решение
r	6	6	3	–1	–1	0	0	0	10
R_1	4	2	1	–1	0	1	0	0	4
R_2	2	4	2	0	–1	0	1	0	6
S_1	1	2	0	0	0	0	0	1	2

Первая итерация

Поскольку $\mu_1 = 0$, вводимой в базис переменной может быть x_1, при этом из числа базисных исключается переменная R_1. Получаем следующую симплекс-таблицу.

Базис	x_1	x_2	λ_1	μ_1	μ_2	R_1	R_2	S_1	Решение
r	0	3	3/2	1/2	–1	–3/2	0	0	4
x_1	1	1/2	1/4	–1/4	0	1/4	0	0	1
R_2	0	3	3/2	1/2	–1	–1/2	1	0	4
S_1	0	3/2	–1/4	1/4	0	–1/4	0	1	1

Вторая итерация

Так как $\mu_2 = 0$, вводимой в базис переменной является x_2. В результате приходим к следующей таблице.

Базис	x_1	x_2	λ_1	μ_1	μ_2	R_1	R_2	S_1	Решение
r	0	0	2	0	–1	–1	0	–2	2
x_1	1	0	1/3	–1/3	0	1/3	0	–1/3	2/3
R_2	0	0	2	0	–1	0	1	–2	2
x_2	0	1	–1/6	1/6	0	–1/6	0	2/3	2/3

Третья итерация

Так как $S_1 = 0$, в число базисных можно ввести переменную λ_1. В результате получаем симплекс-таблицу.

Базис	x_1	x_2	λ_1	μ_1	μ_2	R_1	R_2	S_1	Решение
r	0	0	0	0	0	–1	–1	0	0
x_1	1	0	0	–1/3	1/6	1/3	–1/6	0	1/3
λ_1	0	0	1	0	–1/2	0	1/2	–1	1
x_2	0	1	0	1/6	–1/12	–1/6	1/12	1/2	5/6

Последняя таблица дает оптимальное решение рассматриваемой на этапе 1 задачи. Так как $r = 0$, полученное решение $x_1 = 1/3$, $x_2 = 5/6$ является допустимым. Оптимальное значение z вычисляется подстановкой полученного решения в выражение для целевой функции исходной задачи и равняется 4,16.

3.2.3. Геометрическое программирование

В *геометрическом программировании* рассматриваются задачи нелинейного программирования специального вида. Подход геометрического программирования позволяет находить решение исходной задачи с помощью соответствующей двойственной задачи (которая будет сформулирована ниже).

Методами геометрического программирования решаются нелинейные задачи, в которых как целевая функция, так и функции ограничений имеют следующий вид:

$$z = f(x) = \sum_{j=1}^{N} U_j ,$$

где

$$U_j = c_j \prod_{i=1}^{n} x_i^{a_{ij}} , \; j = 1, 2, \ldots, N.$$

Предполагается, что здесь все $c_j > 0$, а число N является конечным. Показатели степени a_{ij} по знаку не ограничены. Функция $f(X)$ имеет вид полинома, если не учитывать того, что показатели степени a_{ij} могут принимать отрицательные значения. По этой причине, а также в связи с тем, что все коэффициенты $c_j > 0$, функция $f(X)$ называется **позиномом**.

В этом подразделе будут рассмотрены задачи геометрического программирования без ограничений. Исследование задач геометрического программирования с ограничениями выходит за рамки разд. 3.

Рассмотрим задачу минимизации позиномиальной функции $f(X)$. Будем называть эту задачу прямой. Предполагается, что переменные x_i принимают строго положительные значения, т.е. область $x_i \le 0$ недопустима. Далее будет показано, что требование $x_i \ne 0$ является существенным при получении основных результатов.

Первые частные производные функции z в точке минимума должны обращаться в нуль. Следовательно,

$$\frac{\partial z}{\partial x_k} = \sum_{j=1}^{N} \frac{\partial U_j}{\partial x_k} = \sum_{j=1}^{N} c_j a_{kj} (x_k)^{a_{kj}-1} \prod_{i \ne k} (x_i)^{a_{ij}} = 0 , \; k = 1, 2, \ldots, n.$$

Поскольку по предположению все $x_k > 0$, имеем

$$\frac{\partial z}{\partial x_k} = 0 = \frac{1}{x_k} \sum_{j=1}^{n} a_{kj} U_j , \; k = 1, 2, \ldots, n.$$

Пусть z^* – минимальное значение функции z. Очевидно, что $z^* > 0$, так как z – позином и все $x_k > 0$. Обозначим

$$y_j = \frac{U_j^*}{z^*}.$$

Из определения следует, что $y_j > 0$ и $\sum_{j=1}^{N} y_j = 1$. Величина y_j характеризует относительный вклад j-го слагаемого U_j в оптимальное значение целевой функции z^*. Необходимые условия экстремума теперь можно записать в следующем виде:

$$\sum_{j=1}^{n} a_{kj} y_j = 0, \; k = 1, 2, \ldots, n,$$

$$\sum_{j=1}^{N} y_j = 1, \; y_j > 0 \text{ для всех } j.$$

Эти уравнения, именуемые **условиями ортогональности и нормировки**, определяют единственное решение для y_j при условии, что все уравнения независимы и $n + 1 = N$. Задача усложняется при $N > n + 1$, так как в этом случае решение для y_j не является единственным. Ниже показано, что и в этом случае существует возможность однозначно определить y_j для минимизации функции z.

Пусть y_j^* – элементы единственного решения представленной выше системы уравнений. Тогда величины z^* и x_j^*, $i = 1, 2, \ldots, n$, определяются следующим образом. Рассмотрим величину

$$z^* = (z^*)^{\sum_{j^*=1}^{N} y_j^*}.$$

Поскольку $z^* = \dfrac{U_j^*}{y_j^*}$, отсюда следует, что

$$z^* = \left(\frac{U_1^*}{y_1^*}\right)^{y_1^*} \left(\frac{U_2^*}{y_2^*}\right)^{y_2^*} \cdots \left(\frac{U_N^*}{y_N^*}\right)^{y_N^*} = \left\{\prod_{j=1}^{N}\left(\frac{c_j}{y_j^*}\right)^{y_j^*}\right\} \left\{\prod_{j=1}^{N}\left(\prod_{i=1}^{n}(x_i^*)^{a_{ij}}\right)^{y_j^*}\right\} =$$

$$= \left\{\prod_{j=1}^{N}\left(\frac{c_j}{y_j^*}\right)^{y_j^*}\right\} \left\{\prod_{i=1}^{n}(x_i^*)^{\sum_{j=1}^{N} a_{ij} y_j^*}\right\} = \prod_{j=1}^{N}\left(\frac{c_j}{y_j^*}\right)^{y_j^*}.$$

В последнем выражении учтено условие $\sum_{j=1}^{N} a_{ij} y_j = 0$. Следовательно, значение z^* можно вычислить, как только будут определены все y_j. При известных y_j^* и z^* можно определить $U_j^* = y_j^* z^*$. Значения x_j^* находятся как решение системы уравнений

$$U_j^* = c_j \prod_{i=1}^{n} (x_i^*)^{a_{ij}}, \; j = 1, 2, \ldots, N.$$

Рассмотренная процедура показывает, что исходная задача минимизации позинома z может быть сведена к решению системы линейных уравнений относительно y_j. При этом значения всех y_j^* определяются из необходимых условий существования минимума. Можно показать, что эти условия являются также и достаточными.

Фактически переменные y_j определяют двойственные переменные, соответствующие прямой задаче минимизации z. Чтобы установить эту зависимость, запишем целевую функцию прямой задачи в виде

$$z = \sum_{j=1}^{N} y_j \left(\frac{U_j}{y_j} \right).$$

Определим теперь функцию

$$w = \prod_{j=1}^{N} \left(\frac{U_j}{y_j} \right)^{y_j} = \prod_{j=1}^{N} \left(\frac{c_j}{y_j} \right)^{y_j}.$$

Поскольку $\sum_{j=1}^{N} y_j = 1$ и $y_j > 0$, в силу неравенства Коши[1] имеем $w \le z$. (Неравенство Коши (неравенство между средним арифметическим и средним геометрическим) устанавливает, что при $z_j > 0$ выполняется $\sum_{j=1}^{N} w_j z_j \ge \prod_{j=1}^{N} (z_j)^{w_j}$, где $w_j > 0$ и $\sum_{j=1}^{N} w_j = 1$.)

Функция w с переменными y_1, y_2, …, y_N определяет задачу, двойственную исходной. Поскольку w является нижней границей значений z и рассматривается задача минимизации z, отсюда следует, что, максимизируя функцию w, мы получим

$$w^* = \max_{y_j} w = \min_{x_i} z = z^*.$$

Это значит, что максимальное значение w (= w^*) на множестве допустимых значений y_j совпадает с минимальным значением z (= z^*) на множестве допустимых значений x_i.

Пример 3.5

В этом примере рассматривается задача, где $N = n + 1$, так что решение, получаемое из условий ортогональности и нормировки, является единственным. Следующий пример иллюстрирует другой случай, когда $N > n + 1$.

Рассмотрим задачу

Минимизировать $z = 7x_1x_2^{-1} + 3x_2x_3^{-2} + 5x_1^{-3}x_2x_3 + x_1x_2x_3$.

Эту функцию можно записать в виде

$$z = 7x_1^{1}x_2^{-1}x_3^{0} + 3x_1^{0}x_2^{1}x_3^{-2} + 5x_1^{-3}x_2^{1}x_3^{1} + x_1^{1}x_2^{1}x_3^{1},$$

так что здесь

$$(c_1, c_2, c_3, c_4) = (7, 3, 5, 1),$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -3 & 1 \\ -1 & 1 & 1 & 1 \\ 0 & -2 & 1 & 1 \end{bmatrix}.$$

Условия ортогональности и нормировки приводят к системе уравнений

$$\begin{bmatrix} 1 & 0 & -3 & 1 \\ -1 & 1 & 1 & 1 \\ 0 & -2 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Эта система имеет единственное решение

$$y_1^* = \frac{12}{24},\; y_2^* = \frac{4}{24},\; y_3^* = \frac{5}{24},\; y_4^* = \frac{3}{24}.$$

Следовательно,

$$z^* = \left(\frac{7}{\left(\frac{12}{24}\right)}\right)^{\frac{12}{24}} \left(\frac{3}{\left(\frac{4}{24}\right)}\right)^{\frac{4}{24}} \left(\frac{5}{\left(\frac{5}{24}\right)}\right)^{\frac{5}{24}} \left(\frac{1}{\left(\frac{3}{24}\right)}\right)^{\frac{3}{24}} \approx 15{,}22.$$

Из уравнения $U_j^* \approx y_j^* z^*$ следует, что

$$7x_1x_2^{-1} = U_1 = \left(\frac{1}{2}\right)(15{,}22) = 7{,}61,$$

$$3x_2x_3^{-2} = U_2 = \left(\frac{1}{6}\right)(15{,}22) = 2{,}54,$$

$$5x_1^{-3}x_2x_3 = U_3 = \left(\frac{5}{24}\right)(15{,}22) = 3{,}17,$$

$$x_1x_2x_3 = U_4 = \left(\frac{1}{8}\right)(15{,}22) = 1{,}90.$$

Эта система имеет решение $x_1^* = 1{,}315$, $x_2^* = 1{,}21$, $x_3^* = 1{,}2$, которое является оптимальным решением прямой задачи.

Пример 3.6

Рассмотрим задачу.

Минимизировать $z = 5x_1x_2^{-1} + 2x_1^{-1}x_2 + 5x_1 + x_2^{-1}$.

Условия ортогональности и нормировки приводят к системе уравнений

$$\begin{bmatrix} 1 & -1 & 1 & 0 \\ -1 & 1 & 0 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Поскольку $N > n + 1$, эти уравнения не позволяют непосредственно определить искомые значения y_j. Выражая y_1, y_2 и y_3 через y_4, имеем

$$\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 \\ y_4 \\ 1-y_4 \end{bmatrix},$$

$$y_1 = \frac{(1-3y_4)}{2},$$

$$y_2 = \frac{(1-y_4)}{2},$$

$$y_3 = y_4.$$

Двойственную задачу можно записать в следующем виде.

Максимизировать $w = \left[\dfrac{5}{0.5(1-3y_4)}\right]^{0.5(1-3y_4)} \left[\dfrac{2}{0.5(1-y_4)}\right]^{0.5(1-y_4)} \left(\dfrac{5}{y_4}\right)^{y_4} \left(\dfrac{1}{y_4}\right)^{y_4}$.

Максимизация w эквивалентна максимизации функции $\ln w$, переход к которой упрощает вычисления. Получим

$$\ln w = 0{,}5(1-3y_4)\{\ln 10 - \ln(1-3y_4)\} + 0{,}5(1-y_4)\{\ln 4 - \ln(1-y_4)\} + y_4\{\ln 5 - \\ - \ln y_4 + \ln 1 - \ln y_4)\}$$

Значение y_4, максимизирующее функцию $\ln w$, должно быть единственным, поскольку прямая задача имеет единственный минимум. Следовательно,

$$\frac{\partial \ln w}{\partial y_4} = \left(\frac{-3}{2}\right)\ln 10 - \left\{\left(\frac{-3}{2}\right) + \left(\frac{-3}{2}\right)\ln(1-3y_4)\right\} + \left(\frac{-1}{2}\right)\ln 4 - \left\{\left(\frac{-1}{2}\right) + \left(\frac{-1}{2}\right)\ln(1-y_4)\right\} + \\ + \ln 5 - \{1 + \ln y_4\} + \ln 1 - \{1 + \ln y_4\} = 0.$$

После упрощений получаем

$$-\ln\left(\frac{2\times 10^{\frac{3}{2}}}{5}\right) + \ln\left[\frac{(1-3y_4)^{\frac{3}{2}}(1-y_4)^{\frac{1}{2}}}{y_4^2}\right] = 0$$

или

$$\frac{\sqrt{(1-3y_4)^3(1-y_4)}}{y_4^2} = 12{,}6,$$

откуда $y_4^* \approx 0{,}16$. Следовательно, $y_3^* = 0{,}16$, $y_2^* = 0{,}42$, $y_1^* = 0{,}26$.

Значение z^* вычисляется по формуле

$$z^* = w^* = \left(\frac{5}{0{,}26}\right)^{0.26} \left(\frac{2}{0{,}42}\right)^{0.42} \left(\frac{5}{0{,}16}\right)^{0.16} \approx 9{,}661.$$

Следовательно,

$$U_3 = 0{,}16 \times 9{,}661 = 1{,}546 = 5x_1,$$

$$U_4 = 0{,}16 \times 9{,}661 = 1{,}546 = x_2^{-1},$$

Отсюда имеем $x_1^* = 0{,}309$ и $x_2^* = 0{,}647$.

3.2.4. Стохастическое программирование

В стохастическом программировании рассматриваются задачи, в которых отдельные или все параметры являются случайными величинами. Такие ситуации типичны в реальных задачах, когда трудно определить точные значения параметров.

Основной подход к решению задач стохастического программирования состоит в преобразовании исходной вероятностной задачи в эквивалентную детерминированную. В данном подразделе рассматривается оптимизационная задача с вероятностными ограничениями, которая имеет следующий вид.

Максимизировать $z = \sum_{j=1}^{n} c_j x_j$

при ограничениях

$$P\left\{\sum_{j=1}^{n} a_{ij} x_j \le b_i\right\} \ge 1 - \alpha_i,\ i = 1, 2, \ldots, m;\ x_j \ge 0 \text{ для всех } j.$$

Название "вероятностные ограничения" обусловлено тем, что каждое ограничение задачи должно выполняться с вероятностью не меньше, чем $1 - \alpha_i$, $0 < \alpha_i < 1$. Предполагается, что все коэффициенты a_{ij} и b_i являются случайными величинами. Далее рассматриваются 3 случая. Первые два соответствуют предположениям о том, что только или a_{ij} и b_i или b_i являются случайными величинами, а третий объединяет эти два случая. Во всех трех ситуациях предполагается, что параметры являются нормально распределенными случайными величинами с известными математическими ожиданиями и дисперсиями.

Случай 1. Предполагается, что все a_{ij} являются нормально распределенными случайными величинами с математическими ожиданиями $M\{a_{ij}\}$ и дисперсиями $D\{a_{ij}\}$. Также известны ковариации $cov\{a_{ij}, a_{i'j'}\}$ случайных величин a_{ij}, $a_{i'j'}$.

Рассмотрим i-е ограничение задачи

$$P\left\{\sum_{j=1}^{n} a_{ij} x_j \le b_i\right\} \ge 1 - \alpha_i$$

и введем обозначение

$$h_i = \sum_{j=1}^{n} a_{ij} x_j\ .$$

Случайная величина h_i имеет нормальное распределение с математическим ожиданием $M\{h_i\} = \sum_{j=1}^{n} M\{a_{ij}\}x_j$ и дисперсией $D\{h_i\} = X^T D_i X$, где $X = (x_1, x_2, \ldots, x_n)^2$,

D_i = i-я матрица ковариаций = $\begin{bmatrix} D\{a_{i1}\} & \cdots & cov\{a_{i1}, a_{in}\} \\ \vdots & & \vdots \\ cov\{a_{in}, a_{i1}\} & \cdots & D\{a_{in}\} \end{bmatrix}$.

Теперь имеем

$$P\{h_i \le b_i\} = P\left\{\frac{h_i - M(h_i)}{\sqrt{D\{h_i\}}} \le \frac{b_i - M(h_i)}{\sqrt{D\{h_i\}}}\right\} \ge 1 - \alpha_i,$$

где $\frac{h_i - M(h_i)}{\sqrt{D\{h_i\}}}$ – нормированная нормально распределенная случайная величина с нулевым математическим ожиданием и единичной дисперсией. Это означает, что

$$P\{h_i \le b_i\} = \Phi\left(\frac{b_i - M(h_i)}{\sqrt{D\{h_i\}}}\right),$$

где через Φ обозначена функция распределения стандартного нормального распределения.

Пусть K_{α_i} – значение стандартной нормально распределенной случайной величины, определяемое из уравнения

$$\Phi(K_{\alpha_i}) = 1 - \alpha_i .$$

В этом случае неравенство $P\{h_i \le b_i\} \ge 1 - \alpha_i$ выполняется тогда и только тогда, когда

$$\frac{b_i - M(h_i)}{\sqrt{D\{h_i\}}} \ge K_{\alpha_i} .$$

Это приводит к детерминированному нелинейному ограничению, которое эквивалентно исходному вероятностному

$$\sum_{j=1}^{n} M\{a_{ij}\}x_j + K_{\alpha_i}\sqrt{X^T D_i X} \le b_i .$$

В частности, если a_{ij} – независимые нормально распределенные случайные величины, тогда $cov\{a_{ij}, a_{i'j'}\} = 0$ и последнее неравенство принимает вид

$$\sum_{j=1}^{n} M\{a_{ij}\}x_j + K_{\alpha_i}\sqrt{\sum_{j=1}^{n} D\{a_{ij}\}x_j^2} \le b_i .$$

Данное ограничение можно записать в виде ограничений задачи сепарабельного программирования (подразд. 3.2.1), для чего используется замена переменных

$$y_i = \sqrt{\sum_{j=1}^{n} D\{a_{ij}\} x_j^2} \text{ для всех } i.$$

Таким образом, исходное ограничение эквивалентно неравенству

$$\sum_{j=1}^{n} M\{a_{ij}\} x_j + K_{\alpha_i} y_i \le b_i,$$

и уравнению

$$\sum_{j=1}^{n} D\{a_{ij}\} x_j^2 - y_i^2 = 0.$$

Случай 2. Здесь предполагается, что только b_i являются нормально распределенными случайными величинами с математическим ожиданием $M\{b_i\}$ и дисперсией $D\{b_i\}$. Анализ этой ситуации проводится аналогично случаю 1. Рассмотрим стохастическое ограничение

$$P\left\{ b_i \ge \sum_{j=1}^{n} a_{ij} x_j \right\} \ge \alpha_i.$$

Как и в первом случае, имеем

$$P\left\{ \frac{b_i - M(b_i)}{\sqrt{D\{b_i\}}} \ge \frac{\sum_{j=1}^{n} a_{ij} x_j - M(b_i)}{\sqrt{D\{b_i\}}} \right\} \ge \alpha_i.$$

Это ограничение выполнимо лишь при выполнении неравенства

$$\frac{\sum_{j=1}^{n} a_{ij} x_j - M(b_i)}{\sqrt{D\{b_i\}}} \le K_{\alpha_i}.$$

Итак, исходное вероятностное ограничение эквивалентно детерминированному линейному

$$\sum_{j=1}^{n} a_{ij} x_j \le M\{b_i\} + K_{\alpha_i} \sqrt{D\{b_i\}}.$$

Случай 3. Теперь предположим, что все параметры a_{ij} и b_i являются нормально распределенными случайными величинами. Перепишем ограничение

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i$$

в виде

$$\sum_{j=1}^{n} a_{ij} x_j - b_i \le 0.$$

Так как все a_{ij} и b_i распределены по нормальному закону, в соответствии с известными результатами математической статистики величина $\sum_{j=1}^{n} a_{ij}x_j - b_i$ также имеет нормальное распределение. Отсюда следует, что данный вариант подобен случаю 1 и может быть рассмотрен аналогичным образом.

Пример 3.7

Рассмотрим задачу с вероятностными ограничениями.

Максимизировать $z = 5x_1 + 6x_2 + 3x_3$

при ограничениях

$$P\{a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \le 8\} \ge 0{,}95,$$

$$P\{5x_1 + x_2 + 6x_3 \le b_2\} \ge 0{,}10,$$

причем все $x_j \ge 0$. Пусть a_{1j} – независимые нормально распределенные случайные величины со следующими значениями математических ожиданий и дисперсий:

$$M\{a_{11}\} = 1,\ M\{a_{12}\} = 3,\ M\{a_{13}\} = 9,$$

$$D\{a_{11}\} = 25,\ D\{a_{12}\} = 16,\ D\{a_{13}\} = 4.$$

Пусть параметр b_2 является нормально распределенной случайной величиной с математическим ожиданием 7 и дисперсией 9.

По таблице функции распределения стандартного нормального закона находим

$$K_{a1} = K_{0,05} \approx 1{,}645,\ K_{a2} = K_{0,10} \approx 1{,}285.$$

Первое ограничение задачи эквивалентно детерминированному неравенству

$$x_1 + 3x_2 + 9x_3 + 1.645\sqrt{25x_1^2 + 16x_2^2 + 4x_3^2} \le 8,$$

а второе ограничение – неравенству

$$5x_1 + x_2 + 6x_3 \le 7 + 1.285 \times 3 = 10{,}855.$$

Если положить

$$y^2 = 25x_1^2 + 16x_2^2 + 4x_3^2,$$

исходная задача принимает следующий вид.

Максимизировать $z = 5x_1 + 6x_2 + 3x_3$

при ограничениях

$$x_1 + 3x_2 + 9x_3 + 1{,}645y \le 8,$$

$$25x_1^2 + 16x_2^2 + 4x_3^2 - y^2 = 0,$$

$$5x_1 + x_2 + 6x_3 \le 10{,}855,$$

$$x_1,\ x_2,\ x_3,\ y \ge 0,$$

и может быть решена методами сепарабельного программирования.

3.2.5. Метод линейных комбинаций

Этот метод ориентирован на решение задач, в которых все ограничения являются линейными:

Максимизировать $z = f(\boldsymbol{X})$

при ограничениях

$$AX \le \mathrm{b},\ X \ge 0.$$

Основой метода линейных комбинаций является градиентный метод наискорейшего подъема (подразд. 3.1.2). Известно, что в этом методе движение по направлению градиента может вывести за пределы области допустимых решений. Кроме того, в точке условного оптимума градиент не обязательно обращается в нуль. Поэтому метод наискорейшего подъема необходимо модифицировать в целях его применения к рассматриваемой задаче с ограничениями.

Пусть X^k – *допустимая* точка, полученная на k-й итерации. Разложим целевую функцию $f(X)$ в окрестности точки X^k в ряд Тейлора. В результате получим

$$f(X) \approx f(X^k) + \nabla f(X^k)(X - X^k) = f(X^k) - \nabla f(X^k)X^k + \nabla f(X^k)X.$$

Нам необходимо определить допустимую точку $X = X^*$**,** в которой достигается максимум функции $f(X)$ при выполнении (линейных) ограничений задачи. Так как $f(X^k) - \nabla f(X^k)X^k$ – постоянное слагаемое, задача определения X^* сводится к решению следующей задачи линейного программирования.

Максимизировать $w_k(X) = \nabla f(X^k)X$

при ограничениях

$$AX \le b,\ X \ge 0.$$

Так как целевая функция w_k определяется градиентом функции $f(X)$ в точке X^k**,** улучшение имеющегося решения может быть обеспечено лишь тогда, когда $w_k(X^*) > w_k(X^k)$. Из разложения Тейлора следует, что выполнение этого условия не гарантирует, что $f(X^*) > f(X^k)$, так как X^* не обязательно находится в малой окрестности точки X^{k+1}. Однако при выполнении условия $w_k(X^*) > w_k(X^k)$ на отрезке (X^k, X^*) должна существовать точка X^{k+1}, такая, что $f(X^{k+1}) > f(X^k)$. Следовательно, задача сводится к нахождению такой точки X^{k+1}. Определим эту точку следующим образом:

$$X^{k+1} = (1 - r)X^k + rX^* = X^k + r(X^* - X^k),\ 0 < r \le 1.$$

Отсюда следует, что точка X^{k+1} является линейной комбинацией точек X^k и X^*. Так как X^k и X^* – точки выпуклой области допустимых решений, поэтому точка X^{k+1} также является допустимой. Если сравнить эту процедуру с методом наискорейшего подъема (подраздел 3.1.2), параметр r можно рассматривать как длину шага.

Точка X^{k+1} определяется из условия максимума функции $f(X)$. Так как X^{k+1} зависит лишь от переменной r, то X^{k+1} определяется путем максимизации функции

$$h(r) = f[X^* + r(X^* - X^k)].$$

Процедура повторяется до тех пор, пока на k-й итерации не будет выполнено условие $w_k(X^*) \le w_k(X^k)$. В этой точке дальнейшее улучшение имеющегося решения невозможно, процесс вычислений завершается, точка X^k считается наилучшим решением.

Задачи линейного программирования, которые решаются на каждой итерации описанной процедуры, отличаются друг от друга лишь коэффициентами целевой функции. Поэтому для повышения эффективности вычислений можно использовать методы анализа чувствительности.

Пример 3.8

Рассмотрим задачу квадратичного программирования из примера 3.4.

$$\text{Максимизировать } z = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

при ограничениях

$$x_1 + 2x_2 \le 2,$$

$$x_1, x_2 \ge 0.$$

Пусть $X^0 = (1/2, 1/2)$ – начальная точка, которая является допустимой для рассматриваемой задачи. Имеем

$$\nabla f(X) = (4 - 4x_1 - 2x_2, 6 - 2x_1 - 4x_2).$$

Первая итерация

$$\nabla f(X^0) = (1, 3).$$

Соответствующая задача линейного программирования сводится к максимизации функции $w_1 = x_1 + 3x_2$ с учетом ограничений исходной задачи. Ее оптимальным решением есть точка $X^* = (0, 1)$. Значения функции w_1 в точках X^0 и X^* равны 2 и 3 соответственно. Следовательно, новая точка ищется в виде

$$X^1 = \left(\frac{1}{2}, \frac{1}{2}\right) + r\left[(0,1) - \left(\frac{1}{2}, \frac{1}{2}\right)\right] = \left(\frac{1-r}{2}, \frac{1+r}{2}\right).$$

Максимизация функции

$$h(r) = f\left(\frac{1-r}{2}, \frac{1+r}{2}\right)$$

дает $r = 1$. Таким образом, $X^1 = (0, 1)$ и $f(X^1) = 4$.

Вторая итерация

$$\nabla f(X^1) = (2, 2).$$

Целевой функцией новой задачи линейного программирования является $w_2 = 2x_1 + 2x_2$. Оптимальное решение этой задачи – $X^* = (2, 0)$. Поскольку значения w_2 в точках X^1 и X^* равны 2 и 4, следует найти новую точку. Рассматриваем точку

$$X^2 = (0, 1) + r[(2, 0) – (0, 1)] = (2r, 1 – r).$$

Максимизация функции

$$h(r) = f(2r, 1 – r)$$

дает $r = 1/6$. Таким образом, $X^2 = (1/3, 5/6)$, при этом $f(X^2) \approx 4{,}16$.

Третья итерация

$$\nabla f(X^2) = (1, 2).$$

Здесь целевая функция имеет вид $w_3 = x_1 + 2x_2$. Оптимальным решением задачи линейного программирования будут альтернативные точки $X^* = (0, 1)$ и $X^* = (2, 0)$. Значение w_3 в обоих случаях совпадает со значением w_3 в точке X^2. Следовательно, улучшить имеющееся решение невозможно. Получено приближенное оптимальное решение задачи: $X^2 = (1/3, 5/6)$ с $f(X^2) \approx 4{,}16$. В данном случае оно совпадает с точным оптимумом.

3.2.6. Алгоритм последовательной безусловной максимизации

В этом подразделе приводится описание градиентного метода более общего вида. Предполагается, что целевая функция $f(X)$ является вогнутой, а каждая функция ограничений $g_i(X)$ – выпуклой. Кроме того, допустимая область задачи должна содержать внутренние точки. Это исключает как явное, так и неявное использование ограничений в виде равенств.

Алгоритм последовательной безусловной максимизации основан на преобразовании исходной задачи с ограничениями в эквивалентную задачу *без* ограничений. Процедура в некоторой степени подобна применению метода множителей Лагранжа. Преобразованная задача затем может быть решена методом наискорейшего подъема (подразд. 3.1.2).

Введем новую целевую функцию

$$p(X,t) = f(X) + t\left(\sum_{i=1}^{m}\frac{1}{g_i(X)} - \sum_{j=1}^{n}\frac{1}{x_j}\right),$$

где t – неотрицательный параметр. Наличие второй суммы обусловлено учетом требований неотрицательности переменных исходной задачи, которые следует записать в виде $-x_j \le 0$ для их согласования с остальными ограничениями вида $g_i(X) \le 0$. Поскольку функция $g_i(X)$ является выпуклой, $1 / g_i(X)$ будет вогнутой. Отсюда следует, что $p(X, t)$ является вогнутой функцией от переменных X. Следовательно, функция $p(X, t)$ имеет единственный максимум. Поэтому решение исходной оптимизационной задачи с ограничениями эквивалентно максимизации функции $p(X, t)$.

Выполнение алгоритма начинается с произвольного выбора начального *неотрицательного* значения t. Начальная точка X^0 выбирается в качестве первого пробного решения. Эта точка должна быть внутренней точкой области допустимых решений, т.е. не должна находиться на ее границе. При заданном значении t оптимальное решение (точка максимума) функции $p(X, t)$ находится методом наискорейшего подъема.

Точка, представляющая новое решение, всегда будет внутренней, ибо если решение находится близко к границе области, то по крайней мере одна из функций $1 / g_i(X)$ или $-1 / x_j$ примет очень большое отрицательное значение. Поскольку рассматривается задача максимизации $p(X, t)$, то такие решения автоматически исключаются. Основной результат состоит в том, что последовательно получаемые решения всегда будут внутренними точками допустимой области. Следовательно, исходная задача может всегда рассматриваться как задача без ограничений.

Когда получено оптимальное решение, соответствующее заданному значению t, выбирается новое значение t и процесс вычислений (методом наискорейшего подъема) повторяется. Если t' – текущее значение t, то следующее значение t'' необходимо выбирать так, чтобы выполнялись неравенства $0 < t'' < t'$.

Решение задачи методом последовательной безусловной максимизации завершается, когда для двух последовательных значений t соответствующие *оптимальные* решения X, полученные в результате максимизации функции $p(X, t)$, примерно совпадают. В этом случае дальнейшие вычисления лишь незначительно улучшат полученное решение.

Практическая реализация алгоритма последовательной безусловной максимизации имеет ряд нюансов, которые здесь не рассматриваются. В частности, существенным фактором является выбор начального значения параметра t, который может повлиять на скорость сходимости алгоритма. Далее для определения начальной внутренней точки области допустимых решений могут потребоваться специальные алгоритмы.

3.3. Заключение

Методы решения задач нелинейного программирования можно классифицировать как *прямые* и *непрямые*. Примерами прямых методов являются градиентные алгоритмы, в которых поиск максимума (минимума) в оптимизационной задаче ведется по направлению наискорейшего увеличения (уменьшения) значения целевой функции. В непрямых методах исходная задача оптимизации заменяется вспомогательной, оптимальное решение которой принимается за решение исходной. Частными случаями таких методов являются квадратичное, сепарабельное и стохастическое программирование.

4. МЕТОДЫ РЕШЕНИЯ КОМБИНАТОРНО-ОПТИМИЗАЦИОННЫХ ЗАДАЧ

4.1. Стратегии декомпозиции множества решений и дерево поиска

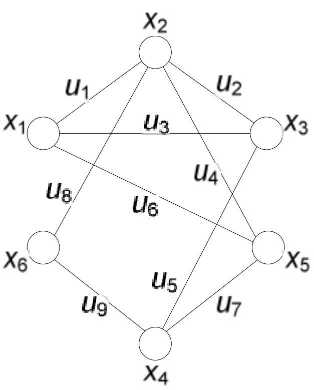
Широкий круг методов поиска решения комбинаторно-оптимизационных задач основан на двух идеях: декомпозиция пространства решений и выбор перспективного подмножества вариантов на основе некоторой оценки.

Рассмотрим процесс декомпозиции. Множество возможных вариантов решения M разбивается в соответствии с каким-либо принципом, как правило, на непересекающиеся подмножества M такие, что $\cup M_i = M$. Далее, используя тот же принцип, полученные подмножества вновь разбиваются на подмножества, включающие меньшее количество вариантов. После некоторого шага разбиения каждое множество содержит по одному варианту решения. Сопоставим каждому подмножеству вершину графа. Соединив ребром вершины, соответствующие подмножествам M_i и M_j (если подмножество M_j получено непосредственным разбиением подмножества M_i), придем к так называемому *дереву решений*.

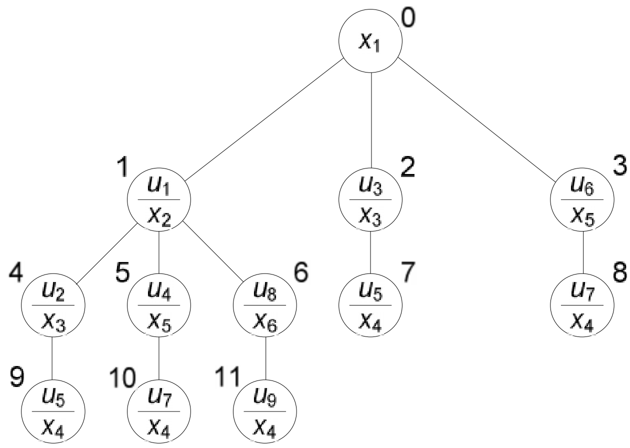
Принцип разбиения пространства решений связан с видом преобразований, которые необходимо выполнить над графом для получения графа результата.

Проиллюстрируем сказанное примером поиска простых цепей (маршрутов) из вершины x_1 в вершину x_4 в графе $G(X, U)$, показанном на рис. 4.1, *а*. Начиная с вершины x_1, будем включать в маршруты ребра *в порядке возрастания их номеров.* Вершина 0 дерева решений (рис. 4.1, б) соответствует множеству всех возможных вариантов маршрутов из x_1 в x_4. Принцип разбиения этого множества в данном случае естественно вытекает из процедуры последовательно-параллельного прослеживания вариантов маршрутов по ребрам графа G, т.е. последовательного включения ребер в возможные маршруты.

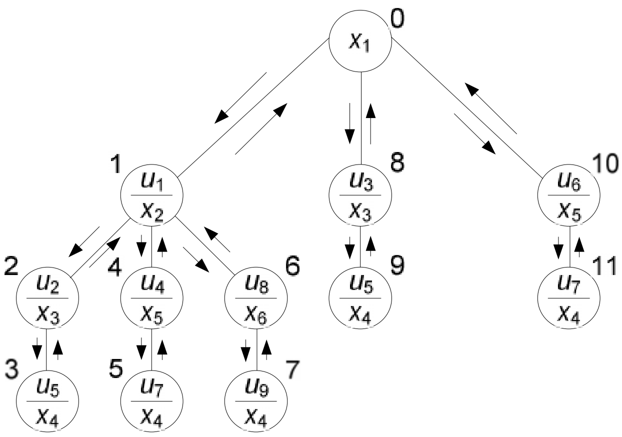
Вершина дерева решений с номером 1 соответствует включению в маршрут ребра u_1 вершины с номерами 2 и 3 – ребер u_3 и u_6. Таким образом, на первом шаге (на 2-м уровне дерева решений) множество M разбивается на три непересекающихся подмножества вариантов, одно из которых M_1, соответствующее вершине 1 дерева решений, порождает следующие три варианта простых цепей графа G: $\{u_1, u_2, u_5\}$, $\{u_1, u_4, u_7\}$ и $\{u_1, u_8, u_9\}$. Далее процесс очевиден. Обратим внимание на то, что в данном случае на следующий уровень дерева решений переходим только после того, как были получены все подмножества текущего уровня.



а



б



в

Рис. 4.1

Описанный процесс реализует получение всех вариантов решения по *методу поиска в ширину* с последовательным формированием состава вариантов и соответствует полному перебору. Очевидно, что в рассмотренном виде этот метод можно использовать, если необходимо найти все варианты решения.

Другая стратегия разбиения множества всех возможных вариантов на подмножества приводит к иному методу, который назовем *поиск в глубину с возвращением.* Рассмотрим его на том же примере. Принцип разбиения пространства решений – тот же, что и выше. Выделим на 1-м шаге из множества M подмножество M_1, включив в маршрут ребро u_1 из возможных $\{u_1, u_3, u_6\}$ в соответствии с минимальным индексом.

Таким образом, разобьем множество M на подмножества M_1 и $M \setminus M_1$. Тогда на 2-м уровне дерева решений получим вершину 1 (см. рис. 4.1, в). Множество M_1 разобьем на подмножества M_2 и $M_1 \setminus M_2$, добавив в маршрут ребро u_2. На третьем уровне дерева решений получим вершину 2. Продвигаясь по этой ветви в глубь дерева решений, получим один из возможных вариантов маршрута, соединяющего вершины x_1 и x_4 – x_1, x_2, x_3, x_4 или в виде последовательности ребер – u_1, u_2, u_5, что соответствует вершине 3 дерева решений (сравните номер такой же вершины на рис. 4.1,б). Теперь возвращаемся по ветви вверх, пока не придем в вершину, такую, что из множества вариантов ей сопоставленных можно выделить некоторое подмножество в соответствии с принятым принципом. В нашем примере такой вершиной будет вершина с номером 1.

Процесс повторяется до тех пор, пока не будут получены все варианты маршрута. Построено дерево решений по методу поиска в глубину с возвращением. Как видно из рис. 4.1, *б* и *в*, реализация методов поиска в ширину и в глубину с возвращением приводит к разному порядку получения решений. В данном случае, когда ищем все решения, это не играет роли, однако, что и увидим в дальнейшем, может быть весьма существенным при решении других задач.

4.2. Методы поиска решения, использующие идею отсечения

В практике проектирования задачи нахождения всех возможных решений довольно редки. Обычно требуется найти одно решение, наилучшее в смысле некоторого функционала – целевой функции. Поиск такого решения декомпозицией множества вариантов по методу в ширину или в глубину с возвращением подразумевает вычисление значений целевой функции для каждого варианта, сравнение их и выбор оптимального, т.е. приводит к полному перебору. Полный перебор требует слишком много времени либо просто нереализуем по этой причине при достаточно большой размерности задачи.

Существование некоторой оценки, которая с той или иной степенью достоверности дает возможность судить о том, содержит ли подмножество вариантов оптимальное решение, позволило бы избежать полного перебора. В общем случае оценка - это значение функции $F(M_i)$ на вершинах дерева решений, равная в его конечных вершинах целевой функции для соответствующего варианта решения, а в остальных – нижней или верхней границе функции $\boldsymbol{F}$ для вариантов, входящих в это множество.

Нижняя и верхняя границы означают, что значение целевой функции для всех вариантов, порождаемых множеством M_i, не меньше и не больше числа $F(M_i)$ соответственно. Поэтому оценочная функция, определяющая нижнюю границу, в процессе разбиения множества на подмножества не убывает, а определяющая верхнюю – не возрастает (напомним, что нижняя граница вычисляется в задачах на минимум, а верхняя – на максимум). Таким образом, если по мере разбиения множества M значение нижней границы на каком-то шаге уменьшилось (для верхней – увеличилось), это означает, что оценочная функция не отвечает предъявляемым к ней требованиям, т.е. выбрана неправильно.

Оценочные функции можно определить на любых подмножествах вариантов либо лишь на некоторых. В первом случае выбор принципа разбиения множества М на подмножества не зависит от оценочной функции. Во втором случае используемый принцип разбиения должен обеспечивать получение только таких подмножеств, на которых определена оценочная функция. Отсюда следует, что вид дерева решений зависит от оценочной функции.

Вид оценочной функции и та степень достоверности, с которой можно судить по ней о наличии в подмножестве оптимального варианта, порождает различные методы поиска по дереву решений.

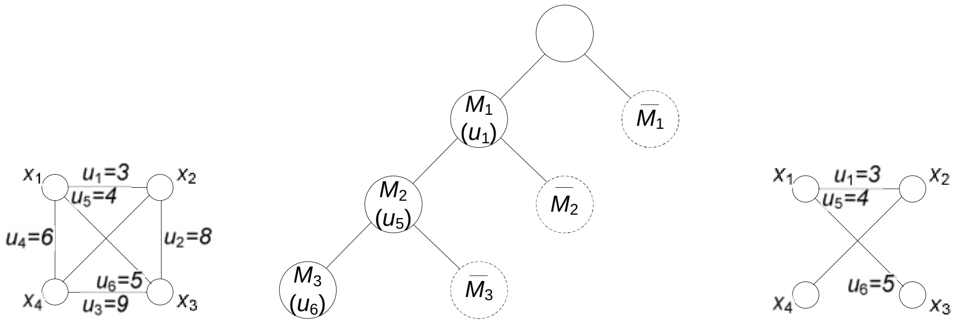
При полной достоверности оценки того, что подмножество не содержит оптимального решения, его можно исключить из процесса разбиения – говорят, что в дереве решений отсекаются ветви и вершины, следующие за вершиной, сопоставленной этому подмножеству. В противном случае такая оценка может служить для обоснования очередности разбиения подмножеств, т.е. выбора на каждом шаге построения дерева решений наиболее «перспективной» вершины, которая с большей вероятностью, чем другие, может содержать оптимальное решение. В дальнейшем будем на этом основании различать *отсекающую оценку* и *оценку перспективности.*

Выбор оценочной функции – один из самых сложных и ответственных этапов разработки алгоритмов, использующих методы поиска с отсечением. От него зависит как эффективность таких алгоритмов, так и их точность. В свою очередь, возможности оценочной функции в значительной степени определяются спецификой задачи и математическими свойствами графов – объекта и результата проектирования.

Сделаем несколько общих замечаний. Оценочная функция в виде верхней или нижней границы целевой функции (возможно ее текущего значения, например сумма длин ребер построенного участка маршрута) обычно является основанием для выбора более перспективного подмножества, но может служить и отсекающей оценкой. В качестве отсекающей оценки для большинства комбинаторно-оптимизационных задач может выступать опорное решение, т.е. значение целевой функции какого-либо варианта решения. Несомненно, что если для задачи на минимум для какого-то подмножества нижняя граница или текущее значение целевой функции больше или равно значению целевой функции некоторого решения, это подмножество не содержит оптимального варианта. Однако трудно ожидать выполнения этого условия на ранних этапах разбиения пространства решений.

Метод поиска в глубину. Этот метод является вырожденным случаем разбиения пространства решений. Он заключается в том, что на каждом уровне дерева решений в соответствии с некоторой оценкой, определяемой спецификой задачи, выбирают одно подмножество, не формируя (отсекая) остальные, до тех пор, пока не получим решение. Таким образом, здесь формируется только одна ветвь дерева решений. Очевидно, что алгоритмы, основанные на этом методе, должны быть весьма эффективными в смысле затрат времени и памяти ЭВМ. Если оценка выбора является отсекающей, т.е. с вероятностью, равной единице, позволяет судить о том, что оптимальный вариант находится в данном подмножестве, получаем точное решение. В противном случае гарантируется лишь приближенное. Это положение имеет очень большое значение, так как в большинстве источников утверждается, что последовательные алгоритмы, реализующие метод поиска в глубину, гарантируют лишь приближенное решение.

Проиллюстрируем сказанное примером идеи алгоритма Прима, предназначенного для построения остовного дерева минимального веса графа ***G***. В ходе доказательства правильности этого алгоритма будет показано, что выбор на каждом шаге ребра минимальной длины (не образующего цикл) обеспечивает получение точного решения. Таким образом, минимальная длина очередного ребра является достоверной оценкой, а включение в строящееся дерево такого ребра отсекает все вершины дерева решений, которые соответствуют вариантам остовного дерева графа ***G***, не содержащего этого ребра. На рис. 4.3 показаны полный граф G на четыре вершины, дерево решений и полученное остовное дерево минимальной длины; здесь M_1 и $\overline{M_1}$ – все варианты деревьев графа $G(X, U)$, содержащих и не содержащих ребро u_1; $M_2 \subset M_1$ – все варианты деревьев, включающих ребра u_1 и u_5; и $\overline{M_2} \subset M_1$ – все варианты деревьев, содержащих ребро u_1 и не содержащих ребро u_5; M_3 – остовное дерево минимальной длины графа G: $\{u_1, u_5, u_6\}$, вес которого равен 12.



а *б* *в*

Рис. 4.3

Метод параллельного поиска. Метод параллельного, или одновременного, поиска относится к классу эвристических и представляет упрощенный вариант метода поиска в ширину. Он заключается в следующем: множество решений разбивается на подмножества 1-го уровня, каждая вершина которого становится началом ветви, формируемой по методу поиска в глубину. Переход к формированию подмножеств следующего уровня дерева решений выполняется после получения подмножеств всех ветвей текущего уровня. Дерево решений, иллюстрирующее идею метода, показано на рис. 4.4; здесь n – количество искомых решений, следовательно, на j-м уровне дерева n подмножеств $M_i^{\,j}$, $i = \overline{1, n}$ – множество индексов этих подмножеств (верхний индекс соответствует номеру уровня дерева решений).

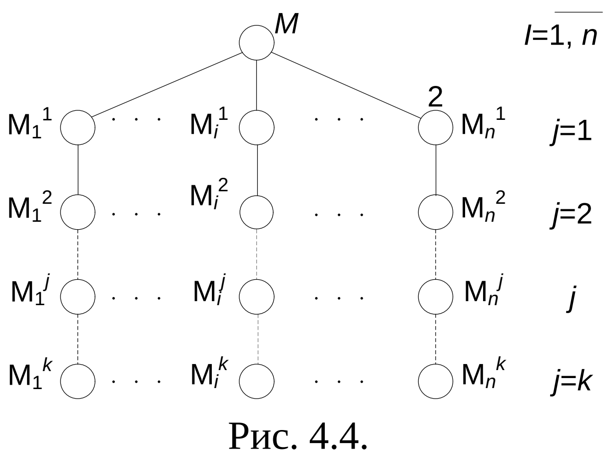


Рис. 4.4.

В зависимости от специфики задачи подмножества разных ветвей могут быть как пересекающимися, так и непересекающимися. Рассмотрим задачу компоновки схемы в $\boldsymbol{n}$ конструктивных модулей. Множество элементов схемы $\boldsymbol{Э}$ поставлено во взаимно-однозначное соответствие множеству вершин $\boldsymbol{X}$ гиперграфа $\boldsymbol{Э} \leftrightarrow \boldsymbol{X}$. Указанная задача может быть решена алгоритмом, реализующим метод параллельного поиска (алгоритм параллельного разрезания гиперграфа). Результатом его работы будет разбиение множества вершин гиперграфа на n подмножеств X, $i = \overline{\mathbf{1},\boldsymbol{n}}$. Так как один и тот же элемент схемы не может входить в разные конструктивные модули и состав X_i определяется по методу поиска в глубину последовательным включением вершин $x \in X$ в X^1_i (X^1_i соответствует M^j_i), то $X^j_i \cap X^j_p = \varnothing$ для всех $i, p \in I = \overline{\mathbf{1},\boldsymbol{n}}$. Таким образом, в данной задаче подмножества, принадлежащие разным ветвям дерева решений, должны быть непересекающимися.

Если подмножества вариантов 1-го уровня содержат все варианты решения, т.е. удовлетворяют условию $\cup M^1_i$ = М и оценка выбора подмножества в каждой ветви является отсекающей, то метод обеспечивает точное решение.

Вырожденным случаем оценочной функции является невычисляемая отсекающая оценка, например, некоторое заранее сформулированное условие. Примером может служить задача о кодовом замке. Пусть кодовый замок имеет четыре разряда, и каждый разряд может принимать значения «0» и «1». Мы забыли код, но помним, что в нем не может быть подряд трех нулей или единиц – условие отсечения. Мы готовы перепробовать все коды, но хотели бы избежать этого. Нам нужен алгоритм для систематического генерирования четырехразрядных комбинаций из «0» и «1». Учитывая вероятность того, что мы откроем замок раньше, чем будут испытаны все комбинации, целесообразно использовать метод поиска в глубину с возвращением. Примем направление обхода дерева решений слева направо. Принцип разбиения множества решений – присваивание соответствующему разряду значения «0» или «1» (количество сгенерированных разрядов равно номеру уровня дерева решений). Очевидно, что дерево будет бинарным. Тогда алгоритм можно сформулировать так: движемся вниз по дереву, придерживаясь левой ветви, пока не получим полной комбинации либо не выясним нецелесообразность этого (рис. 4.5 – отсечение левой ветви после вершины с кодом 11). Если комбинация не подходит либо ее или некоторое множество комбинаций нет смысла набирать (отсечение по условию), то возвращаемся в ближайшую вершину и пробуем спускаться по другой ветви (левая ветвь – разряд ставим в положение «1», правая – в «0»). В некоторых источниках этот метод называют «программирование с отходом назад».

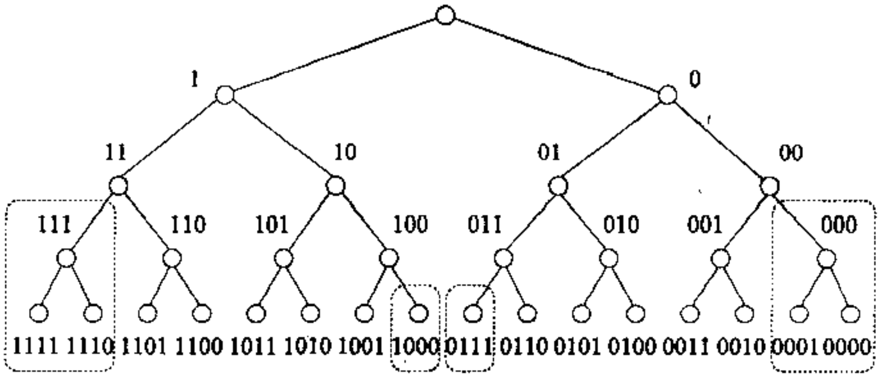


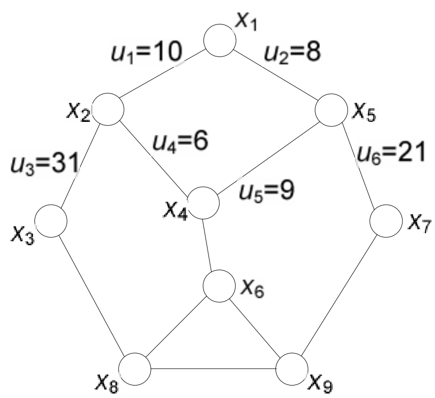
Рис. 4.5

Рассмотрим пример использования текущего значения целевой функции в качестве оценки перспективности во всех вершинах дерева решений и этой же величины в качестве отсекающей – в «особых» вершинах дерева решений. Ранее была коротко рассмотрена задача нахождения всех простых цепей из исходной вершины графа в некоторую конечную. Модифицируем ее: необходимо найти простую цепь (маршрут) из некоторой вершины x_i в вершину x_j графа $G(X, U)$, такой, чтобы суммарная длина ребер, его составляющих, была минимальна.

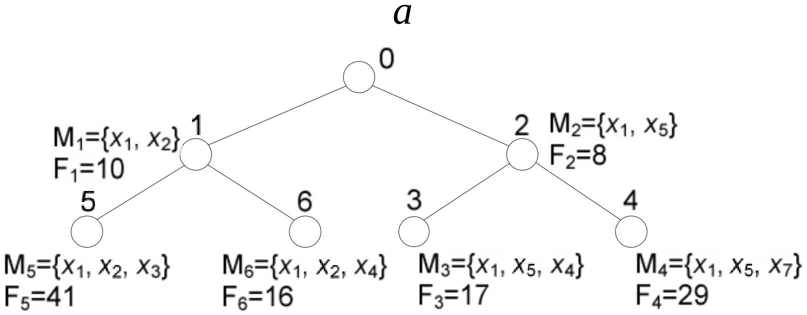
Примем стратегию формирования дерева решения по методу поиска в ширину. Принцип разбиения множества на подмножества тот же, что и выше, т.е. включение в фрагмент маршрута некоторого ребра. В качестве оценочной функции примем суммарную длину ребер, составляющих фрагмент маршрута. Нетрудно убедиться, что эта величина является нижней границей длины маршрута. Действительно, это функция, определенная на вершинах дерева решений, в каждой конечной вершине она принимает значение целевой функции для соответствующего варианта маршрута и, наконец, функция возрастающая, так как длины ребер – величины положительные. Так как доказательства того, что эта оценка является отсекающей, нет, будем использовать ее в качестве оценки перспективности ветви.

Разные простые цепи, соединяющие вершину x_i с вершиной x_j, могут проходить через одну и ту же вершину x_k. Это свойство позволяет в особых точках дерева решений использовать суммарную длину ребер как отсекающую оценку. Отсюда, если суммарная длина ребер от вершины x_i до вершины x_k фрагмента маршрута M_r больше, чем фрагмента маршрута M_t то все варианты маршрутов, порождаемые фрагментом M_r следует отбросить.

На рис. 4.6, *а* показан граф *G(X, U).* Пусть исходной является вершина x_1, а конечной – x_8. На 1 - м уровне дерева решений (рис. 4.6, б) множество возможных вариантов маршрута из вершины x_1 разбивается на два подмножества: маршруты из x_1 через вершину x_2 (вершина дерева решений 1) и через вершину x_5 (вершина дерева решений 2) – длины фрагментов $F_1 = 10$ и $F_2 = 8$ соответственно. На основании оценки перспективности выбираем вторую вершину дерева решений и выполняем ветвление в ней. Из вершины x_5 графа G можно продолжать маршруты в вершины x_4 и x_7. Получаем вершины дерева решений 3 и 4 (фрагмент $M_3 = \{x_1\ x_5, x_4\}$ с $F_3 = 17$ и фрагмент $M_4 = \{x_1, x_5, x_7\}$ с $F_4 = 29$). Выбирая минимальную оценку $F_1 = 10$ (из $F_1 = 10$, $F_3 = 17$ и $F_4 = 29$), выполняем ветвление в вершине 1 дерева решений. Из вершины x_2 графа G можно продолжить маршруты в вершины x_3 и x_4. Соответствующие вершины 5 и 6 дерева решений имеют оценку $F_5 = 41$ и $F_6 = 16$. Вершины 3 и 6 дерева решений имеют оценки $F_3 = 17$ и $F_6 = 16$ и оба фрагмента $\{x_1, x_5, x_4\}$ и $\{x_1, x_2, x_4\}$ порождают все маршруты от вершины x_1 графа G до вершины x_8 через вершину x_4.



а



б

Рис. 4.6

На основании рассмотренного выше свойства простых цепей отсекаем ребра и вершины дерева решений, следующие за вершиной 3. Этот прием используется в алгоритме Дейкстры.

Изложенный в данном подразделе материал позволил подойти к наиболее общему методу решения комбинаторно-оптимизационных задач - методу ветвей и границ. При рассмотрении этого метода коснемся вопроса конструирования оценочной функции для вычисления нижней и верхней границ.

4.3. Метод ветвей и границ

Метод ветвей и границ – универсальный, хотя и достаточно сложный метод точного решения широкого круга комбинаторно-оптимизационных задач направленным перебором вариантов при условии, что их число конечно. В общем случае выполняется частичный перебор. Сокращение числа просмотренных вариантов достигается как за счет организации ветвления, так и за счет отсечения подмножеств вариантов, не содержащих оптимального. Ветвление, т.е. разбиение множества вариантов на подмножества, выполняется по методу в ширину или в глубину с возвращением с заданным порядком построения ветвей. Реализация метода, прежде всего принцип разбиения множества вариантов на подмножества и вычисление условий отсечения, существенно зависит от вида задачи, а степень сокращения перебора – от ее конкретных данных. В худшем случае метод может превратиться в полный перебор.

Организация ветвления подразумевает выбор в процессе разбиения наиболее перспективной вершины для ветвления, т.е. в разбиении того подмножества вариантов, к которому вероятнее всего принадлежит оптимальный. Таким образом, основные операции метода ветвей и границ – это ветвление и отсечение. Как было показано выше, в качестве оценки перспективности ветви используется верхняя или нижняя границы целевой функции, вычисляемые для каждого подмножества вариантов. В ряде случаев они могут использоваться и для отсечения ветвей.

Можно выделить три основных способа отсечения ветвей.

1. Сравнение оценки (нижней – $F_H(M_i)$ или верхней – $F_B(M_i)$ границы) со значением целевой функции для уже найденного (опорного) решения $F_{ОП}$. Действительно, если при решении задачи на минимум целевой функции $F_H(M_i) > F_{ОП}$, то подмножество M_i не содержит оптимального варианта, и соответствующая вершина дерева решений отсекается. Опорное решение можно получить приближенным алгоритмом заранее либо в ходе решения

задачи методом ветвей и границ. При разбиении пространства решений по методу поиска в глубину с возвращением опорное решение получается на более ранних этапах построения дерева решений, чем при методе поиска в ширину.

2. Сравнение двух оценок. Такое отсечение выполняется, если, например, в задаче на минимум целевой функции, для подмножества вариантов соответствующей вершины можно найти оценку снизу $F_H(M_i)$ и сверху $F_B(M_i)$. Тогда, если для некоторого подмножества M_i окажется, что $F_H(M_i) > F_B(M_i)$, то ветвление в вершине, соответствующей M_i, прекращается.

3. Прекращение ветвления, если соответствующее подмножество не содержит оптимального решения, можно установить, во-первых, в «особых точках» по значениям оценочной функции, например нижней границы, и, во-вторых, если известен оптимальный среди вариантов этого подмножества. Чем точнее будет получена оценка, т.е. чем ближе она будет к значению целевой функции для оптимального варианта подмножества, тем меньшее количество вершин дерева решений будет построено и исследовано.

Точность оценки зависит от принципа разбиения множества вариантов на подмножества и вида оценочной функции или способа ее вычисления. Следовательно, если возможно использование нескольких принципов разбиения, необходимо выбирать тот, при котором оценки более точны. Как правило, число отсечений тем больше, чем сильнее отличаются оценки подмножеств.

Таким образом, лучшими являются тот принцип разбиения и такая оценочная функция, при которых разность между оценками подмножеств наибольшая, а сами оценки вычисляются с наибольшей точностью в смысле их близости к значению целевой функции для оптимального варианта подмножества. Число отсечений зависит и от способа ветвления, хотя никаких точных оценок и рекомендаций указать нельзя.

Рассмотрим основные способы ветвления.

1. Разбиение множества вариантов на подмножества по методу поиска в ширину и выбор вершины ветвления по минимуму (максимуму) оценочной функции. Сначала выполняется разбиение всего множества вариантов, т.е. строятся на следующем от корня уровне все или часть вершин его потомков. Затем на каждом шаге выбирается вершина ветвления

по минимуму нижней границы (максимуму верхней для задачи на максимум целевой функции) и разбивается соответствующее ей подмножество вариантов. В ходе ветвления используется, если это возможно, тот или иной способ отсечения ветвей и вершин. Процесс выбора вершин и ветвления повторяется для всех висячих вершин, т.е. еще не отсеченных и не ставших концевыми в смысле их соответствия одному из вариантов решения задачи.

2. Разбиение множества вариантов по методу поиска в глубину с возвращением, т.е. последовательное построение ветвей. Строят полностью одну ветвь дерева решений, т.е. находят один вариант решения задачи. Полученное для этого варианта значение целевой функции может использоваться как отсекающая оценка. Далее ветвление выполняется последовательно от построенной ветви, начиная с вершины M_i, предшествовавшей конечной. При этом, если не происходит отсечение, новая ветвь достраивается до конца. Процедура повторяется по отношению к вершинам новой ветви. После построения всех возможных ветвей, проходящих через вершину, сопоставленную множеству M_i отыскивается следующая вершина и процесс повторяется. При последовательном способе ветвления построение дерева решений может выполняться, начиная с левой ветви – слева направо, или с правой – справа налево. Количество построенных вершин дерева решений зависит от очередности развития ветвей.

3. Комбинация двух рассмотренных способов. Простейшая комбинация заключается в следующем. Строится одна ветвь. Далее развитие дерева решений происходит по методу поиска в ширину, полученное опорное решение используется для отсечения, выбор вершины ветвления выполняется по минимуму нижней границы в задачах на минимум целевой функции (по максимуму верхней).

В ходе решения, если возможно, уточняется значение отсекающей оценки. Оптимальное решение будет найдено, когда в задаче на минимум целевой функции значение для некоторой конечной вершины $F(M_i^k)$ меньше нижней границы $F_H(M_j)$ для всех висячих вершин и меньше значения целевой функции $F(\{M_i^k\})$ для всех остальных конечных вершин (в задаче на поиск максимума целевой функции соответственно больше).

Вычисление верхних и нижних границ и различные способы ветвления рассмотрим на примере нахождения простой цепи (маршрута) максимальной и минимальной длины, соединяющей некоторую исходную и конечную вершины в ориентированном графе G, показанном на рис. 4.7, а. Принцип разбиения множества вершин на подмножества тот же, что и выше, т.е. множество маршрутов, порождаемых множеством M_i, разбиваем на подмножества маршрутов $\{M_j, ..., M_k\}$, содержащих ребра графа G $\{(x_i, x_j), ..., (x_i, x_k)\}$.

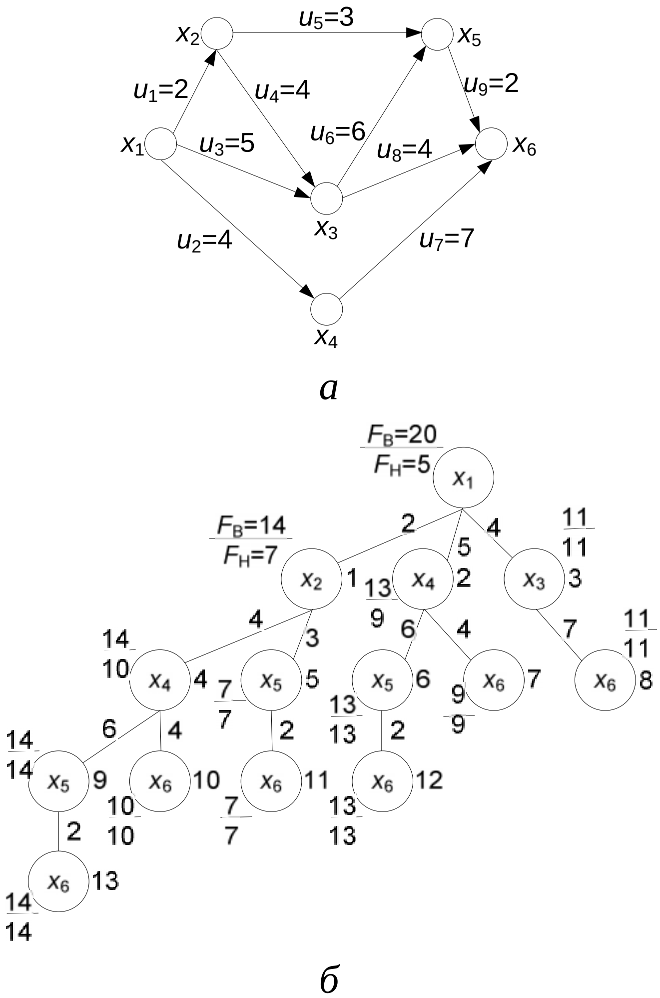


Рис. 4.7

На последующих рисунках (рис. 4.8 – 4.12) в центре кружка, изображающего вершину дерева решений, будем указывать номер вершины графа G. Таким образом, ребро, соединяющее две вершины дерева решений, будет соответствовать ребру графа G, инцидентному таким же вершинам, а само дерево решений – это полное дерево простых цепей, соединяющих некоторые исходную и конечную вершины графа G. Около ребер проставим их длины (веса), сверху вершин дерева решений (слева или справа) – верхнюю и/или нижнюю границы, справа от вершины – номер вершины в порядке появления при построении дерева решений.

Пусть исходной будет вершина x_1, а конечной – x_6. В данном графе простых цепей шесть:

$$c_1 = \{x_1, x_2, x_5, x_6\};$$
$$c_2 = \{x_1, x_2, x_4, x_5, x_6\};$$
$$c_3 = \{x_1, x_2, x_4, x_6\};$$
$$c_4 = \{x_1, x_4, x_5, x_6\};$$
$$c_5 = \{x_1, x_4, x_6\};$$
$$c_5 = \{x_1, x_3, x_6\};$$

Дерево простых цепей, соединяющих в графе G вершины x_1 и x_6 показано на рис. 4.7, б.

Рассмотрим примеры оценочных функций и результаты решения задачи при различных способах ветвления. Выше было показано, что в задачах на поиск маршрута минимальной длины в качестве нижней границы можно использовать сумму длин ребер построенного фрагмента. Решение задачи методом ветвей и границ для графа G с $x_н = x_1$ и $x_к = x_6$ при ветвлении по методу поиска в ширину показано в виде дерева решений на рис. 4.8.

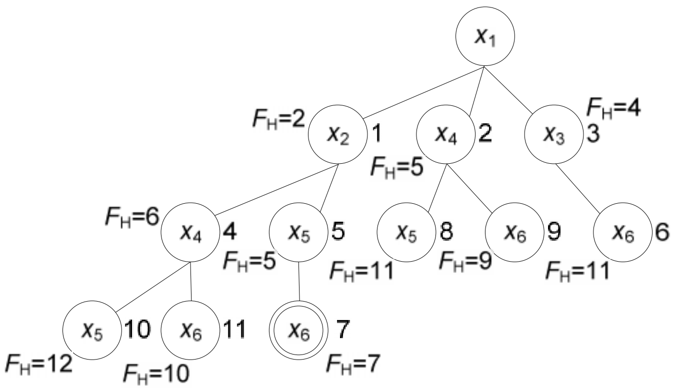


Рис. 4.8

Рассмотрим по шагам этот процесс.

Начинаем построение возможных маршрутов с включения в них ребер $u(x_1, x_2)$, $u(x_1, x_4)$, $u(x_1, x_3)$ графа G, т.е. получаем три вершины дерева решений {1, 2, 3} с нижними границами 2, 5 и 4 соответственно. Выбираем для ветвления вершину 1. После ее ветвления множество вершин дерева решений – {4, 5, 2, 3} с нижними границами 6, 5, 5, 4. Затем ветвим вершину 3. Теперь множество вершин – {4, 5, 2, 6}, их нижние границы – 6, 5, 5,11. Вершина 6 является конечной, нижняя граница для нее равна значению целевой функции для маршрута x_1, x_3, x_6 и на последующих шагах будет использоваться как отсекающая оценка, равная 11.

Множество вершин – кандидатов на ветвление – теперь будет {4, 5, 2}. На 4-м шаге ветвим вершину 5. Теперь множество вершин дерева решений с нижними границами 6, 7 и 5 будет {4, 7, 2}. Вершина 7 является конечной, исключаем ее из списка ветвления и меняем значение отсекающей оценки (теперь она равна 7). На 5-м шаге ветвим вершину 2; вершину 8 с нижней границей, равной 11, и вершину 9 с целевой функцией, равной 9, отсекаем. Наконец, на 6-м шаге ветвим вершину 4, вершины-потомки 10 с нижней границей, равной 12, и 11 с целевой функцией, равной 10, отсекаем. Решением является маршрут, соответствующий конечной вершине 7 – x_1, x_2, x_5, x_6 с суммой длин ребер, равной 7.

Оценочная функция в виде суммы длин ребер, уже включенных в формируемый маршрут, является минимальным значением нижней границы.

Конструирование оценочных функций является сложной творческой задачей. Оно должно базироваться на анализе сущности задачи (в данном случае процесса построения маршрута) и рассмотренных ранее свойств функций, которые используются в качестве верхних и нижних границ. Процесс построения маршрута – это последовательное включение в него ребер до тех пор, пока не придем в конечную вершину. Оценочная функция в конечной вершине должна быть равна значению целевой функции, т.е. сумме длин ребер, составляющих соответствующий маршрут. Следовательно, одной из составляющих оценочной функции будет сумма длин ребер, уже вошедших в формируемый маршрут от корня до рассматриваемой вершины. Нижняя граница для некоторой вершины дерева решений должна быть не больше, а верхняя не меньше значения целевой функции для любой текущей и конечной вершины поддерева, начинающегося в ней. Очевидно, что вторая составляющая нашей оценочной функции, которая должна приблизить нижнюю или верхнюю границу к *значению целевой функции оптимального варианта*, должна удовлетворять этому условию.

Верхняя граница для любой вершины дерева решений будет не меньше целевой функции маршрута максимальной длины, если второй составляющей оценочной функции является сумма ребер максимальной длины *среди ребер маршрутов для каждого уровня поддерева решений с корнем в этой вершине*. Очевидно, что суммирование должно выполняться от уровня рассматриваемой вершины до самой нижней конечной вершины указанных маршрутов. Данная оценка действительно является верхней границей, так как в основу положено предположение, что на каждом шаге в строящийся маршрут включается ребро максимальной длины, тогда как в действительности это может быть ребро другого маршрута, и суммирование выполняется до максимально возможного конечного количества ребер маршрутов, проходящих через данную вершину. Например, для вершины 1 дерева решений, показанного на рис. 4.7, б, имеем:

$$F_B = 2 + \max\{4, 3\} + \max\{6, 4, 2\} + \max\{2\} = 14.$$

Сконструированная для оценки верхней границы функция не возрастает и ее значение в конечных вершинах равно значению целевой функции для соответствующего варианта маршрута.

Компонентами второй составляющей для нижней границы должны быть ребра минимальной длины для каждого уровня дерева решений среди ребер маршрутов, проходящих через эту вершину. Суммируя длины этих ребер от уровня рассматриваемой вершины до самой верхней, т.е. ближайшей к корню, конечной вершины этих маршрутов, получим вторую составляющую. Очевидно, что оценка не убывает и в конечной вершине равна значению целевой функции.

Сказанное выше не означает, что для получения таких оценок необходимо построить полностью дерево решений и сформировать все варианты маршрутов. Однако вычисление оценок этого типа по сравнению с оценкой в виде суммы длин ребер, включенных в фрагмент маршрута, требует выполнения довольно сложных преобразований. Рассмотрим пример вычисления нижней границы для вершины 2 (см. рис. 4.7, *б*). Пусть граф G задан аналитически через отображение множества X в себя, т.е. в форме $G(X, F)$. Здесь $X = \{x_i \mid i = \overline{\mathbf{1,6}}\}$ – множество вершин графа, $F = \{Fx_i \mid i = \overline{\mathbf{1,6}}\}$ – прямое отображение множества X в себя, т. е. $Fx_i = X_i \subset X$ – множество тех вершин $x_j \in X$, в которые заходят дуги из вершины x_i. Каждому отображению Fx_i поставим во взаимно-однозначное соответствие вектор длин соответствующих дуг $L_i = \{l(x_i, x_j)\} \mid \forall x_j \in Fx_i\}$. Тогда первая компонента нижней границы вершины 2 определяется как $l(x_1, x_4) = 5$. Для определения второй составляющей находим $Fx_4 = \{x_5, x_6\}$ и выбираем минимум из $l(x_4, x_5)$ и $l(x_4, x_6)$. Таким образом, получим нижнюю границу для вершины 2, равную 9.

На рис. 4.7, б над вершинами (слева или справа) показана пара чисел в виде *а* / *в*, где *а* – верхняя граница; *в* – нижняя граница, рассчитанные по сконструированным выше оценочным функциям.

На рис. 4.9 представлено дерево решения задачи поиска маршрута максимальной длины по методу ветвей и границ при ветвлении в ширину (выбор вершины ветвления по максимуму верхней границы), а на рис. 4.10 –

при последовательном ветвлении, причем на рис. 4.10, а порядок построения ветвей справа налево, а на рис. 4.10, б – слева направо.

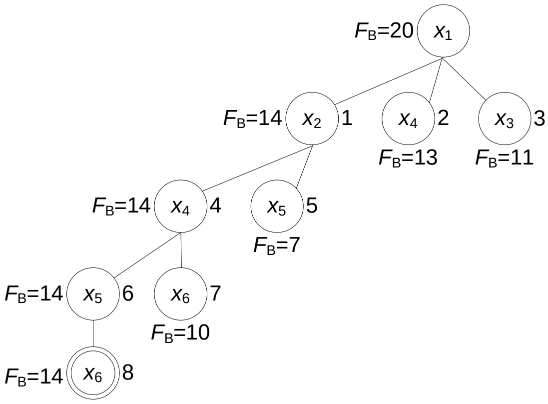
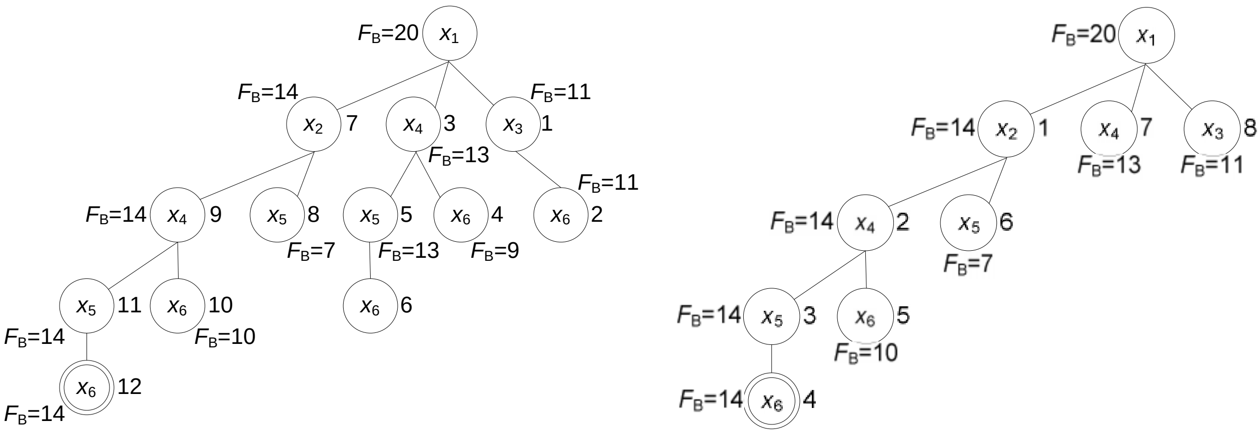


Рис. 4.9



а *б*

Рис. 4.10

На рис. 4.11 показано дерево решения при поиске маршрута минимальной длины по методу ветвей и границ при ветвлении в ширину. Выбор вершины ветвления выполняется по минимуму нижней границы, сконструированной выше двухкомпонентной функции (сравните с рис. 4.8). На рис. 4.12 представлены деревья решения той же задачи при последовательном построении ветвей (на рис. 4.12, а – справа налево и на рис. 4.12, б – слева направо).

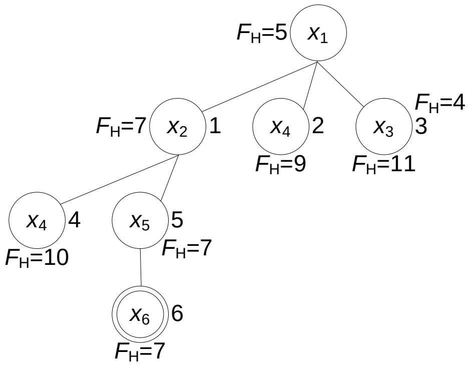
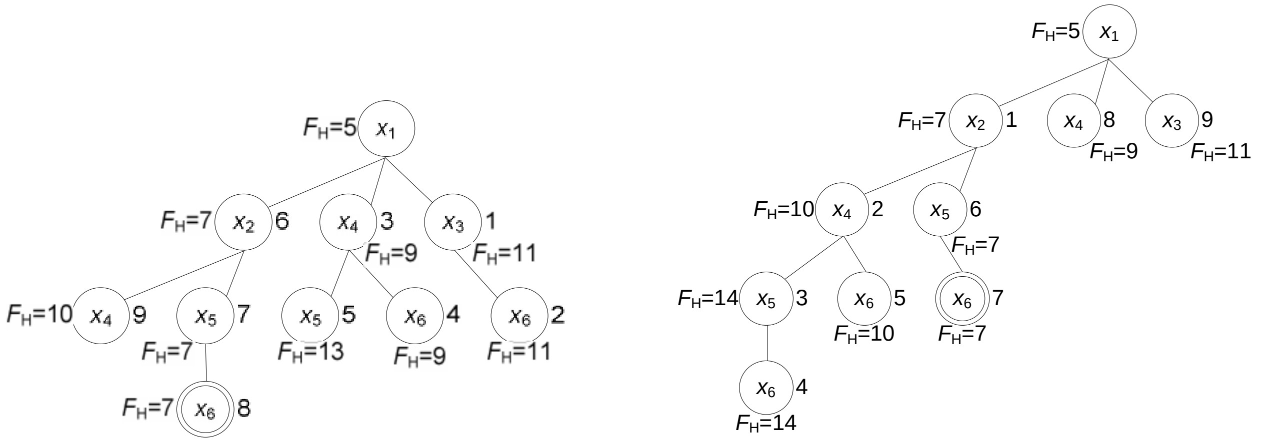


Рис. 4.11



$F_н$=5 x_1
$F_н$=7 x_2 6
x_4 3 $F_н$=9
x_3 1 $F_н$=11
$F_н$=10 x_4 9
x_5 7 $F_н$=7
x_5 5 $F_н$=13
x_6 4 $F_н$=9
x_6 2 $F_н$=11
$F_н$=7 x_6 8
$F_н$=5 x_1
$F_н$=7 x_2 1
x_4 8 $F_н$=9
x_3 9 $F_н$=11
$F_н$=10 x_4 2
x_5 6 $F_н$=7
$F_н$=14 x_5 3
x_6 5 $F_н$=10
x_6 7 $F_н$=7
x_6 4 $F_н$=14

а *б*

Рис. 4.12

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Таха Х. Введение в исследование операций: Учебное пособие. – М.: Изд. дом «Вильямс», 2007.

2. Аттетков А.В. Методы оптимизации: Учебник для студ. втузов /Под ред. В.С. Зарубина, А.П. Крищенко. – М.: МГТУ им. Н.Э. Баумана, 2003.

3. Корнеенко В.П. Методы оптимизации: Учебник. – М.: Высшая школа, 2007.

4. Минько Э.В., Минько А.Э. Методы прогнозирования и исследования операций: Учебное пособие. – М.: Финансы и статистика: ИНФРА-М, 2010.

5. Вентцель Е.С. Исследование операций: задачи, принципы, методология: Учебное пособие. – М.: Кнорус, 2010.

6. Гончаров В.А.. Методы оптимизации: Учебное пособие для студ. вузов. – М.: Юрайт: Высшее образование, 2010.

7. Пантелеев А.В. Методы оптимизации в примерах и задачах: Учебное пособие для студ. втузов. – М.: Высшая школа, 2005.

8. Давыдов Э.Г. Элементы исследования операций: Учебное пособие для студ. вузов – М.: Кнорус, 2010.

9. Овчинников В.А. Алгоритмизация комбинаторно-оптимизационных задач при проектировании ЭВМ и систем. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2001.

СОДЕРЖАНИЕ

Учебное издание

Гладков Леонид Анатольевич
Гладкова Надежда Викторовна

Методы оптимизации

Часть 2
Конспект лекций

Ответственный за выпуск	Гладков Л.А.
Редактор	Проценко И.А.
Корректор	Надточий З.И.

ЛР № 020565 от 23.06.1997 г.	Подписано к печати
Формат 60 x 84 1/16.	Бумага офсетная
Печать офсетная.	Усл. п.л. – 7,0. Уч. - изд.л. – 6,8.
Заказ №	Тираж 300 экз.

«С»

Издательство ЮФУ
ГСП 17А, Таганрог, 28, Некрасовский, 44

Типография ЮФУ
Таганрог, 28, Некрасовский, 44