

Министерство образования и науки Российской Федерации
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Б. Ю. ЛЕМЕШКО

МЕТОДЫ ОПТИМИЗАЦИИ

Утверждено
Редакционно-издательским советом университета
в качестве конспекта лекций

НОВОСИБИРСК
2009

УДК 519.852(075.8)
Л 442

Рецензенты: д-р техн. наук, проф. *A.A. Попов*;
д-р физ.-мат. наук, проф. *B.A. Селезнев*

Работа подготовлена на кафедре прикладной математики
для студентов III курса ФПМИ (направление 010500 –
Прикладная математика и информатика, специальности 010503 –
Математическое обеспечение и администрирование
информационных систем)

Лемешко Б.Ю.
Л 442 Методы оптимизации : конспект лекций / Б.Ю. Лемешко. –
Новосибирск : Изд-во НГТУ, 2009. – 156 с.

ISBN 978-5-7782-1202-2

Курс лекций рассчитан на один семестр и предназначен для студентов
ФПМИ, но может быть полезен и студентам других специальностей. Настоя-
щее издание должно помочь студентам овладеть прикладными методами оп-
тимизации.

УДК 519.852(075.8)

ISBN 978-5-7782-1202-2

© Лемешко Б.Ю., 2009
© Новосибирский государственный
технический университет, 2009

ОГЛАВЛЕНИЕ

Введение	5
1. Методы одномерного поиска	7
1.1. Метод дихотомии	7
1.2. Метод золотого сечения.....	9
1.3. Метод Фибоначчи.....	10
Контрольные вопросы.....	12
2. Прямые методы поиска	12
2.1. Алгоритм Гаусса.....	13
2.2. Алгоритм Хука и Дживса	14
2.3. Алгоритм Розенброка.....	16
2.4. Симплексный метод Нелдера–Мида или поиск по деформируемому многограннику	20
2.5. Метод Пауэлла и сопряженные направления	24
2.5.1. Обоснование применения сопряженных направлений в алгоритмах оптимизации	24
2.5.2. Алгоритм метода Пауэлла.....	30
Контрольные вопросы.....	35
3. Методы первого порядка.....	36
3.1. Алгоритм наискорейшего спуска	36
3.2. Метод сопряженных градиентов	38
3.3. Многопараметрический поиск	42
Контрольные вопросы.....	43
4. Методы второго порядка (метод Ньютона)	44
Контрольные вопросы.....	47
5. Методы переменной метрики	47
5.1. Введение.....	47
5.2. Метод Бройдена	50
5.3. Метод Дэвидона–Флетчера–Пауэлла	52
5.4. Алгоритмы Пирсона	54
5.5. Проективный алгоритм Ньютона–Рафсона.....	55
5.6. Методы Гринштадта и Гольдфарба	55
5.7. Алгоритм Флетчера	56
5.8. Алгоритмы с аппроксимацией матрицы Гессе	58
Контрольные вопросы	60
6. Методы штрафных функций	60
Контрольные вопросы.....	64
7. Статистические методы поиска	64
7.1. Введение	64
7.2. Простой случайный поиск	66

7.3. Простейшие алгоритмы направленного случайного поиска	68
7.3.1. Алгоритм парной пробы.....	68
7.3.2. Алгоритм наилучшей пробы.....	69
7.3.3. Метод статистического градиента.....	70
7.3.4. Алгоритм наилучшей пробы с направляющим гиперквадратом	72
7.4. Алгоритмы глобального поиска	73
Контрольные вопросы.....	76
8. Линейное программирование.....	77
8.1. Основные определения и теоремы	77
8.2. Основная теорема линейного программирования	81
8.3. Симплекс-метод.....	83
8.3.1. Введение в симплекс-метод	83
8.3.2. Алгоритм симплекс-метода	87
8.3.3. Вырожденность в задачах линейного программирования	92
8.4. Двойственность задач линейного программирования	94
8.4.1. Понятие двойственной задачи	94
8.4.2. Преобразования при решении прямой и двойственной задач	95
8.4.3. Теоремы двойственности линейного программирования	98
8.4.4. Метод последовательного уточнения оценок	102
Контрольные вопросы.....	105
9. Методы решения транспортной задачи.....	106
9.1. Формулировка классической транспортной задачи	106
9.2. Метод северо-западного угла	107
9.3. Метод минимального элемента	108
9.4. Теорема, лежащая в основе метода потенциалов	109
9.5. Алгоритм метода потенциалов.....	112
9.6. О вырожденности транспортной задачи	117
Контрольные вопросы.....	118
10. Транспортная задача с ограничениями.....	119
10.1. Постановка задачи	119
10.2. Метод потенциалов для определения оптимального плана.....	120
10.3. Построение опорного плана	122
Контрольные вопросы.....	127
11. Транспортная задача по критерию времени	127
Контрольные вопросы.....	131
12. Задача о максимальном потоке в транспортной сети.....	131
12.1. Постановка задачи	131
12.2. Алгоритм построения максимального потока в транспортной сети...	134
Контрольные вопросы.....	143
13. Параметрическое линейное программирование.....	143
13.1. Постановка задачи	143
13.2. Алгоритм	145
Контрольные вопросы.....	152
Библиографический список	153

ВВЕДЕНИЕ

Методы оптимизации занимаются построением оптимальных решений для математических моделей. В эту дисциплину не входит само построение математических моделей. Но именно вид модели определяет метод или методы, используемые для построения оптимального решения.

В большинстве случаев математическую модель объекта можно представить в виде целевой функции $f(\bar{x})$ или критерия оптимальности (иногда без ограничений), которую нужно максимизировать или минимизировать. Таким образом необходимо найти максимум или минимум поставленной задачи, причем $\bar{x} \in D$ – области возможных значений $\bar{x} = (x_1, x_2, \dots, x_n)^T$. Как правило, область допустимых значений D задается. Тогда задача формулируется следующим образом:

$$f(\bar{x}) \rightarrow \max_{\bar{x} \in D} (\min_{\bar{x} \in D}), \quad (1)$$

или по другому

$$f(\bar{x}) \rightarrow \max (\min)$$

при

$$\bar{x} \in D.$$

Область допустимых значений D определяется системой линейных или нелинейных ограничений, накладываемых на \bar{x} :

$$D = \bar{x} \mid q_j(\bar{x}) \leq q_j^0; j = 1, m . \quad (2)$$

В реальных задачах ограничения на область возможных значений переменных модели отсутствуют чрезвычайно редко, потому что, как правило, переменные бывают связаны с некоторым ограниченным ре-

сурсом. Но все-таки с задачами без ограничений сталкиваются. Это бывает в условиях «неограниченных» ресурсов или в условиях, не накладывающих ограничений на переменные задачи. В таком случае мы имеем безусловную задачу, задачу без ограничений:

$$f(\bar{x}) \rightarrow \max_{\bar{x}} (\min_{\bar{x}}) . \quad (3)$$

Сложность задачи зависит от вида критерия $f(\bar{x})$ и функций $q_j(\bar{x})$, определяющих допустимую область. Функции могут быть линейными и нелинейными, непрерывными или могут принимать дискретные значения. Область возможных значений может быть выпуклой и невыпуклой, несвязной, представлять собой дискретное множество точек. В зависимости от этого задачи могут быть одноэкстремальными или многоэкстремальными, могут использоваться одни или другие методы поиска решения.

Например, если функции $f(\bar{x})$ и $q_j(\bar{x})$ линейны, имеем задачу линейного программирования и можем использовать для поиска решения методы линейного программирования (варианты симплекс-метода).

Если функции $f(\bar{x})$ и $q_j(\bar{x})$ нелинейны, используем методы нелинейного программирования. Если при этом минимизируем выпуклую $f(\bar{x})$ при выпуклых функциях $q_j(\bar{x})$, то знаем, что задача одноэкстремальна (выпуклое нелинейное программирование).

Если $q_j(\bar{x})$ линейна, а минимизируемая $f(\bar{x})$ представляет собой квадратичную выпуклую функцию, можем использовать алгоритмы квадратичного программирования.

При минимизации вогнутой функции на выпуклой области можем столкнуться с многоэкстремальностью задачи и необходимостью поиска глобального экстремума.

Если на переменные, входящие в задачу, наложено требование целочисленности или дискретности, то используются методы дискретного программирования, среди которых наиболее хорошо разработаны методы решения линейных дискретных задач.

Если система ограничений отсутствует и $f(\bar{x})$ представляет собой нелинейную функцию, то для решения задачи (3), т. е. для определения минимума или максимума этой функции, используются различные алгоритмы поиска. В зависимости от информации о функции $f(\bar{x})$, ис-

пользуемой в алгоритме, могут применяться прямые методы поиска, методы поиска первого или второго порядка.

Прямые методы поиска или методы нулевого порядка – это методы, в которых при поиске экстремума используется информация только о самой функции и не используется информация о ее производных. Плюсом таких методов является возможность оптимизации функций, аналитическое представление которых неизвестно, т. е. эти функции определены только алгоритмически.

Методы первого порядка при поиске решения используют не только информацию о самой функции, но и о ее производных первого порядка. К таким методам относятся различные градиентные алгоритмы.

Методы второго порядка при поиске решения используют информацию о самой функции и о ее производных первого и второго порядка. Сюда относятся метод Ньютона и его модификации.

1. МЕТОДЫ ОДНОМЕРНОГО ПОИСКА

Как правило, реально мы сталкиваемся с необходимостью решения многомерных задач. И очень редко практическая задача изначально является одномерной. Для многомерных задач мы используем многомерные методы. Но на этапах поиска в многомерных методах почти обязательно сталкиваются с задачами одномерной минимизации в направлении некоторого вектора.

Существует множество методов поиска минимума или максимума функции на отрезке. Наиболее известны из них методы дихотомии (деления отрезка пополам), золотого сечения и Фибоначчи [1]. В каждом из этих методов последовательно сокращается интервал, содержащий точку минимума.

1.1. МЕТОД ДИХОТОМИИ

Предполагается, что минимизируемая функция $f(x)$ унимодальна на отрезке $[a_0, b_0]$, и необходимо найти минимум этой функции на

заданном отрезке с некоторой точностью ε . Вычисляем две точки согласно следующим соотношениям:

$$x_1 = \frac{a_0 + b_0 - \delta}{2} \text{ и } x_2 = \frac{a_0 + b_0 + \delta}{2},$$

где $\delta < \varepsilon$ (рис. 1.1). И в каждой из найденных точек вычисляем значения функции: $f(x_1)$ и $f(x_2)$.

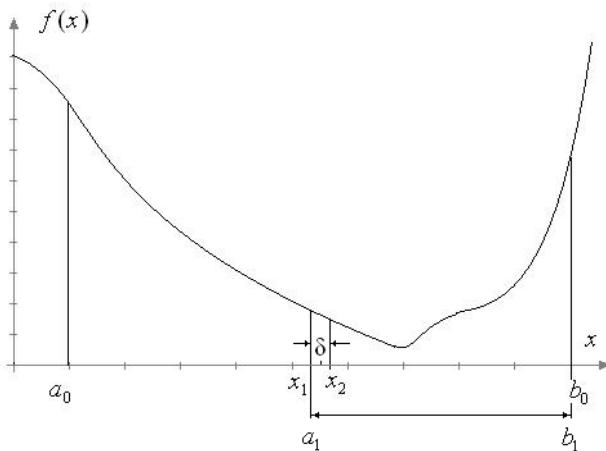


Рис. 1.1. Метод дихотомии

Затем сокращаем интервал неопределенности и получаем интервал $[a_1, b_1]$ следующим образом. Если $f(x_1) < f(x_2)$, то $a_1 = a_0$ и $b_1 = x_2$. В противном случае, если $f(x_1) > f(x_2)$, то $a_1 = x_1$ и $b_1 = b_0$.

Далее по аналогичным формулам на этом интервале вычисляем следующую пару точек x_1 и x_2 . С помощью найденных точек определяем новый интервал неопределенности.

Поиск заканчивается, если длина интервала неопределенности $[a_k, b_k]$ на текущей итерации k становится меньше заданной точности:

$$|b_k - a_k| < \varepsilon.$$

В данном методе на каждой итерации минимизируемая функция $f(x)$ вычисляется дважды, а интервал неопределенности сокращается практически в два раза (при малых $\delta < \varepsilon$).

1.2. МЕТОД ЗОЛОТОГО СЕЧЕНИЯ

Этот метод позволяет найти минимум унимодальной функции на заданной области $[a_0, b_0]$, как правило, с меньшими вычислительными затратами, чем метод дихотомии.

На первой итерации находим две точки по следующим формулам:

$$x_1 = a_0 + \frac{3 - \sqrt{5}}{2}(b_0 - a_0) = a_0 + 0.381966\ 011(b_0 - a_0),$$

$$\begin{aligned} x_2 &= b_0 + \frac{\sqrt{5} - 3}{2}(b_0 - a_0) = b_0 - 0.381966\ 011(b_0 - a_0) = \\ &= a_0 + 0.6180\ 033\ 989(b_0 - a_0) \end{aligned}$$

и вычисляем значения функции $f(x_1)$ и $f(x_2)$.

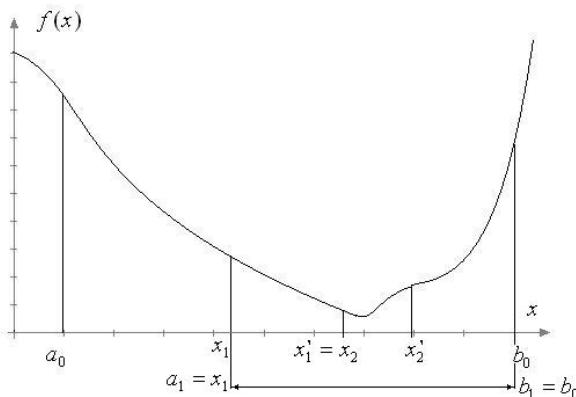


Рис. 1.2. Методы золотого сечения, дихотомии и Фибоначчи

Обратим внимание, что на первой итерации находим две точки и дважды вычисляем $f(x)$.

Сокращаем интервал неопределенности.

- 1) Если $f(x_1) < f(x_2)$, то $a_1 = a_0$, $b_1 = x_2$, $x_2 = x_1$.
- 2) В противном случае, если $f(x_1) > f(x_2)$, то $a_1 = x_1$, $b_1 = b_0$, $x_1 = x_2$.

На последующих итерациях производим расчет только той точки и значение функции в ней, которые необходимо обновить: в случае 1) вычисляем новое значение x_1 и $f(x_1)$; в случае 2) x_2 и $f(x_2)$.

Поиск прекращается при выполнении условия $|b_k - a_k| < \varepsilon$.

На i -й итерации интервал неопределенности сокращается до величины $0.618\ 003\ 399(b_{i-1} - a_{i-1})$. Это меньше, чем в два раза, но зато всего один раз вычисляем значение $f(x)$ в новой точке.

1.3. МЕТОД ФИБОНАЧЧИ

Последовательность чисел Фибоначчи подчиняется соотношению:

$$F_{n+2} = F_{n+1} + F_n,$$

где $n = 1, 2, 3, \dots$ и $F_1 = F_2$. Она имеет вид 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597...

С помощью индукции можно показать, что n -е число Фибоначчи вычисляется по формуле Бинэ:

$$F_n = \frac{\left[\frac{1+\sqrt{5}}{2} \right]^n - \left[\frac{1-\sqrt{5}}{2} \right]^n}{\sqrt{5}},$$

где $n = 1, 2, 3, \dots$

Из этой формулы видно, что при больших значениях n выполняется соотношение

$$F_n \approx \frac{\left[\frac{1+\sqrt{5}}{2} \right]^n}{\sqrt{5}},$$

так что числа Фибоначчи с увеличением n растут очень быстро.

Сам алгоритм метода Фибоначчи очень похож на алгоритм метода золотого сечения. На начальном интервале вычисляются точки по следующим формулам:

$$x_1 = a_0 + \frac{F_n}{F_{n+2}}(b_0 - a_0) \text{ и } x_2 = a_0 + \frac{F_{n+1}}{F_{n+2}}(b_0 - a_0) = a_0 + b_0 - x_1.$$

Интервал неопределенности сокращается точно так же, как в методе золотого сечения (см. рис. 1.2). И на новой итерации вычисляется только одна новая точка и значение функции в ней.

На k -й итерации получаем точку с минимальным значением, которая совпадает с одной из точек, вычисляемых по формулам

$$x'_1 = a_k + \frac{F_{n-k+1}}{F_{n-k+3}}(b_k - a_k) = a_k + \frac{F_{n-k+1}}{F_{n+2}}(b_0 - a_0),$$

$$x'_2 = a_k + \frac{F_{n-k+2}}{F_{n-k+3}}(b_k - a_k) = a_k + \frac{F_{n-k+2}}{F_{n+2}}(b_0 - a_0)$$

и расположенных на отрезке $[a_k, b_k]$ симметрично относительно его середины.

Нетрудно заметить, что при $k = n$ точки

$$x'_1 = a_n + \frac{F_1}{F_{n+2}}(b_0 - a_0) \text{ и } x'_2 = a_n + \frac{F_2}{F_{n+2}}(b_0 - a_0)$$

совпадают и делят отрезок $[a_n, b_n]$ пополам.

$$\text{Следовательно, } \frac{b_n - a_n}{2} = \frac{b_0 - a_0}{F_{n+2}} < \varepsilon.$$

Отсюда можно выбрать n из условия $\frac{b_0 - a_0}{\varepsilon} < F_{n+2}$.

Таким образом, это условие позволяет до начала работы алгоритма найти число итераций, необходимое для определения минимума с точностью ε при начальной величине интервала $[a_0, b_0]$.

С ростом n из-за того, что F_n / F_{n+2} – бесконечная десятичная дробь, возможны «искажение» метода и потеря интервала с точкой минимума (вследствие погрешностей вычислений).

Следует также отметить, что при практическом применении метод золотого сечения по эффективности, скорости сходимости и точности получаемого решения практически не уступает методу Фибоначчи. Алгоритмически же реализация метода золотого сечения является более простой.

При реализации многомерных алгоритмов используются не только рассмотренные выше методы. Применяются различные эвристические алгоритмы, используется интерполяция (аппроксимация) минимизируемой функции более простой с последующим поиском минимума этой интерполирующей функции, например, квадратичной. Зачастую это оказывается очень эффективным приемом [2].

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для каких функций эффективно применение методов типа дихотомии, золотого сечения, Фибоначчи?
2. Количество вычислений минимизируемой функции на одну итерацию в методе дихотомии. В методе золотого сечения. В методе Фибоначчи.
3. Редукция интервала неопределенности в методе дихотомии. В методе золотого сечения. В методе Фибоначчи.

2. ПРЯМЫЕ МЕТОДЫ ПОИСКА

Прямые методы, или методы нулевого порядка, не требуют знания целевой функции в явном виде. Они не требуют регулярности и непрерывности целевой функции и наличия производных. Это является существенным достоинством при решении сложных технических и экономических задач.

При реализации прямых методов значительно сокращается этап подготовки решения задачи, так как нет необходимости в определении первых и вторых производных. К прямым методам относится целый ряд алгоритмов, которые отличаются по своей эффективности. Такие методы носят в основном эвристический характер.

Прямые методы предназначены для решения безусловных задач оптимизации вида

$$\min_{\bar{x} \in E^n} f(\bar{x}).$$

2.1. АЛГОРИТМ ГАУССА

Это простейший алгоритм [2], заключающийся в том, что на каждом шаге (каждой итерации) минимизация осуществляется только по одной компоненте вектора переменных \bar{x} .

Пусть нам дано начальное приближение $\bar{x}^0 = [x_1^0, x_2^0, \dots, x_n^0]^T$. На первой итерации находим значение минимума функции при изменяющейся первой координате и фиксированных остальных компонентах, т. е.

$$x_1^1 = \arg \min_{x_1} f(x_1, x_2^0, \dots, x_n^0).$$

В результате получаем новую точку $\bar{x}^1 = [x_1^1, x_2^0, \dots, x_n^0]$. Далее из точки \bar{x}^1 ищем минимум функции, изменяя вторую координату и считая фиксированными все остальные координаты (рис. 2.1). В результате находим значение

$$x_2^1 = \arg \min_{x_2} f(x_1^1, x_2, x_3^0, \dots, x_n^0)$$

и новую точку $\bar{x}^2 = [x_1^1, x_2^1, x_3^0, \dots, x_n^0]$. Продолжая процесс, после n шагов получаем точку $\bar{x}^n = [x_1^1, x_2^2, \dots, x_n^n]$, начиная с которой процесс поиска возобновляется по первой переменной.

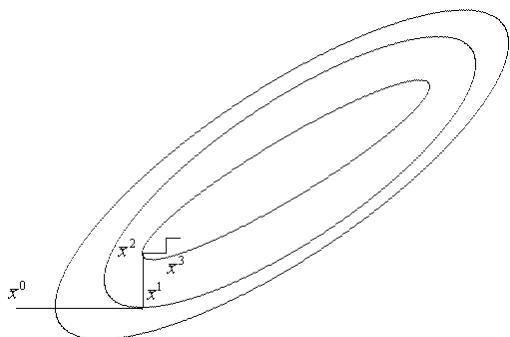


Рис. 2.1. Пример траектории спуска в алгоритме Гаусса

В качестве условий прекращения поиска можно использовать следующие два критерия:

- 1) $\|f(\bar{x}^{k+1}) - f(\bar{x}^k)\| \leq \varepsilon_0;$
- 2) $|x_i^{k+1} - x_i^k| \leq \varepsilon_i, \forall i.$

Метод очень прост, но не очень эффективен. Проблемы могут возникнуть, когда линии уровня сильно вытянуты и «эллипсоиды» ориентированы, например, вдоль прямых вида $x_1 = x_2$. В подобной ситуации поиск быстро застревает на дне такого «оврага», а если начальное приближение оказывается на оси «эллипсоида», то процесс так и останется в этой точке.

Хорошие результаты получаются в тех случаях, когда целевая функция представляет собой выпуклую сепарабельную функцию вида

$$f(\bar{x}) = \sum_{i=1}^n f_i(x_i).$$

2.2. АЛГОРИТМ ХУКА И ДЖИВСА

В данном алгоритме [2] предлагается логически простая стратегия поиска, в которой используются априорные сведения о топологии

функции и в то же время отвергается уже устаревшая информация об этой функции. В интерпретации Вуда алгоритм включает два основных этапа:

- 1) исследующий поиск вокруг базисной точки \bar{x}^k ;
- 2) поиск по образцу, т. е. в направлении, выбранном для минимизации.

В первую очередь задаются начальная точка поиска \bar{x}^0 и начальное приращение (шаг) $\Delta\bar{x}^0$. После этого начинается исследующий поиск.

Исследующий поиск. Делаем пробный шаг по переменной x_1 , т. е. определяем точку $x_1^0 + \Delta x_1^0$ и вычисляем значение функции в точке $\bar{x}' = x_1^0 + \Delta x_1^0, x_2^0, \dots, x_n^0$.

Если значение функции в данной точке больше, чем значение функции $f(\bar{x}^0)$, то делаем пробный шаг по этой же переменной, но в противоположном направлении. Если значение функции и в точке $\bar{x}'' = (x_1^0 - \Delta x_1^0, x_2^0, \dots, x_n^0)$ больше чем $f(\bar{x}^0)$, то оставляем точку x_1^0 без изменений. Иначе заменяем точку \bar{x}^0 на \bar{x}' или на \bar{x}'' в зависимости от того, где значение функции меньше исходного. Из вновь полученной точки делаем пробные шаги по оставшимся координатам, используя тот же самый алгоритм (рис. 2.2).

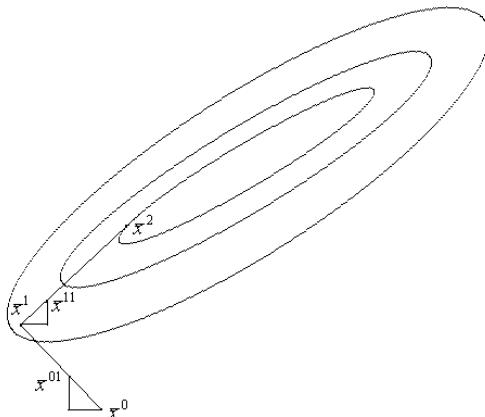


Рис. 2.2. Пример траектории спуска в алгоритме Хука и Дживса

Если в процессе исследующего поиска не удается сделать ни одного удачного пробного шага, то $\Delta\bar{x}$ необходимо скорректировать (уменьшить). После этого вновь переходим к исследующему поиску.

Если в процессе исследующего поиска сделан хотя бы один удачный пробный шаг, то переходим к поиску по образцу.

Поиск по образцу. После исследующего поиска мы получаем точку \bar{x}^{01} . Направление $\bar{x}^{01} - \bar{x}^0$ определяет направление, в котором функция уменьшается. Поэтому проводим минимизацию функции в указанном направлении, решая задачу

$$\min_{\lambda} f \quad \bar{x}^0 + \lambda(\bar{x}^{01} - \bar{x}^0) .$$

В поиске по «образцу» величина шага по каждой переменной пропорциональна величине шага на этапе исследующего поиска. Если сможем сделать удачный шаг в поиске по «образцу», то в результате найдем новое приближение $\bar{x}^1 = \bar{x}^0 + \lambda^0(\bar{x}^{01} - \bar{x}^0)$, где

$$\lambda^0 = \arg \min_{\lambda} f(\bar{x}^0 + \lambda(\bar{x}^{01} - \bar{x}^0)).$$

Из точки \bar{x}^1 начинаем новый исследующий поиск и т.д.

Возможны модификации алгоритма, в которых в процессе исследующего поиска ищется минимум по каждой переменной или в процессе поиска по образцу не ищется минимум функции, а просто делается шаг в заданном найденном направлении с фиксированным значением параметра λ .

2.3. АЛГОРИТМ РОЗЕНБРОКА

Этот итерационный метод имеет некоторое сходство с алгоритмом Хука и Дживса. Метод Розенброка называют также методом вращающихся координат. Он заметно эффективнее предыдущих методов, особенно при минимизации функций овражного типа.

Общая идея метода [2] заключается в том, что выбирается система ортогональных направлений $\bar{S}_1^0, \bar{S}_2^0, \dots, \bar{S}_n^0$, в каждом из которых последовательно ищется минимальное значение, после чего система направлений поворачивается так, чтобы одна из осей совпала с направлением полного перемещения, а остальные были ортогональны между собой.

Пусть \bar{x}^0 – вектор начального приближения; $\bar{S}_1^0, \bar{S}_2^0, \dots, \bar{S}_n^0$ – система ортогональных направлений. На первой итерации это может быть ортонормированная система координат. Начиная с \bar{x}^0 , последовательно минимизируем функцию $f(\bar{x})$ в направлениях, соответствующих $\bar{S}_1^0, \bar{S}_2^0, \dots, \bar{S}_n^0$, находя последовательные приближения:

$$\bar{x}_1^0 = \bar{x}_0^0 + \lambda_1 \bar{S}_1^0, \text{ где } \lambda_1 = \arg \min_{\lambda} f(\bar{x}_0^0 + \lambda \bar{S}_1^0),$$

....

$$\bar{x}_n^0 = \bar{x}_{n-1}^0 + \lambda_n \bar{S}_n^0, \text{ где } \lambda_n = \arg \min_{\lambda} f(\bar{x}_{n-1}^0 + \lambda \bar{S}_n^0).$$

Следующая итерация начнется с точки $\bar{x}^1 = \bar{x}_n^0$.

Если не изменить систему направлений, то мы будем иметь алгоритм Гаусса. Поэтому после завершения очередного k -го этапа вычисляем новые направления поиска. Ортогональные направления поиска поворачиваются так, чтобы они оказались вытянутыми вдоль оврага («хребта») и, таким образом, будет исключаться взаимодействие переменных ($x_i x_j$). Направления поиска вытягиваются вдоль главных осей квадратичной аппроксимации целевой функции.

Рассмотрим k -ю итерацию алгоритма Розенброка (рис. 2.3). В результате минимизации по каждому из ортогональных направлений мы имеем на данной итерации систему параметров $\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k$, с помощью которой определим систему векторов $\bar{A}_1^k, \bar{A}_2^k, \dots, \bar{A}_n^k$, вычисляемых по формулам следующего вида:

$$\bar{A}_1^k = \lambda_1^k \bar{S}_1^k + \lambda_2^k \bar{S}_2^k + \dots + \lambda_n^k \bar{S}_n^k;$$

$$\bar{A}_2^k = \lambda_2^k \bar{S}_2^k + \dots + \lambda_n^k \bar{S}_n^k;$$

$\dots;$

$$\bar{A}_n^k = \lambda_n^k \bar{S}_n^k.$$

С помощью полученной системы векторов $\bar{A}_1^k, \bar{A}_2^k, \dots, \bar{A}_n^k$ строим новую систему ортогональных направлений $\bar{S}_1^{k+1}, \bar{S}_2^{k+1}, \dots, \bar{S}_n^{k+1}$. Причем первый вектор направляется так, чтобы он совпал с направлением общего перемещения на k -м шаге, а остальные направления получаются с помощью процедуры ортогонализации Грама–Шмидта:

$$\bar{S}_1^{k+1} = \frac{\bar{A}_1^k}{\|\bar{A}_1^k\|};$$

$$\bar{B}_2^k = \bar{A}_2^k - \left[\bar{A}_2^k \begin{smallmatrix} \top \\ \bar{S}_1^{k+1} \end{smallmatrix} \right] \bar{S}_1^{k+1};$$

$$\bar{S}_2^{k+1} = \frac{\bar{B}_2^k}{\|\bar{B}_2^k\|}; \quad (1)$$

$$\bar{B}_l^k = \bar{A}_l^k - \sum_{m=1}^{l-1} \left[\bar{A}_l^{k^T} \bar{S}_m^{k+1} \right] \bar{S}_m^{k+1};$$

$$\bar{S}_l^{k+1} = \frac{\bar{B}_l^k}{\|\bar{B}_l^k\|}; \quad l = 2, \dots, n.$$

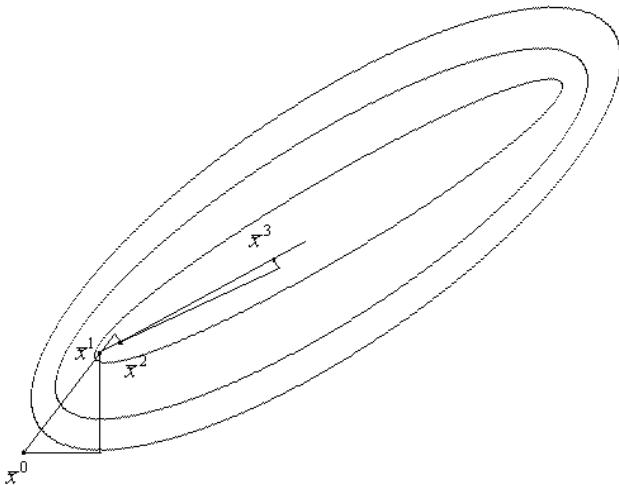


Рис. 2.3. Пример траектории спуска в алгоритме Розенброка

Для эффективной работы алгоритма необходимо, чтобы ни один из векторов системы $\bar{S}_1^{k+1}, \bar{S}_2^{k+1}, \dots, \bar{S}_n^{k+1}$ не стал нулевым вектором. Для этого в алгоритме следует располагать параметры $\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k$ в порядке убывания по абсолютному значению, т. е. $|\lambda_1^k| > |\lambda_2^k| > \dots > |\lambda_n^k|$. Тогда, если любые m из λ_i^k обращаются в нуль, то отыскиваются

новые направления по соотношениям (1) только для тех $(n-m)$ направлений, для которых $\lambda_i^k \neq 0$. Оставшиеся же m направлений остаются неизменными: $\bar{S}_i^{k+1} = \bar{S}_i^k$, $i = \overline{(n-m+1), n}$. Так как эти $(n-m)$ векторов взаимно ортогональны, $\lambda_i^k = 0$, $i = \overline{(n-m+1), n}$, первые $(n-m)$ векторов не будут иметь составляющих в направлениях \bar{S}_i^{k+1} , $i = \overline{(n-m+1), n}$. А поскольку эти последние направления взаимно ортогональны, все направления являются взаимно ортогональными.

Палмером [2] было показано, что \bar{B}_{j+1}^k и $\|\bar{B}_{j+1}^k\|$ пропорциональны λ_j^k $\left(\text{при условии, что } \sum_{i=j}^n \lambda_i^k \neq 0 \right)$. Следовательно, при вычислении $\bar{S}_j^{k+1} = \bar{B}_j^k / \|\bar{B}_j^k\|$ величина λ_j^k сокращается, и, таким образом, направление \bar{S}_j^{k+1} остается определенным, если даже $\lambda_j^k = 0$. Имея это в виду, Палмер предложил для вычисления \bar{S}_j^{k+1} следующие соотношения:

$$\begin{aligned} \bar{A}_i^k &= \sum_{j=i}^n \lambda_j^k \bar{S}_j^k, \quad i = 1, \dots, n, \\ \bar{S}_i^{k+1} &= \frac{\bar{A}_i^k \|\bar{A}_{i-1}^k\|^2 - \bar{A}_{i-1}^k \cdot \|\bar{A}_i^k\|^2}{\|\bar{A}_{i-1}^k\| \|\bar{A}_i^k\| \left[\|\bar{A}_{i-1}^k\|^2 - \|\bar{A}_i^k\|^2 \right]^{1/2}}, \quad i = 2, \dots, n, \\ \bar{S}_1^{k+1} &= \frac{\bar{A}_1^k}{\|\bar{A}_1^k\|}. \end{aligned}$$

Критерии останова алгоритма могут быть стандартными (т.е. описанными в предыдущих алгоритмах прямых методов).

2.4. СИМПЛЕКСНЫЙ МЕТОД НЕЛДЕРА–МИДА ИЛИ ПОИСК ПО ДЕФОРМИРУЕМОМУ МНОГОГРАННИКУ

В данном методе [2] в процессе поиска осуществляется работа с регулярными симплексами. Регулярные многогранники в пространстве E^n называются **симплексами**. Для $n = 2$ регулярный симплекс представляет собой равносторонний треугольник, при $n = 3$ – тетраэдр и т. д.

Координаты вершин регулярного симплекса в n -мерном пространстве могут быть определены следующей матрицей D , в которой столбцы представляют собой вершины симплекса, пронумерованные от 1 до $(n+1)$, а строки – координаты вершин, $i = \overline{1, n}$. Матрица имеет размерность $n \times (n+1)$:

$$D = \begin{bmatrix} 0 & d_1 & d_2 & \dots & d_2 \\ 0 & d_2 & d_1 & \dots & d_2 \\ 0 & d_2 & d_2 & \dots & d_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & d_2 & d_2 & \dots & d_1 \end{bmatrix}_{n \times (n+1)},$$

где

$$d_1 = \frac{t}{n\sqrt{2}}(\sqrt{n+1} + n - 1); \quad d_2 = \frac{t}{n\sqrt{2}(\sqrt{n+1} - 1)};$$

t – расстояние между вершинами.

В самом простом виде симплексный алгоритм заключается в следующем. Строится регулярный симплекс. Из вершины, в которой $f(\bar{x})$ максимальна (точка 1, рис. 2.4), проводится проектирующая прямая через центр тяжести симплекса. Затем точка 1 исключается и строится новый *отраженный* симплекс из оставшихся старых точек и одной новой, расположенной на проектирующей прямой на надлежащем расстоянии от центра тяжести.

Продолжение этой процедуры, в которой каждый раз исключается вершина, где целевая функция максимальна, а также использование правил уменьшения размера симплекса и предотвращения циклическо-

го движения в окрестности экстремума, позволяют достаточно эффективно определять минимум для «хороших» функций. Но для овражных функций такой поиск неэффективен.

Представление об идее алгоритма дает рис. 2.4.

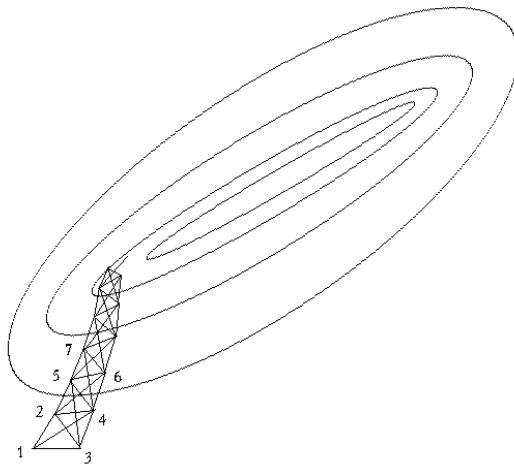


Рис. 2.4. Траектория спуска в простейшем симплексном алгоритме

В симплексном алгоритме Нелдера и Мида минимизация функций n переменных осуществляется с использованием деформируемого многогранника.

Будем рассматривать k -ю итерацию алгоритма. Пусть $\bar{x}_i^k = [x_{i1}^k, x_{i2}^k, \dots, x_{in}^k]^T$, $i = 1, \dots, (n+1)$, является i -й вершиной в E^n на k -м этапе поиска, $k = 0, 1, 2, \dots$, и пусть значения целевой функции в этой вершине $f(\bar{x}_i^k)$. Отметим вершины с минимальным и максимальным значениями. И обозначим их следующим образом:

$$f(\bar{x}_h^k) = \max f(\bar{x}_1^k), \dots, f(\bar{x}_{n+1}^k) ;$$

$$f(\bar{x}_l^k) = \min f(\bar{x}_1^k), \dots, f(\bar{x}_{n+1}^k) .$$

Многогранник в E^n состоит из $n+1$ вершин $\bar{x}_1^k, \bar{x}_2^k, \dots, \bar{x}_{n+1}^k$. Обозначим через \bar{x}_{n+2}^k центр тяжести вершин без точки \bar{x}_h^k с максимальным значением функции. Координаты этого центра вычисляются по формуле

$$x_{n+2,j}^k = \frac{1}{n} \left[\sum_{i=1}^{n+1} x_{i,j}^k - x_{h,j}^k \right], \quad j = 1, \dots, n. \quad (2)$$

Начальный многогранник обычно выбирается в виде регулярного симплекса (с вершиной в начале координат). Можно начало координат поместить в центр тяжести. Процедура отыскания вершин в E^n , в которых $f(\bar{x})$ имеет лучшее значение, состоит из следующих операций: 1) *отражения*; 2) *растяжения*; 3) *сжатия*; 4) *редукции*.

1. Отражение. Отражение представляет собой проектирование точки \bar{x}_h^k через центр тяжести \bar{x}_{n+2}^k в соответствии со следующим соотношением

$$\bar{x}_{n+3}^k = \bar{x}_{n+2}^k + \alpha (\bar{x}_{n+2}^k - \bar{x}_h^k), \quad (3)$$

где $\alpha > 0$ – коэффициент отражения.

Вычисляем значение функции в найденной точке $f(\bar{x}_{n+3}^k)$. Если значение функции в данной точке $f(\bar{x}_{n+3}^k) \geq f(\bar{x}_h^k)$, то переходим к четвертому пункту алгоритма – операции *редукции*.

Если $f(\bar{x}_{n+3}^k) < f(\bar{x}_h^k) \wedge f(\bar{x}_{n+3}^k) < f(\bar{x}_l^k)$, то выполняем операцию *растяжения*.

В противном случае, если $f(\bar{x}_{n+3}^k) < f(\bar{x}_h^k) \wedge f(\bar{x}_{n+3}^k) \geq f(\bar{x}_l^k)$, то выполняется операция *сжатия*.

2. Растяжение. Эта операция заключается в следующем. Если $f(\bar{x}_{n+3}^k) < f(\bar{x}_l^k)$ (меньше минимального значения на k -м этапе), то вектор $\bar{x}_{n+3}^k - \bar{x}_{n+2}^k$ растягивается в соответствии с соотношением

$$\bar{x}_{n+4}^k = \bar{x}_{n+2}^k + \gamma (\bar{x}_{n+3}^k - \bar{x}_{n+2}^k), \quad (4)$$

где $\gamma > 0$ – коэффициент растяжения.

Если $f(\bar{x}_{n+4}^k) < f(\bar{x}_l^k)$, то \bar{x}_l^k заменяется на \bar{x}_{n+4}^k и процедура продолжается с операции *отражения* при $k = k + 1$. В противном случае \bar{x}_l^k заменяется на \bar{x}_{n+3}^k , и также переходим к операции *отражения*.

3. Сжатие. Если $f(\bar{x}_{n+3}^k) > f(\bar{x}_i^k)$ для $\forall i \neq h$, то вектор $\bar{x}_h^k - \bar{x}_{n+2}^k$ сжимается в соответствии с формулой

$$\bar{x}_{n+5}^k = \bar{x}_{n+2}^k + \beta \cdot (\bar{x}_h^k - \bar{x}_{n+2}^k),$$

где $0 < \beta < 1$ – коэффициент сжатия. После этого точка \bar{x}_h^k заменяется на \bar{x}_{n+5}^k , и переходим к операции *отражения* с $k = k + 1$. Заново ищется \bar{x}_h^{k+1} .

4. Редукция. Если $f(\bar{x}_{n+3}^k) > f(\bar{x}_h^k)$, то все векторы $\bar{x}_i^k - \bar{x}_l^k$, где $i = \overline{1, (n+1)}$, уменьшаются в два раза с отсчетом от точки \bar{x}_l^k по формуле

$$\bar{x}_i^k = \bar{x}_l^k + 0.5 (\bar{x}_i^k - \bar{x}_l^k), \quad i = \overline{1, (n+1)}$$

и осуществляется переход к операции *отражения* (на начало алгоритма с $k = k + 1$).

В качестве критерия останова могут быть взяты те же правила, что и в остальных алгоритмах. Можно также использовать критерий останова следующего вида:

$$\left\{ \frac{1}{n+1} \sum_{i=1}^{n+1} [f(\bar{x}_i^k) - f(\bar{x}_{n+2}^k)]^2 \right\}^{1/2} < \varepsilon.$$

Выбор коэффициентов α, β, γ обычно осуществляется эмпирически. После того как многогранник подходящим образом промасштабирован, его размеры должны поддерживаться неизменными до тех пор, пока изменения в топологии задачи не потребуют многогранника другой формы. Чаще всего рекомендуют $\alpha = 1$, $0.4 \leq \beta \leq 0.6$, $2 \leq \gamma \leq 3$.

2.5. МЕТОД ПАУЭЛЛА И СОПРЯЖЕННЫЕ НАПРАВЛЕНИЯ

2.5.1. ОБОСНОВАНИЕ ПРИМЕНЕНИЯ СОПРЯЖЕННЫХ НАПРАВЛЕНИЙ В АЛГОРИТМАХ ОПТИМИЗАЦИИ

Метод Пауэлла [2] относится к прямым методам (методам нулевого порядка). Этим методом наиболее эффективно осуществляется минимизация функций, близких к квадратичным. На каждой итерации алгоритма поиск осуществляется вдоль системы сопряженных направлений.

Два направления поиска \bar{S}_i, \bar{S}_j называются **сопряженными**, если $\bar{S}_i^T H \bar{S}_j = 0, i \neq j$,

$$\bar{S}_i^T H \bar{S}_i \geq 0, i = j,$$

где H – положительно определенная квадратная матрица.

В методе Пауэлла $H = \nabla^2 f(\bar{x}^k)$ – матрица вторых частных производных. Идеи метода Пауэлла относятся к квадратичной функции $f(\bar{x})$.

Основная идея заключается в следующем. Если на каждом этапе поиска определяется минимум квадратичной функции $f(\bar{x})$ вдоль каждого из p ($p < n$) -сопряженных направлений и если затем в каждом из направлений делается шаг до минимальной точки, то полное перемещение от начала до шага с номером p сопряжено ко всем поднаправлениям поиска.

Применение сопряженных направлений лежит в основе ряда алгоритмов.

Пусть $f(\bar{x})$ – квадратичная функция и процесс минимизации начинается в точке \bar{x}^0 с начальным направлением \bar{S}^1 . Для удобства возьмем этот вектор единичным, т. е. $\bar{S}^1^T \bar{S}^1 = 1$. Тогда вектор

$\bar{x}^1 = \bar{x}^0 + \lambda^1 \cdot \bar{S}^1$ и длина шага λ^1 определяется из условия минимальности функции в данном направлении т. е.

$$\lambda^1 = \arg \min_{\lambda} f(\bar{x}^0 + \lambda \bar{S}^1).$$

Для квадратичной функции

$$\frac{df(\bar{x}^0 + \lambda \bar{S}^1)}{d\lambda} = \nabla^T f(\bar{x}^0) \bar{S}^1 + (\bar{S}^1)^T \nabla^2 f(\bar{x}^0) \lambda \bar{S}^1 = 0, \quad (5)$$

и, таким образом, оптимальное значение λ на первом шаге определяется в соответствии с соотношением

$$\lambda^1 = \frac{-\nabla^T f(\bar{x}^0) \bar{S}^1}{(\bar{S}^1)^T \nabla^2 f(\bar{x}^0) \bar{S}^1} = -\frac{\nabla^T f(\bar{x}^0) \bar{S}^1}{(\bar{S}^1)^T H \bar{S}^1}, \quad (6)$$

где $H = \nabla^2 f(\bar{x}^k)$.

Из точки \bar{x}^1 процесс минимизации должен осуществляться в другом сопряженном направлении \bar{S}^2 , при этом в силу сопряженности

$$\bar{S}^2{}^T H \bar{S}^1 = 0.$$

Отметим, что квадратичная функция может быть представлена в виде

$$f(\bar{x}) = a + b^T \bar{x} + \frac{1}{2} \bar{x}^T H \bar{x},$$

где положительно определенная матрица $H = \nabla^2 f(\bar{x})$.

В общем случае система n линейно независимых направлений поиска $\bar{S}^1, \bar{S}^2, \dots, \bar{S}^n$ называется **сопряженной** по отношению к некоторой положительно определенной матрице H , если

$$(\bar{S}^i)^T H \bar{S}^j = 0, \quad 1 \leq i \neq j \leq n.$$

Так как сопряженные направления линейно независимы, то любой вектор в пространстве E^n можно выразить через систему $\bar{S}^1, \bar{S}^2, \dots, \bar{S}^n$ следующим образом:

$$\bar{V} = \sum_{j=1}^n v^j \bar{S}^j,$$

где

$$v^j = \frac{\bar{S}^j^T H \bar{V}}{\bar{S}^j^T H \bar{S}^j}. \quad (7)$$

Известная формула, примем ее на веру.

Для некоторой матрицы H всегда существует, по крайней мере, одна система из n взаимно сопряженных направлений, так как сами собственные векторы матрицы H представляют собой такую систему.

Отметим, что для квадратичной функции справедливо следующее соотношение, которое потребуется в дальнейшем:

$$H^{-1} = \sum_{j=1}^n \frac{\bar{S}^j \bar{S}^j^T}{\bar{S}^j^T H \bar{S}^j}. \quad (8)$$

Чтобы убедиться в его справедливости, рассмотрим матрицу $\sum_{j=1}^n \alpha_j \bar{S}^j (\bar{S}^j)^T$. Умножение ее справа на $H \bar{S}^k$ дает

$$\left[\sum_{j=1}^n \alpha_j \bar{S}^j (\bar{S}^j)^T \right] H \bar{S}^k = \alpha_k \bar{S}^k (\bar{S}^k)^T H \bar{S}^k = \bar{S}^k,$$

если положить $\alpha_k = \left[\bar{S}^k^T H \bar{S}^k \right]^{-1}$.

Вообще говоря, справедливо общее правило, заключающееся в том, что если используются сопряженные направления для поиска минимума квадратичной функции $f(\bar{x})$, то эта функция может быть мини-

мизирована за n шагов, по одному в каждом из сопряженных направлений. Более того, порядок использования сопряженных направлений несуществен.

Покажем, что это действительно так. Пусть $f(\bar{x})$ – квадратичная функция и

$$f(\bar{x}) = a + \bar{b}^T \bar{x} + \frac{1}{2} \bar{x}^T H \bar{x},$$

при этом

$$\nabla f(\bar{x}) = \bar{b} + H \bar{x}.$$

В точке минимума

$$\nabla f(\bar{x}^*) = 0,$$

и эта точка

$$\bar{x}^* = -H^{-1}\bar{b}.$$

Заметим, что

$$\nabla^T f(\bar{x}^k) \bar{S}^k = \bar{S}^k{}^T \nabla f(\bar{x}^k).$$

Так как

$$\bar{x}^1 = \bar{x}^0 + \lambda^1 \bar{S}^1, \quad (9)$$

где λ^1 определяется в соответствии с соотношением (6):

$$\lambda^1 = \frac{-\nabla^T f(\bar{x}^0) \bar{S}^1}{\bar{S}^1{}^T \nabla^2 f(\bar{x}^0) \bar{S}^1} = -\frac{\nabla^T f(\bar{x}^0) \bar{S}^1}{\bar{S}^1{}^T H \bar{S}^1},$$

а затем минимум находится в следующем сопряженном направлении по аналогичным формулам

$$\bar{x}^2 = \bar{x}^1 + \lambda^2 \bar{S}^2$$

и так далее, то на n -м шаге имеем

$$\bar{x}^n = \bar{x}^0 + \sum_{i=1}^n \lambda^i \bar{S}^i. \quad (10)$$

На каждом шаге минимизируем одномерную функцию $f(\bar{x}^{i-1} + \lambda^i \bar{S}^i)$ в направлении \bar{S}^i , чтобы получить λ^i . На основании (6) это приводит к следующему выражению

$$\bar{x}^n = \bar{x}^0 - \sum_{i=1}^n \left[\frac{\bar{S}^i {}^T \nabla f(\bar{x}^{i-1})}{\bar{S}^i {}^T H \bar{S}^i} \right] \bar{S}^i. \quad (11)$$

Кроме того,

$$\bar{S}^i {}^T \nabla f(\bar{x}^{i-1}) = \bar{S}^i {}^T H \bar{x}^{i-1} + \bar{b} = \bar{S}^i {}^T \left\{ H \left[\bar{x}^0 + \sum_{k=1}^{i-1} \lambda^k \bar{S}^k \right] + \bar{b} \right\}$$

и получаем

$$\bar{S}^i {}^T \nabla f(\bar{x}^{i-1}) = \bar{S}^i {}^T H \bar{x}^0 + \bar{b},$$

так как все $\bar{S}^i {}^T H \bar{S}^k = 0$, $\forall i \neq k$, вследствие сопряженности \bar{S}^i и \bar{S}^k .

Таким образом,

$$\bar{x}^n = \bar{x}^0 - \sum_{i=1}^n \frac{\bar{S}^i {}^T H \bar{x}^0 + \bar{b}}{\bar{S}^i {}^T H \bar{S}^i} \bar{S}^i. \quad (12)$$

Выразим векторы \bar{x}^0 и $H^{-1}\bar{b}$ через систему сопряженных векторов \bar{S}^i следующим образом (по аналогии с (7)):

$$\bar{x}^0 = \sum_{i=1}^n \frac{\bar{S}^i {}^T H \bar{x}^0 \bar{S}^i}{\bar{S}^i {}^T H \bar{S}^i},$$

$$H^{-1}\bar{b} = \sum_{i=1}^n \frac{\bar{S}^i {}^T H H^{-1}\bar{b} \bar{S}^i}{\bar{S}^i {}^T H \bar{S}^i} = \sum_{i=1}^n \frac{\bar{S}^i {}^T \bar{b} \bar{S}^i}{\bar{S}^i {}^T H \bar{S}^i}.$$

С учетом этого из (12) получим

$$\bar{x}^n = \bar{x}^0 - \bar{x}^0 - H^{-1}\bar{b} = -H^{-1}\bar{b}. \quad (13)$$

Таким образом, точка \bar{x}^n , полученная в результате минимизации квадратичной функции на n -м шаге, совпадает с точкой минимума квадратичной функции $f(\bar{x})$.

Покажем теперь, что для сопряженных направлений, если $f(\bar{x})$ каждый раз минимизируется в сопряженном направлении \bar{S}^j в соответствии с формулой (6), то при этом выполняется следующее равенство:

$$\bar{S}^j \nabla f(\bar{x}^l) = 0, \quad 1 \leq j \leq l-1,$$

при использовании не более чем n направлений, т. е. $\nabla f(\bar{x}^l)$ ортогонален использованным сопряженным направлениям.

Для квадратичной функции $\nabla f(\bar{x}^l) = \bar{b} + H\bar{x}^l$. Следовательно,

$$\nabla f(\bar{x}^l) = \bar{b} + H \left[\bar{x}^k + \sum_{j=k}^{l-1} \lambda^j \bar{S}^j \right] = \bar{b} + H\bar{x}^k + H \sum_{j=k}^{l-1} \lambda^j \bar{S}^j,$$

где \bar{x}^k – произвольная точка, из которой начинается поиск по сопряженным направлениям.

Так как

$$\nabla f(\bar{x}^k) = \bar{b} + H\bar{x}^k,$$

то

$$\nabla f(\bar{x}^l) = \nabla f(\bar{x}^k) + \sum_{j=k}^{l-1} \lambda^j H \bar{S}^j.$$

Умножение этого равенства слева на \bar{S}^{k-1} дает

$$\bar{S}^{k-1} \nabla f(\bar{x}^l) = \bar{S}^{k-1} \nabla f(\bar{x}^k) + \sum_{j=k}^{l-1} \lambda^j \bar{S}^{k-1} H \bar{S}^j.$$

Первый член в правой части $\bar{S}^{k-1}^T \nabla f \bar{x}^k = 0$, так как градиент в точке \bar{x}^k ортогонален направлению предыдущего спуска, если точка получена в результате минимизации функции в этом направлении. Кроме того, все остальные слагаемые под знаком суммы исчезают вследствие сопряженности направлений \bar{S}^{k-1} и \bar{S}^j , и таким образом,

$$\bar{S}^j^T \nabla f \bar{x}^l = 0, \quad 1 \leq j \leq l-1, \quad (14)$$

т. е. градиент в точке \bar{x}^l ортогонален всем использованным ранее сопряженным направлениям.

2.5.2. АЛГОРИТМ МЕТОДА ПАУЭЛЛА

Переход из точки \bar{x}_0^k в точку \bar{x}_n^k на k -м шаге алгоритма Пауэлла осуществляется в соответствии с формулой

$$\bar{x}_n^k = \bar{x}_0^k + \sum_{j=1}^n \lambda_j^k \bar{S}_j^k.$$

При этом последовательно минимизируется исходная функция по сопряженным направлениям $\bar{S}_1^k, \dots, \bar{S}_n^k$. Результатом минимизации по каждому из сопряженных направлений является система параметров $\lambda_1^k, \dots, \lambda_n^k$, при которых функция минимальна в каждом из сопряженных направлений:

$$\lambda_j^k = \arg \min_{\lambda} f(\bar{x}_{j-1}^k + \lambda \bar{S}_j^k),$$

$$\bar{x}_j^k = \bar{x}_{j-1}^k + \lambda_j^k \bar{S}_j^k.$$

Начальную систему используемых направлений можно выбрать параллельной осям системы координат.

В конце каждой итерации алгоритма Пауэлла необходимо выбрать новую систему направлений (часть из которых станет сопряженной), так как если этого не сделать, то получим простой покоординатный поиск.

Построение новой системы базируется на следующей теореме.

Теорема. Если при начальной точке \bar{x}^0 поиска в направлении вектора \bar{S} минимум функции $f(\bar{x})$ находится к точке \bar{x}^a , а при начальной точке $\bar{x}^1 \neq \bar{x}^0$ поиск минимума функции $f(\bar{x})$ в том же направлении \bar{S} приводит к точке \bar{x}^b , то при $f(\bar{x}^b) < f(\bar{x}^a)$ направление $\bar{x}^b - \bar{x}^a$ сопряжено с направлением поиска \bar{S} .

Доказательство. Используя ранее полученные результаты (14), можно записать, что в первом случае

$$\bar{S}^T \nabla f(\bar{x}^a) = \bar{S}^T H \bar{x}^a + \bar{b} = 0,$$

аналогично во втором случае можно записать

$$\bar{S}^T \nabla f(\bar{x}^b) = \bar{S}^T H \bar{x}^b + \bar{b} = 0.$$

Вычитая из второго выражения первое, получим, что

$$\bar{S}^T H (\bar{x}^b - \bar{x}^a) = 0.$$

Следовательно, векторы \bar{S} и $\bar{x}^b - \bar{x}^a$ являются сопряженными.

Эта теорема непосредственно может быть распространена на случай нескольких сопряженных направлений следующим образом. Если, начиная из точки \bar{x}^0 , точка \bar{x}^a определяется после использования при минимизации нескольких сопряженных направлений p ($p < n$) аналогично, если из точки $\bar{x}^1 \neq \bar{x}^0$ точка \bar{x}^b определяется после использования тех же направлений и функция $f(\bar{x})$ минимизируется на каждом шаге, то вектор $\bar{x}^b - \bar{x}^a$ сопряжен ко всем p направлениям.

Рис. 2.5 служит иллюстрацией теоремы.

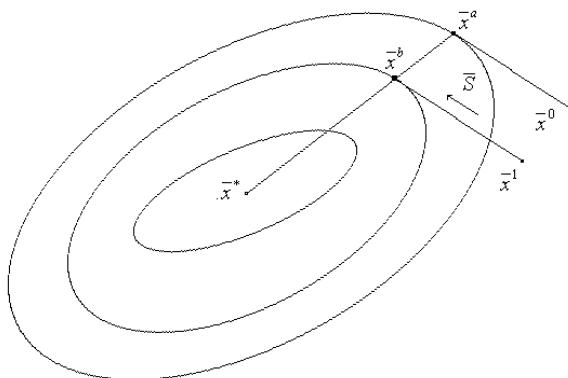


Рис. 2.5. Сопряженные направления для квадратичной функции

Пусть в начальный момент для двумерной задачи поиск осуществляется из точки \bar{x}^0 вдоль направлений, параллельных осям координат: \bar{S}_1^0 и \bar{S}_2^0 . Последовательно были найдены точки \bar{x}_1^0 , \bar{x}_2^0 , \bar{x}_3^0 (рис. 2.6). Таким образом, определили два сопряженных направления, в которых следует вести поиск: \bar{S}_2^0 и $\bar{x}_3^0 - \bar{x}_1^0$.

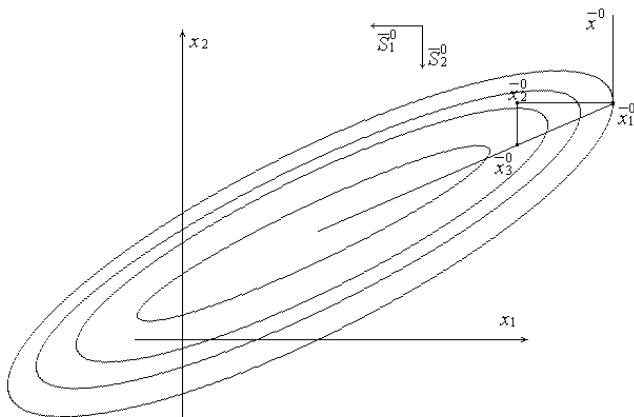


Рис. 2.6. Построение сопряженных направлений в процессе поиска

В системе исходных направлений \bar{S}_1^0 должно быть заменено на $\bar{x}_3^0 - \bar{x}_1^0$, представляющее собой полное перемещение из первого минимума. Направления поиска на следующем этапе:

$$\bar{S}_1^1 = \bar{S}_2^0,$$

$$\bar{S}_2^1 = \bar{x}_3^0 - \bar{x}_1^0.$$

Второй этап начинается с минимизации вдоль направления \bar{S}_2^1 , затем, если это необходимо, выполняется перемещение в направлении \bar{S}_1^1 . Но в случае квадратичной функции двух переменных после минимизации по двум сопряженным направлениям будет достигнута точка минимума.

В общем случае на k -м шаге алгоритма Паузлла используется n линейно независимых направлений поиска. Поиск начинается с точки \bar{x}_0^k и ведется по следующему алгоритму.

1. Начиная с точки \bar{x}_0^k , решается последовательность задач минимизации функции $\min_{\lambda} f(\bar{x}_{j-1}^k + \lambda \bar{S}_j^k)$, $j = \overline{1, n}$, в направлениях $\bar{S}_1^k, \dots, \bar{S}_n^k$. При этом находятся точки $\bar{x}_1^k, \dots, \bar{x}_n^k$, которые минимизируют исходную функцию в заданных направлениях, причем $\bar{x}_1^k = \bar{x}_0^k + \lambda_1 \bar{S}_1^k$, $\bar{x}_2^k = \bar{x}_1^k + \lambda_2 \bar{S}_2^k, \dots, \bar{x}_n^k = \bar{x}_{n-1}^k + \lambda_n \bar{S}_n^k$.

2. Поиск, осуществляемый на первом этапе, может привести к линейно зависимым направлениям, если, например, в одном из направлений \bar{S}^i не удается найти меньшего значения функции. Вследствие этого два направления могут стать коллинеарными. Отсюда вытекает, что в системе сопряженных направлений **не следует** заменять старое направление на новое, если после такой замены направления нового набора становятся линейно зависимыми.

На примере квадратичной функции Паузл показал, что при нормировании направлений поиска в соответствии с соотношением

$$\bar{S}_i^k {}^T H \bar{S}_i^k = 1, \quad i = \overline{1, n},$$

определитель матрицы, столбцы которой представляют собой направления поиска, принимает максимальное значение тогда и только тогда, когда \bar{S}_i^k взаимно сопряжены относительно матрицы H . Он пришел к выводу, что направление полного перемещения на k -м шаге должно заменять предыдущее направление только в том случае, когда заменяющий вектор увеличивает определитель матрицы направлений поиска, так как только в этом случае новый набор направлений будет более эффективным.

Для такой проверки из точки \bar{x}_n^k делается дополнительный шаг в направлении $\bar{x}_n^k - \bar{x}_0^k$, соответствующий полному перемещению на k -м этапе, и получают точку $2\bar{x}_n^k - \bar{x}_0^k$. Для проверки того, что определитель матрицы направлений поиска увеличивается при включении нового направления, делается шаг 3.

3. Обозначим наибольшее уменьшение $f(\bar{x})$ на k -м шаге

$$\Delta^k = \max_{i=1, n} f \bar{x}_{i-1}^k - f \bar{x}_i^k ,$$

а соответствующее направление поиска – через \bar{S}_m^k .

Обозначим:

$$f_1 = f \bar{x}_0^k , \quad f_2 = f \bar{x}_n^k , \quad f_3 = f 2\bar{x}_n^k - \bar{x}_0^k ,$$

где

$$\bar{x}_0^k = \bar{x}_n^{k-1} , \quad \bar{x}_n^k = \bar{x}_{n-1}^k + \lambda_n \bar{S}_n^k = \bar{x}_0^k + \sum_{i=1}^n \lambda_i \bar{S}_i^k .$$

Тогда, если $f_3 \geq f_1$ и (или) $(f_1 - 2f_2 + f_3)(f_1 - f_2 - \Delta^k)^2 \geq 0.5\Delta^k(f_1 - f_3)^2$, то следует использовать на $k+1$ -м этапе те же

направления $\bar{S}_1^k, \dots, \bar{S}_n^k$, что и на k -м этапе, т. е. $\bar{S}_i^{k+1} = \bar{S}_i^k$, $i = \overline{1, n}$, и начать поиск из точки $\bar{x}_0^{k+1} = \bar{x}_n^k$ или из точки $\bar{x}_0^{k+1} = 2\bar{x}_n^k - \bar{x}_0^k = \bar{x}_{n+1}^k$, в зависимости от того, в какой точке функция принимает минимальное значение.

4. Если тест на шаге 3 не прошел, то ищется минимум $f(\bar{x})$ в направлении вектора \bar{S}_{n+1}^k , проведенного из \bar{x}_0^k в \bar{x}_n^k : $\bar{S}_{n+1}^k = \bar{x}_n^k - \bar{x}_0^k$. Точка этого минимума берется в качестве начальной точки на $k+1$ -м этапе. А в системе сопряженных направлений сохраняются все, кроме направления \bar{S}_m^k , которое заменяется на новое направление \bar{S}_{n+1}^k , но новое направление помещается в последний столбец матрицы направлений. На $k+1$ -м этапе будут использоваться направления

$$\left[\bar{S}_1^{k+1}, \bar{S}_2^{k+1}, \dots, \bar{S}_n^{k+1} \right] = \left[\bar{S}_1^k, \bar{S}_2^k, \dots, \bar{S}_{m-1}^k, \bar{S}_{m+1}^k, \dots, \bar{S}_n^k, \bar{S}_{n+1}^k \right].$$

5. Критерий останова. Алгоритм прерывается, если изменение по каждой переменной оказывается меньше заданной точности по соответствующей переменной или $\|\bar{x}_n^k - \bar{x}_0^k\| \leq \varepsilon$.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Алгоритм Гаусса, его достоинства и недостатки, критерии останова.
2. Алгоритм Хука и Дживса, этапы алгоритма.
3. Алгоритм Розенброка, как осуществляется поворот системы координат? Для чего используется ортогонализация Грама–Шмидта?
4. Симплексный метод Нелдера–Мида, идея метода, этапы алгоритма поиска по деформируемому многограннику.
5. Что дает применение сопряженных направлений в алгоритмах оптимизации и для каких функций? Теорема, лежащая в основе алгоритма Паулла.
6. Алгоритм Паулла, построение сопряженных направлений.

3. МЕТОДЫ ПЕРВОГО ПОРЯДКА

К методам первого порядка относятся алгоритмы, в которых в процессе поиска кроме информации о самой функции используется информация о производных первого порядка. К группе таких методов относятся различные градиентные методы.

3.1. АЛГОРИТМ НАИСКОРЕЙШЕГО СПУСКА

Градиент функции в любой точке показывает направление наибольшего локального увеличения $f(\bar{x})$. Поэтому при поиске минимума $f(\bar{x})$ следует двигаться в направлении, противоположном направлению градиента $\nabla f(\bar{x})$ в данной точке, т. е. в направлении *наискорейшего спуска*.

Итерационная формула процесса наискорейшего спуска [2–4] имеет вид

$$\bar{x}^{k+1} = \bar{x}^k - \lambda^k \nabla f(\bar{x}^k) ,$$

или

$$\bar{x}^{k+1} = \bar{x}^k - \lambda^k \frac{\nabla f(\bar{x}^k)}{\|\nabla f(\bar{x}^k)\|} = \bar{x}^k + \lambda^k \bar{S}^k .$$

Очевидно, что в зависимости от выбора параметра λ траектории спуска будут существенно различаться (рис. 3.1). При большом значении λ траектория спуска будет представлять собой колебательный процесс, а при слишком больших λ процесс может расходиться. При выборе малых λ траектория спуска будет плавной, но и процесс будет сходиться очень медленно.

Обычно λ выбирают из условия

$$\lambda^k = \arg \min_{\lambda} f(\bar{x}^k + \lambda \bar{S}^k) ,$$

решая одномерную задачу минимизации с использованием некоторого метода. В этом случае получаем алгоритм наискорейшего спуска.

Если λ определяется в результате одномерной минимизации, то градиент в точке очередного приближения будет ортогонален направлению предыдущего спуска $\nabla f \bar{x}^{k+1} \perp \bar{S}^k$.

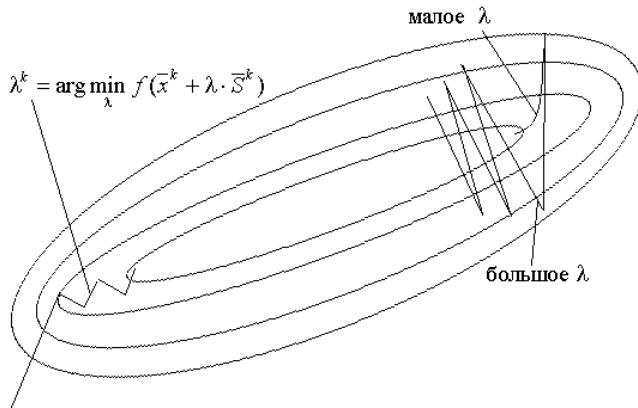


Рис. 3.1. Траектории спуска в зависимости от величины шага

Вообще говоря, процедура наискорейшего спуска может закончиться в стационарной точке любого типа, в которой $\nabla f \bar{x} = \bar{0}$. Поэтому следует проверять, не завершился ли алгоритм в седловой точке.

Эффективность алгоритма зависит от вида минимизируемой функции. Алгоритм наискорейшего спуска сойдется за одну итерацию при любом начальном приближении для функции $f \bar{x} = x_1^2 + x_2^2$ (рис. 3.2). Но сходимость будет очень медленной, например, в случае функции вида $f \bar{x} = x_1^2 + 100x_2^2$. В тех ситуациях, когда линии уровня минимизируемой функции представляют собой прямолинейный или, хуже того, криволинейный овраг, эффективность алгоритма оказывается очень низкой.



Рис. 3.2. Траектории спуска в зависимости от вида функций

Очевидно, что хорошие результаты может давать предварительное масштабирование функций.

Процесс наискорейшего спуска обычно быстро сходится вдали от точки экстремума и медленно в районе экстремума. Поэтому метод наискорейшего спуска нередко используют в комбинации с другими алгоритмами.

3.2. МЕТОД СОПРЯЖЕННЫХ ГРАДИЕНТОВ

В алгоритме наискорейшего спуска на каждом этапе поиска используется только текущая информация о функции $f(\bar{x}^k)$ и градиенте $\nabla f(\bar{x}^k)$. В алгоритмах сопряженных градиентов [2–5] используется информация о поиске на предыдущих этапах спуска.

Направление поиска \bar{S}^k на текущем шаге k строится как линейная комбинация наискорейшего спуска $-\nabla f(\bar{x}^k)$ на этом шаге и направлений спуска $\bar{S}^0, \bar{S}^1, \dots, \bar{S}^{k-1}$ на предыдущих шагах. Веса в линейной комбинации выбираются таким образом, чтобы сделать эти направления сопряженными. В этом случае квадратичная функция будет минимизироваться за n шагов алгоритма [2].

При выборе весов используется только текущий градиент и градиент в предыдущей точке.

В начальной точке \bar{x}^0 направление спуска $\bar{S}^0 = -\nabla f(\bar{x}^0)$:

$$\bar{x}^1 = \bar{x}^0 + \lambda^0 \bar{S}^0, \quad (1)$$

где λ^0 выбирается из соображений минимальности целевой функции в данном направлении

$$\lambda^0 = \arg \min_{\lambda} f(\bar{x}^0 + \lambda \bar{S}^0).$$

Новое направление поиска

$$\bar{S}^1 = -\nabla f(\bar{x}^1) + \omega_1 \bar{S}^0, \quad (2)$$

где ω_1 выбирается так, чтобы сделать направления \bar{S}^1 и \bar{S}^0 сопряженными по отношению к матрице H :

$$\bar{S}^0{}^T H \bar{S}^1 = 0. \quad (3)$$

Для квадратичной функции справедливы соотношения:

$$f(\bar{x}) = f(\bar{x}^0) + \nabla^T f(\bar{x}^0) \Delta \bar{x} + \frac{1}{2} \Delta \bar{x}^T \nabla^2 f(\bar{x}^0) \Delta \bar{x},$$

где $\Delta \bar{x} = \bar{x} - \bar{x}^0$,

$$\nabla f(\bar{x}) = \nabla f(\bar{x}^0) + \nabla^2 f(\bar{x}^0) \Delta \bar{x}.$$

Если положить $\bar{x} = \bar{x}^1$, тогда $\bar{x}^1 - \bar{x}^0 = \lambda^0 \bar{S}^0$ и

$$\nabla f(\bar{x}^1) - \nabla f(\bar{x}^0) = \nabla^2 f(\bar{x}^0) \bar{x}^1 - \bar{x}^0 = H \lambda^0 \bar{S}^0. \quad (4)$$

Воспользуемся (4), чтобы исключить \bar{S}^0 из уравнения (3). Для квадратичной функции $H = H^T$, поэтому, транспонировав (4) и умножив справа на H^{-1} , получим

$$\left[\nabla f(\bar{x}^1) - \nabla f(\bar{x}^0) \right]^T H^{-1} = (\bar{x}^1 - \bar{x}^0)^T H^T H^{-1}$$

и далее

$$\bar{S}^0{}^T = \frac{\bar{x}^1 - \bar{x}^0}{\lambda^0} = \frac{\left[\nabla f(\bar{x}^1) - \nabla f(\bar{x}^0) \right]^T H^{-1}}{\lambda^0}.$$

Следовательно, для сопряженности \bar{S}^0 и \bar{S}^1

$$\begin{aligned} & \left[\nabla f(\bar{x}^1) - \nabla f(\bar{x}^0) \right]^T H^{-1} H \bar{S}^1 = \\ & = \left[\nabla f(\bar{x}^1) - \nabla f(\bar{x}^0) \right]^T \left[-\nabla f(\bar{x}^1) + \omega_1 \bar{S}^0 \right] = 0. \end{aligned}$$

Вследствие изложенных ранее свойств сопряженности все перекрестные члены исчезают. Учитывая, что $\bar{S}^0 = -\nabla f \bar{x}^0$ и, следовательно,

$$-\nabla^T f \bar{x}^1 \nabla f \bar{x}^1 + \omega_1 \nabla^T f \bar{x}^0 \nabla f \bar{x}^0 = 0,$$

получим для ω_1 следующее соотношение:

$$\omega_1 = \frac{\nabla^T f \bar{x}^1 \nabla f \bar{x}^1}{\nabla^T f \bar{x}^0 \nabla f \bar{x}^0} = \frac{\|\nabla f \bar{x}^1\|^2}{\|\nabla f \bar{x}^0\|^2}. \quad (5)$$

Направление поиска \bar{S}^2 строится в виде линейной комбинации векторов $\nabla f \bar{x}^2$, \bar{S}^0 , \bar{S}^1 , причем так, чтобы оно было сопряженным с \bar{S}^1 .

Если распространить сделанные выкладки на \bar{S}^2 , \bar{S}^3, \dots , опуская их содержание и учитывая, что $\bar{S}^k \nabla^T f \bar{x}^{k+1} = 0$ приводит к $\nabla^T f \bar{x}^k \nabla f \bar{x}^{k+1} = 0$, можно получить общее выражение для ω_k :

$$\omega_k = \frac{\|\nabla f \bar{x}^k\|^2}{\|\nabla f \bar{x}^{k-1}\|^2}. \quad (6)$$

Все весовые коэффициенты, предшествующие ω_k , что особенно интересно, оказываются нулевыми.

Полностью алгоритм описывается такой последовательностью действий (рис. 3.3):

1. В точке начального приближения \bar{x}^0 вычисляется $\bar{S}^0 = -\nabla f \bar{x}^0$.
2. На k -м шаге с помощью одномерного поиска в направлении \bar{S}^k определяется минимум функции, т. е. решается задача

$$\lambda^k = \arg \min_{\lambda} f(\bar{x}^k + \lambda \bar{S}^k)$$

и находится очередное приближение $\bar{x}^{k+1} = \bar{x}^k + \lambda^k \bar{S}^k$.

3. Вычисляется $f(\bar{x}^{k+1})$ и $\nabla f(\bar{x}^{k+1})$.

4. Определяется направление $\bar{S}^{k+1} = -\nabla f(\bar{x}^{k+1}) + \omega_{k+1} \bar{S}^k$.

5. Алгоритм заканчивается, если $\|\bar{S}^k\| < \varepsilon$, где ε – заданная величина.

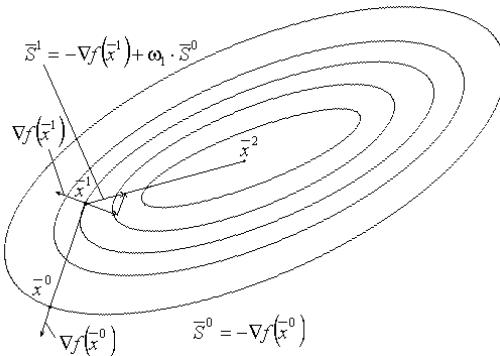


Рис. 3.3. Траектория спуска метода сопряженных градиентов для квадратичной функции

После $n+1$ итераций ($k = n$), если не произошел останов алгоритма, процедура циклически повторяется с заменой \bar{x}^0 на \bar{x}^{n+1} и возвратом на первый пункт алгоритма. Если исходная функция является квадратичной, то $(n+1)$ -е приближение даст точку экстремума данной функции. Описанный алгоритм с построением ω_k по формулам (6) соответствует методу сопряженных градиентов **Флетчера–Ривса** [2].

В модификации **Полака–Рибьера** (Пшеничного) метод сопряженных градиентов отличается только вычислением

$$\omega_k = \frac{\nabla^T f(\bar{x}^k) [\nabla f(\bar{x}^k) - \nabla f(\bar{x}^{k-1})]}{\nabla^T f(\bar{x}^{k-1}) \bar{S}^{k-1}}. \quad (7)$$

В случае квадратичных функций обе модификации примерно эквивалентны. Если функции произвольные, заранее ничего сказать нельзя: где-то эффективнее может оказаться один алгоритм, где-то – другой.

3.3. МНОГОПАРАМЕТРИЧЕСКИЙ ПОИСК

Миль и Кентрелл [2] предложили метод поиска, основанный на использовании двух подбираемых параметров для минимизации $f(\bar{x})$ в каждом направлении поиска. В этом алгоритме последовательность действий определяется формулой

$$\bar{x}^{k+1} = \bar{x}^k - \lambda_0^k \nabla f \bar{x}^k + \lambda_1^k \Delta \bar{x}^{k-1}, \quad (8)$$

где $\Delta \bar{x}^{k-1} = \bar{x}^k - \bar{x}^{k-1}$.

На каждом шаге решается задача минимизации по двум параметрам:

$$\min_{\lambda_0, \lambda_1} f \bar{x}^k - \lambda_0 \nabla f \bar{x}^k + \lambda_1 \Delta \bar{x}^{k-1}.$$

После чего находится очередное приближение по формуле (8). При этом можно показать, что

$$\nabla^T f \bar{x}^k \nabla f \bar{x}^{k+1} = 0,$$

$$\nabla^T f \bar{x}^{k+1} \Delta \bar{x}^{k+1} = 0$$

и

$$\nabla^T f \bar{x}^{k+1} \Delta \bar{x}^k = 0.$$

На первом шаге $\Delta \bar{x}^{k-1} = 0$, а \bar{x}^0 должно быть задано. На k -м шаге:

1) вычисляют \bar{x}^k , $\nabla f \bar{x}^k$ и $\Delta \bar{x}^{k-1} = \bar{x}^k - \bar{x}^{k-1}$;

2) с помощью одного из эффективных методов, например, метода Ньютона находят с требуемой точностью λ_0^k и λ_1^k ;

3) по соотношению (8) вычисляют \bar{x}^{k+1} и переходят к пункту 1;

4) каждый ($n+1$)-й шаг начинается с $\Delta\bar{x}^{k-1} = 0$;

5) процесс заканчивается, когда $|\Delta f(\bar{x}^k)| < \varepsilon$.

На квадратичных функциях алгоритм по эффективности близок к методу сопряженных градиентов.

Крэгг и Леви [2] распространяли данный метод на случай большего числа параметров. На каждом шаге очередное приближение находится как

$$\bar{x}^{k+1} = \bar{x}^k - \lambda_0 \nabla f(\bar{x}^k) + \sum_{i=1}^m \lambda_i \Delta\bar{x}^{i-1}$$

при $m \leq n-1$, а следовательно, на каждом шаге при минимизации $f(\bar{x})$ в заданном направлении решается задача вида

$$\min_{\lambda_0, \lambda_1, \dots, \lambda_m} f\left(\bar{x}^k - \lambda_0 \nabla f(\bar{x}^k) + \sum_{i=1}^m \lambda_i \Delta\bar{x}^{i-1}\right).$$

Достоинства и недостатки такого подхода очевидны.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что собой представляют градиентные методы?
2. Алгоритм наискорейшего спуска. Выбор величины шага в алгоритме наискорейшего спуска.
3. Метод сопряженных градиентов Флетчера–Ривса. Для каких функций алгоритм сходится за n шагов? Что должно быть предусмотрено в алгоритме при минимизации произвольных функций?
4. Чем отличается алгоритм метода сопряженных градиентов Полака–Рибьера?
5. Основная идея многопараметрического поиска. Его достоинства. Недостатки.

4. МЕТОДЫ ВТОРОГО ПОРЯДКА (МЕТОД НЬЮТОНА)

В методах второго порядка при поиске минимума используют информацию о функции и ее производных до второго порядка включительно. К этой группе относят метод Ньютона и его модификации [2].

В основе метода лежит квадратичная аппроксимация $f(\bar{x})$, которую можно получить, отбрасывая в рядах Тейлора члены третьего и более высокого порядка:

$$f(\bar{x}) \approx f(\bar{x}^k) + \nabla^T f(\bar{x}^k)(\bar{x} - \bar{x}^k) + \frac{1}{2} (\bar{x} - \bar{x}^k)^T \nabla^2 f(\bar{x}^k)(\bar{x} - \bar{x}^k), \quad (1)$$

где $\nabla^2 f(\bar{x}^k) = H(\bar{x}^k)$ – матрица Гессе, представляющая собой квадратную матрицу вторых частных производных $f(\bar{x})$ в точке \bar{x}^k .

Направление поиска \bar{s}^k в методе Ньютона определяется следующим образом. Если заменить в выражении (1) \bar{x} на \bar{x}^{k+1} и обозначить $\Delta\bar{x}^k = \bar{x}^{k+1} - \bar{x}^k$, то получим

$$f(\bar{x}^{k+1}) \approx f(\bar{x}^k) + \nabla^T f(\bar{x}^k) \Delta\bar{x}^k + \frac{1}{2} (\Delta\bar{x}^k)^T \nabla^2 f(\bar{x}^k) \Delta\bar{x}^k. \quad (2)$$

Минимум функции $f(\bar{x})$ в направлении $\Delta\bar{x}^k$ определяется дифференцированием $f(\bar{x})$ по каждой из компонент $\Delta\bar{x}$ и приравниванием к нулю полученных выражений

$$\nabla f(\bar{x}^k) + \nabla^2 f(\bar{x}^k) \Delta\bar{x}^k = \bar{0}. \quad (3)$$

Это приводит к

$$\Delta\bar{x}^k = -[\nabla^2 f(\bar{x}^k)]^{-1} \nabla f(\bar{x}^k), \quad (4)$$

$$\bar{x}^{k+1} = \bar{x}^k - [\nabla^2 f(\bar{x}^k)]^{-1} \nabla f(\bar{x}^k). \quad (5)$$

В данном случае и величина шага и направление поиска полностью определены.

Если $f(\bar{x})$ – квадратичная функция (выпуклая вниз), то для достижения минимума достаточно одного шага (рис. 4.1).

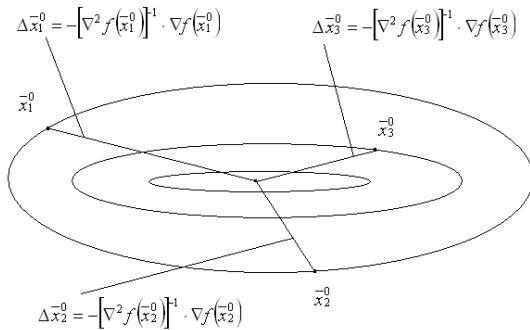


Рис. 4.1. Траектории спуска метода Ньютона для квадратичной функции

Но в общем случае нелинейной функции $f(\bar{x})$ за один шаг минимум не достигается. Поэтому итерационную формулу (5) обычно приводят к виду

$$\bar{x}^{k+1} = \bar{x}^k - \lambda^k \frac{\left[\nabla^2 f(\bar{x}^k) \right]^{-1} \nabla f(\bar{x}^k)}{\left\| \left[\nabla^2 f(\bar{x}^k) \right]^{-1} \nabla f(\bar{x}^k) \right\|}, \quad (6)$$

где λ^k – параметр длины шага, или к виду

$$\bar{x}^{k+1} = \bar{x}^k - \lambda^k \left[\nabla^2 f(\bar{x}^k) \right]^{-1} \nabla f(\bar{x}^k) = \bar{x}^k - \lambda^k H^{-1}(\bar{x}^k) \nabla f(\bar{x}^k). \quad (7)$$

Направление спуска определяется вектором (рис. 4.2)

$$\bar{S}^k = -H^{-1}(\bar{x}^k) \nabla f(\bar{x}^k).$$

Итерационный процесс (6) или (7) продолжается до тех пор, пока не будет выполнен некоторый критерий останова.

Условие, гарантирующее сходимость метода Ньютона в предположении, что функция $f(\bar{x})$ дважды дифференцируема, заключается в том, что матрица $H^{-1}(\bar{x}^k)$ должна быть положительно определенной.

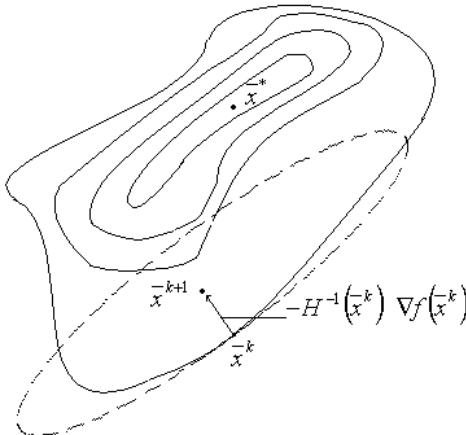


Рис. 4.2. Иллюстрация работы метода Ньютона
в случае произвольной функции

Иногда определенную сложность вызывает вычисление на каждом шаге матрицы $H^{-1}(\bar{x}^k)$. Тогда вместо метода Ньютона используют его модификацию, заключающуюся в следующем. Пусть начальное приближение является достаточно хорошим. Вычисляется матрица $[\nabla^2 f(\bar{x}^0)]^{-1}$ и в дальнейшем на всех итерациях метода вместо матрицы $[\nabla^2 f(\bar{x}^k)]^{-1}$ используется $[\nabla^2 f(\bar{x}^0)]^{-1}$.

Очередные приближения определяются соотношением

$$\bar{x}^{k+1} = \bar{x}^k - \lambda^k [\nabla^2 f(\bar{x}^0)]^{-1} \nabla f(\bar{x}^k) = \bar{x}^k - \lambda^k H^{-1}(\bar{x}^0) \nabla f(\bar{x}^k). \quad (8)$$

Естественно, что число итераций, необходимое для достижения минимума, обычно возрастает, но в целом процесс может оказаться экономичнее.

Градиентные методы, в частности метод наискорейшего спуска, обладают линейной скоростью сходимости. Метод Ньютона обладает квадратичной скоростью сходимости.

Применение метода Ньютона очень эффективно при условии, что выполняются необходимые и достаточные условия его сходимости. Однако само исследование необходимых и достаточных условий сходимости метода в случае конкретной $f(\bar{x})$ может быть достаточно сложной задачей.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для каких функций эффективно применение методов второго порядка?
2. В каких случаях модифицированный метод Ньютона сходится, а в каких – нет?
3. Что необходимо предусмотреть в реализуемом алгоритме для обеспечения сходимости метода?

5. МЕТОДЫ ПЕРЕМЕННОЙ МЕТРИКИ

5.1. ВВЕДЕНИЕ

Методы переменной метрики называют также квазиньютоновскими или градиентными с большим шагом [2].

В этих методах в процессе поиска осуществляется аппроксимация матрицы вторых частных производных или обратной к ней. Причем для этого используются только первые производные.

Очередное приближение \bar{x}^{k+1} в этих методах находится по формуле

$$\bar{x}^{k+1} = \bar{x}^k + \lambda^k \bar{S}^k = \bar{x}^k - \lambda^k \eta \bar{x}^k \nabla f(\bar{x}^k), \quad (1)$$

где матрица $\eta \bar{x}^k$, которую иногда называют матрицей направлений, представляет собой аппроксимацию для матрицы

$$H^{-1} \bar{x}^k = [\nabla^2 f(\bar{x}^k)]^{-1}.$$

Для квадратичной целевой функции (или квадратичной аппроксимации целевой функции) имеем:

$$f(\bar{x}) \approx f(\bar{x}^k) + \nabla^T f(\bar{x}^k)(\bar{x} - \bar{x}^k) + \frac{1}{2} (\bar{x} - \bar{x}^k)^T \nabla^2 f(\bar{x}^k)(\bar{x} - \bar{x}^k),$$

где $\nabla^2 f(\bar{x}^k) = H(\bar{x}^k)$.

Если вместо \bar{x} подставить в это соотношение \bar{x}^{k+1} и продифференцировать, то получим

$$\nabla f(\bar{x}^{k+1}) = \nabla f(\bar{x}^k) + H(\bar{x}^k)(\bar{x}^{k+1} - \bar{x}^k),$$

$$\nabla f(\bar{x}^{k+1}) - \nabla f(\bar{x}^k) = H(\bar{x}^k)(\bar{x}^{k+1} - \bar{x}^k).$$

Умножив на $H^{-1}(\bar{x}^k)$, получим

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

При этом если $f(\bar{x})$ – квадратичная функция, то $H(\bar{x}^k) = H$ – постоянная матрица.

Уравнение (2) можно рассматривать как систему n линейных уравнений с n неизвестными параметрами, которые необходимо оценить для того, чтобы аппроксимировать $H^{-1}(\bar{x})$ или $H(\bar{x})$ при заданных значениях $f(\bar{x})$, $\nabla f(\bar{x})$ и $\Delta \bar{x}$ на более ранних этапах поиска.

Для решения этих линейных уравнений могут быть использованы различные методы, каждый из которых приводит к различным методам переменной метрики.

В большой группе методов матрица $H^{-1}(\bar{x}^{k+1})$ аппроксимируется с помощью информации, полученной на предыдущем k -м шаге:

$$H^{-1}(\bar{x}^{k+1}) \approx \omega \eta^{k+1} = \omega \eta^k + \Delta \eta^k, \quad (3)$$

где η^k – матрица, аппроксимирующая $H^{-1}(\bar{x}^k)$ на предыдущем шаге.

Вообще $\eta^k = \eta^k \bar{x}^k$. В (3) $\Delta\eta^k$ представляет собой определяемую матрицу, а ω – масштабный (постоянный) множитель, в большинстве случаев равный единице.

Выбор $\Delta\eta^k$, по существу, и определяет соответствующий метод переменной метрики.

Для обеспечения сходимости метода матрица $\omega\eta^{k+1}$ должна быть положительно определенной и удовлетворять уравнению (2) в том случае, когда она заменяет H^{-1} .

На $k+1$ -м шаге мы знаем \bar{x}^k , $\nabla f(\bar{x}^k)$, $\nabla f(\bar{x}^{k+1})$ и η^k . И нам требуется вычислить η^{k+1} так, чтобы удовлетворялось соотношение (2).

Из выражения (2) с учетом (3)

$$\Delta\bar{x}^k = \omega\eta^{k+1} [\nabla f(\bar{x}^{k+1}) - \nabla f(\bar{x}^k)] = \omega\eta^{k+1}\Delta\bar{g}^k$$

и

$$\eta^{k+1}\Delta\bar{g}^k = \frac{1}{\omega}\Delta\bar{x}^k. \quad (4)$$

Так как $\eta^{k+1} = \eta^k + \Delta\eta^k$, то на основании (4) уравнение

$$\Delta\eta^k\Delta\bar{g}^k = \frac{1}{\omega}\Delta\bar{x}^k - \eta^k\Delta\bar{g}^k \quad (5)$$

следует разрешить относительно $\Delta\eta^k$.

Прямой подстановкой результата можно показать, что уравнение (5) имеет следующее решение:

$$\Delta\eta^k = \frac{1}{\omega} \frac{\Delta\bar{x}^k \bar{y}^T}{\bar{y}^T \Delta\bar{g}^k} - \frac{\eta^k \Delta\bar{g}^k \bar{z}^T}{\bar{z}^T \Delta\bar{g}^k}, \quad (6)$$

где \bar{z} и \bar{y} – произвольные векторы размерности n .

Например, если для $\omega=1$ выбирается специальная линейная комбинация двух направлений $\Delta\bar{x}^k$ и $\eta^k\Delta\bar{g}^k$

$$\bar{y} = \bar{z} = \Delta \bar{x}^k - \eta^k \Delta \bar{g}^k,$$

то получают метод **Бройдена** [2].

Если же берется

$$\bar{y} = \Delta \bar{x}^k, \quad \bar{z} = \eta^k \Delta \bar{g}^k,$$

то матрица η^{k+1} строится в соответствии с алгоритмом Дэвидона–Флетчера–Пауэлла [2].

Так как \bar{z} и \bar{y} – произвольные векторы, то оказываются допустимыми и другие возможности.

Если в этих алгоритмах шаги $\Delta \bar{x}^k$ строятся последовательно в результате минимизации функции $f(\bar{x})$ в направлении \bar{S}^k , то все методы, с помощью которых вычисляют симметрическую матрицу η^{k+1} , удовлетворяющую соотношению (4), дают направления, являющиеся взаимно сопряженными (в случае квадратичной целевой функции).

5.2. МЕТОД БРОЙДЕНА

Бройден показал, что если $\Delta \eta^k$ оказывается симметрической матрицей с рангом, равным единице, и должно удовлетворяться соотношение

$$\eta^{k+1} \Delta \bar{g}^k = \Delta \bar{x}^k,$$

то единственным возможным выбором $\Delta \eta^k$ является соотношение

$$\Delta \eta^k = \frac{[\Delta \bar{x}^k - \eta^k \Delta \bar{g}^k] [\Delta \bar{x}^k - \eta^k \Delta \bar{g}^k]^T}{[\Delta \bar{x}^k - \eta^k \Delta \bar{g}^k]^T \Delta \bar{g}^k}, \quad (7)$$

где

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

Последовательность шагов алгоритма

1. Задаются начальное приближение \bar{x}^0 и некоторая положитель-но определенная матрица η^0 (например, единичная $\eta^0 = E$).

2. Вычисляется

$$\bar{x}^{k+1} = \bar{x}^k - \lambda^k \eta \bar{x}^k \nabla f \bar{x}^k ,$$

так, что

$$\lambda^k = \arg \min_{\lambda} f \bar{x}^k - \lambda \eta \bar{x}^k \nabla f \bar{x}^k .$$

3. Находится очередное приближение матрицы

$$\eta^{k+1} = \eta^k + \Delta \eta^k ,$$

где $\Delta \eta^k$ находится по формуле (7).

4. Проверяется критерий останова, например, $\|\nabla f \bar{x}^{k+1}\| \leq \varepsilon$. Если он не выполняется, то осуществляется переход на шаг 2.

Если целевая функция является квадратичной, то направления поиска $\bar{s}^k = -\eta \bar{x}^k \nabla f \bar{x}^k$ на последующих итерациях оказываются сопряженными и для определения минимума достаточно сделать n шагов (рис. 5.1).

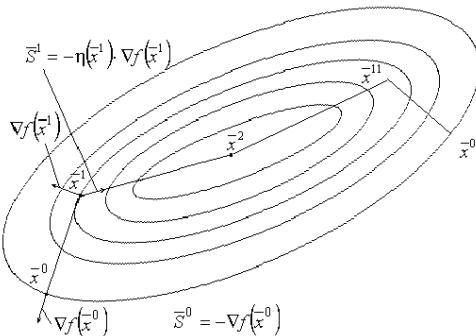


Рис. 5.1. Траектории спуска метода переменной метрики для квадратичной функции

В случае минимизации неквадратичной функции возможны нежелательные явления, например:

1) матрица η^{k+1} может перестать быть положительно определенной. В этом случае необходимо каким-либо способом обеспечить ее положительную определенность;

2) вычисляемая величина $\Delta\eta^k$ вследствие ошибок округления может стать неограниченной;

3) если $\Delta\bar{x}^k = -\lambda^k \eta^{-1} \bar{x}^k \quad \nabla f(\bar{x}^k)$ на текущем шаге случайно совпадает с направлением поиска на предыдущем шаге, то матрица η^{k+1} становится сингулярной или неопределенной.

Чтобы избежать этих явлений, стараются обновлять алгоритм после n шагов, считая $(n+1)$ -ю итерацию начальной.

5.3. МЕТОД ДЭВИДОНА–ФЛЕТЧЕРА–ПАУЭЛЛА

Реальная разница алгоритмов метода переменной метрики заключается в нахождении $\Delta\eta$ [2]. В данном алгоритме матрица $\Delta\eta$ имеет ранг 2. Как и в предыдущем случае, матрица η перевычисляется таким образом, чтобы для квадратичной функции после n шагов она совпала с матрицей $H^{-1} \bar{x}^k = [\nabla^2 f(\bar{x}^k)]^{-1}$.

Исходная матрица обычно выбирается единичной: $\eta^0 = E$. Хотя, возможно, предпочтительней задание начального приближения элементов этой матрицы конечно-разностными приближениями вторых частных производных.

Соотношение для $\Delta\eta^k$ в алгоритме Дэвидона–Флетчера–Пауэлла можно получить путем подстановки

$$\bar{y} = \Delta\bar{x}^k, \quad \bar{z} = \eta^k \Delta\bar{g}^k$$

$$\text{в уравнение (6)} \left(\Delta\eta^k = \frac{1}{\omega} \frac{\Delta\bar{x}^k \bar{y}^T}{\bar{y}^T \Delta\bar{g}^k} - \frac{\eta^k \Delta\bar{g}^k \bar{z}^T}{\bar{z}^T \Delta\bar{g}^k} \right).$$

Тогда получим:

$$\eta^{k+1} = \eta^k + A^k - B^k = \eta^k + \frac{\Delta\bar{x}^k \Delta\bar{x}^k^T}{\Delta\bar{x}^k^T \Delta\bar{g}^k} - \frac{\eta^k \Delta\bar{g}^k \Delta\bar{g}^k^T \eta^k^T}{\Delta\bar{g}^k^T \eta^k \Delta\bar{g}^k}. \quad (8)$$

Следует отметить, что A^k и B^k являются симметрическими матрицами, так что η^k – симметрическая и η^{k+1} будет симметрической.

Этот алгоритм является одним из наиболее эффективных алгоритмов переменной метрики. Алгоритм, использующий соотношение (8), оказывается достаточно эффективным при выполнении следующих условий:

- 1) ошибки округления при вычислении $\nabla f(\bar{x}^k)$ невелики;
- 2) матрица η^k в процессе вычислений не становится «плохой».

В ходе оптимизации этим методом происходит постепенный переход от градиентного направления спуска к ньютоновскому. При этом используются преимущества каждого метода на соответствующем этапе.

Роль матриц A^k в (8) заключается в обеспечении того, чтобы $\eta \rightarrow H^{-1}$, тогда как матрица B^k обеспечивает положительную определенность матрицы η^{k+1} на всех этапах и в пределе их сумма исключает начальную матрицу η^0 . Используем (8) на нескольких этапах, начиная с η^0 :

$$\eta^1 = E + A^0 - B^0,$$

$$\eta^2 = \eta^1 + A^1 - B^1 = E + (A^0 + A^1) - (B^0 + B^1),$$

...

$$\eta^{k+1} = E + \sum_{i=0}^k A^i - \sum_{i=0}^k B^i.$$

В случае квадратичной целевой функции при $k = n - 1$ должно выполняться равенство $H^{-1} = \sum_{i=0}^{n-1} A^i$, а сумма матриц $\sum_{i=0}^k B^i$ строится таким образом, чтобы она сократилась с начальным значением η^0 .

При квадратичной функции используемые направления поиска являются сопряженными. Именно это определяет эффективность алгоритма.

5.4. АЛГОРИТМЫ ПИРСОНА

Если в выражении (6) $\left(\Delta\eta^k = \frac{1}{\omega} \frac{\Delta\bar{x}^k \bar{y}^T}{\bar{y}^T \Delta\bar{g}^k} - \frac{\eta^k \Delta\bar{g}^k \bar{z}^T}{\bar{z}^T \Delta\bar{g}^k} \right)$ положить

$\bar{y} = \bar{z} = \Delta\bar{x}^k$ и $\omega = 1$, то очередное приближение матрицы направлений определится выражением [2]:

$$\eta^{k+1} = \eta^k + \frac{[\Delta\bar{x}^k - \eta^k \Delta\bar{g}^k] \Delta\bar{x}^k^T}{\Delta\bar{x}^k^T \Delta\bar{g}^k}, \quad (9)$$

где $\eta^0 = R^0$ – произвольная положительно определенная матрица.

Соответствующий алгоритм переменной метрики получил название **второго алгоритма Пирсона**. Метод обычно приводит к плохим направлениям поиска. Однако были примеры очень эффективного применения метода в приложениях.

Третий алгоритм Пирсона получается при подстановке в уравнение (6) следующих параметров: $\bar{y} = \bar{z} = \eta^k \Delta\bar{g}^k$ и $\omega = 1$. В этом случае итерационная формула принимает вид [2]:

$$\eta^{k+1} = \eta^k + \frac{[\Delta\bar{x}^k - \eta^k \Delta\bar{g}^k] [\eta^k \Delta\bar{g}^k]^T}{\Delta\bar{g}^k^T \eta^k \Delta\bar{g}^k}, \quad (10)$$

$$\eta^0 = R^0.$$

Третий алгоритм на тестовых функциях оказывается более эффективным.

5.5. ПРОЕКТИВНЫЙ АЛГОРИТМ НЮТОНА–РАФСОНА

Этот алгоритм также предложен Пирсоном [2]. Получается из уравнения (6) при $\omega \rightarrow \infty$ и $\bar{z} = \eta^k \Delta \bar{g}^k$:

$$\eta^{k+1} = \eta^k - \frac{[\eta^k \Delta \bar{g}^k] [\eta^k \Delta \bar{g}^k]^T}{\Delta \bar{g}^k \Delta \bar{g}^k}, \quad (11)$$

$$\eta^0 = R^0.$$

5.6. МЕТОДЫ ГРИНШТАДТА И ГОЛЬДФАРБА

Очередное приближение матрицы направлений определяется соотношением

$$\eta^{k+1} = \eta^k + \Delta \eta^k,$$

где в алгоритме Гринштадта [2] –

$$\Delta \eta^k = \frac{1}{\Delta \bar{g}^k \Delta \bar{g}^k} \left\{ \Delta \bar{x}^k \Delta \bar{g}^k \eta^k + \eta^k \Delta \bar{g}^k \Delta \bar{x}^k - \right.$$

$$\left. - \frac{\left[\Delta \bar{g}^k \Delta \bar{x}^k - \Delta \bar{g}^k \eta^k \Delta \bar{g}^k \right] \eta^k \Delta \bar{g}^k \Delta \bar{g}^k \eta^k}{\Delta \bar{g}^k \Delta \bar{g}^k} \right\}, \quad (12)$$

а в алгоритме Гольдфарба [2] –

$$\Delta \eta^k = \frac{1}{\Delta \bar{g}^k \Delta \bar{g}^k} \left\{ \Delta \bar{x}^k \Delta \bar{g}^k \eta^k + \eta^k \Delta \bar{g}^k \Delta \bar{x}^k - \right.$$

$$-\left[1 + \frac{\Delta \bar{g}^k \top \Delta \bar{x}^k}{\Delta \bar{g}^k \top \eta^k \Delta \bar{g}^k} \right] \eta^k \Delta \bar{g}^k \quad \Delta \bar{g}^k \top \eta^k \Biggr\}. \quad (13)$$

По эффективности данные методы сравнимы с алгоритмом Дэвидона–Флетчера–Пауэлла.

5.7. АЛГОРИТМ ФЛЕТЧЕРА

Флетчером был предложен алгоритм, в котором условие окончания процесса для квадратичной функции после n шагов было отброшено, но сохранено свойство, заключающееся в том, что для квадратичных функций $\eta \rightarrow H^{-1} \bar{x}$ в том смысле, что собственные значения η стремятся к собственным значениям H^{-1} .

Полученное Флетчером соотношение для очередного приближения матрицы η имеет вид [2]

$$\eta^{k+1} = \left[E - \frac{\Delta \bar{x}^k \Delta \bar{g}^k \top}{\Delta \bar{x}^k \top \Delta \bar{g}^k} \right] \eta^k \left[E - \frac{\Delta \bar{g}^k \Delta \bar{x}^k \top}{\Delta \bar{x}^k \top \Delta \bar{g}^k} \right] + \frac{\Delta \bar{x}^k \Delta \bar{x}^k \top}{\Delta \bar{x}^k \top \Delta \bar{g}^k}. \quad (14)$$

Однако в реализованном Флетчером алгоритме матрица η^k в зависимости от выполнения условий вычисляется по-разному.

Если

$$\Delta \bar{g}^k \top H^{-1}(\bar{x}^k) \Delta \bar{g}^k < \Delta \bar{g}^k \top \eta^k \Delta \bar{g}^k, \quad (a)$$

то для вычисления очередного приближения η^{k+1} используется соотношение (8) (метод Дэвидона–Флетчера–Пауэлла). А если

$$\Delta \bar{g}^k \top H^{-1}(\bar{x}^k) \Delta \bar{g}^k \geq \Delta \bar{g}^k \top \eta^k \Delta \bar{g}^k, \quad (b)$$

то используется соотношение (14).

Очевидно, что проверка условий (а)–(б) затруднительна и теряет смысл, так как приходится находить $H^{-1} \bar{x}^k$. Поэтому при реализации алгоритма обычно используют формулу (14) без проверки условий (а)–(б). Сравнение алгоритмов на тестовых функциях показывает, что в этом случае алгоритм Флетчера проигрывает по эффективности алгоритму Дэвидона–Флетчера–Пауэлла.

Эффективность алгоритмов Дэвидона–Флетчера–Пауэлла и Флетчера в существенной мере определяют используемые методы одномерного поиска. Зачастую успех конкретной реализации определяет именно эффективность используемых методов одномерного поиска. В авторских реализациях этих методов при решении задачи

$$\lambda^k = \arg \min f(\bar{x}^k + \lambda \bar{S}^k),$$

где

$$\bar{S}^k = -\eta \bar{x}^k \nabla f(\bar{x}^k),$$

оптимальное значение λ^k определяется на интервале $(0, \lambda')$ с помощью кубической интерполяции. Величина

$$\lambda' = \min \left\{ 1, \frac{2[f(\bar{x}^k) - f_0]}{\nabla^T f(\bar{x}^k) \bar{S}^k} \right\},$$

где f_0 – нижняя оценка значений $f(\bar{x})$, найденная в процессе одномерного поиска по направлению \bar{S}^k (в процессе выделения интервала, содержащего минимум). Если полученное в результате кубической интерполяции значение λ оказывается больше λ' , то для пробных шагов начальное значение λ берется в виде

$$\lambda^{r+1} = 0,1\lambda^r,$$

где r обозначает номер в последовательности шагов при одномерном поиске. А если $\lambda < \lambda'$ и $f(\bar{x}^k + \lambda \bar{S}^k) < f_0$, то одномерный поиск заканчивается.

Вообще говоря, применение квадратичной интерполяции при одномерном поиске оказывается нисколько не хуже кубической. И алгоритмы оказываются столь же эффективными.

5.8. АЛГОРИТМЫ С АППРОКСИМАЦИЕЙ МАТРИЦЫ ГЕССЕ

Можно строить аналогичные алгоритмы, аппроксимируя в процессе поиска не матрицу $H^{-1} \bar{x}^k$, а матрицу $H \bar{x}^k$. А затем строить обратную к ней.

В этом случае на каждой итерации метода находится очередное приближение $H(\bar{x}^{k+1}) \approx \Gamma^{k+1} = \Gamma^k + \Delta\Gamma^k$, где Γ^k – оценка $H(\bar{x}^k)$, а матрица $\Delta\Gamma^k$ – симметрическая матрица ранга 1, такая, что Γ^{k+1} удовлетворяет уравнению $\Gamma^{k+1} \Delta\bar{x}^k = \Delta\bar{g}^k$. При этом

$$\Delta\Gamma^k = \frac{[\Delta\bar{g}^k - \Gamma^k \Delta\bar{x}^k] [\Delta\bar{g}^k - \Gamma^k \Delta\bar{x}^k]^T}{[\Delta\bar{g}^k - \Gamma^k \Delta\bar{x}^k]^T \Delta\bar{x}^k}. \quad (15)$$

Поскольку в процессе поиска матрица Γ^{k+1} может не быть положительно определенной, следует использовать ограничительные условия, обеспечивающие положительную определенность такой матрицы. В качестве начального приближения можно выбирать $\Gamma^0 = [\eta^0]^{-1}$.

К алгоритмам такого типа относится алгоритм **Гольштайн** и **Прайса**, который описывается определенной последовательностью действий [2] и предназначен для минимизации выпуклых функций.

Гольштайн и Прайс не использовали (15), а аппроксимировали матрицу $H(\bar{x}^k)$ при помощи разностной схемы, основанной на полифакториальном построении (полифакториал – произведение n первых четных чисел), а затем проводили обращение матрицы. При этом для оценки $H \bar{x}^k$ требуется лишь информация о $f \bar{x}^k$ и $\nabla f \bar{x}^k$.

На k -м этапе алгоритм выглядит следующим образом. Заранее заданы величины $0 < \delta < 1/2$ и $r > 0$.

1. Вычисляется в качестве аппроксимации $H \bar{x}^k$ матрица $\tilde{H} \bar{x}^k$, j -й столбец которой определяется по формуле

$$\nabla f \bar{x}^k + \theta^k E_j - \nabla f \bar{x}^k ,$$

где $\theta^k = r \|\varphi \bar{x}^{k-1}\|$ для $k > 0$, $\theta^0 = r$, E_j – j -й столбец единичной матрицы E размерности $n \times n$. $\varphi \bar{x}^k$ – вектор-столбец, определяемый в соответствии с условиями:

- $\varphi \bar{x}^k = -\nabla f \bar{x}^k$, если $k = 0$ или $\tilde{H} \bar{x}^k$ сингулярна, или $\nabla^T f \bar{x}^k \tilde{H}^{-1} \bar{x}^k \nabla f \bar{x}^k \leq 0$, так что матрица $\tilde{H}^{-1} \bar{x}^k$ не является положительно определенной;
- $\varphi \bar{x}^k = -\tilde{H}^{-1} \bar{x}^k \nabla f \bar{x}^k$ – в противном случае.

Заметим, что матрица $\tilde{H} \bar{x}^k$ не обязательно симметрическая матрица, и если $\nabla^T f \bar{x}^k \tilde{H}^{-1} \bar{x}^k \nabla f \bar{x}^k \leq 0$, то предлагаемое направление поиска $\varphi \bar{x}^k$ и направление градиента $\nabla f \bar{x}^k$ отличаются более чем на 90° . Следовательно, $\varphi \bar{x}^k$ может быть направлено в сторону, в которой $f \bar{x}$ увеличивается.

2. Для выражения

$$F \bar{x}^k, \lambda = \frac{f \bar{x}^k - f \bar{x}^k + \lambda \varphi \bar{x}^k}{\lambda [\nabla^T f \bar{x}^k \varphi \bar{x}^k]}$$

вычисляется λ^k так, чтобы $\delta \leq F(\bar{x}^k, 1)$ или $\delta \leq F(\bar{x}^k, \lambda^k) \leq 1 - \delta$, $\lambda^k \neq 1$. Эти условия нужны для того, чтобы не допускать шагов поиска, которые далеко выходят за область линейного изменения целевой функции в окрестности \bar{x}^k , предполагавшуюся при аппроксимации $H \bar{x}$.

3. Берется $\bar{x}^{k+1} = \bar{x}^k + \lambda^k \varphi(\bar{x}^k)$.
4. Процесс заканчивается, когда $\|\varphi(\bar{x}^k)\| < \varepsilon$.

Параметр r следует выбирать так, чтобы матрица $\tilde{H}(\bar{x}^k)$ аппроксимировала $H(\bar{x}^k)$ как можно лучше. Величина δ выбирается так, чтобы значения $f(\bar{x}^k)$, $k = 1, 2, \dots$, представляли собой монотонно убывающую последовательность; чем ближе значение δ к $1/2$, тем в большей степени $f(\bar{x}^k) + \lambda \varphi(\bar{x}^k)$ приближается к своему минимуму по λ .

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем заключается общая идея методов переменной метрики?
2. Итерационные соотношения метода Брайдена и шаги алгоритма.
3. Формирование очередного приближения матрицы η^{k+1} в алгоритме Дэвидона–Флетчера–Пауэлла.
4. Формирование очередного приближения матрицы η^{k+1} в алгоритмах Пирсона.
5. Формирование очередного приближения матрицы η^{k+1} в алгоритмах Гринштадта и Гольдфарба.
6. Алгоритм Флетчера.
7. Алгоритмы Гольштайна и Прайса. Их отличие от предыдущих.

6. МЕТОДЫ ШТРАФНЫХ ФУНКЦИЙ

Естественно, что наиболее распространенные задачи на практике – это оптимизационные задачи **при наличии ограничений**, т. е. задачи поиска оптимального решения, удовлетворяющего некоторой системе ограничений. Существование эффективных алгоритмов решения безусловных задач оптимизации всегда толкает на попытку использовать эти методы для решения условных задач после соответствующего преобразования условной задачи к некоторой эквивалентной безусловной задаче [6].

Пусть необходимо решить задачу

$$\min f(\bar{x}) \left| \begin{array}{l} h_j(\bar{x}) = 0, j = \overline{1, m}; \\ g_j(\bar{x}) \leq 0, j = \overline{m+1, k} \end{array} \right. , \quad (1)$$

в которой целевая функция и функции системы ограничений представляют собой выпуклые функции (желательно).

Основная идея метода штрафных функций заключается в следующем [6]. Строят такую вспомогательную функцию

$$Q(\bar{x}, \bar{r}) = f(\bar{x}) + \sum_{j=1}^m r_j H[h_j(\bar{x})] + \sum_{j=m+1}^k r_j G[g_j(\bar{x})], \quad (2)$$

чтобы приближенное решение задачи (1) находилось в результате решения *последовательности* задач безусловной минимизации функций (2)

$$\min Q(\bar{x}, \bar{r}). \quad (3)$$

В методе (внешних) **штрафных функций** функции H · и G · выбираются таким образом, чтобы они становились отличными от нуля (*положительными*) при нарушении соответствующего ограничения. А так как мы минимизируем (2), то движение в сторону нарушения ограничения становится невыгодным (рис. 6.1). В данном методе функции H · и G · внутри допустимой области должны быть равны нулю. Например, для ограничений неравенств:

$$G_j[g_j(\bar{x})] \rightarrow 0 \text{ при } g_j(\bar{x}) \rightarrow 0^+.$$

Приближенное решение задачи (1) получается в результате решения последовательности задач (3) при $r_j \rightarrow \infty$, $j = \overline{1, k}$. Соответствующие методы называют еще *методами внешней точки*.

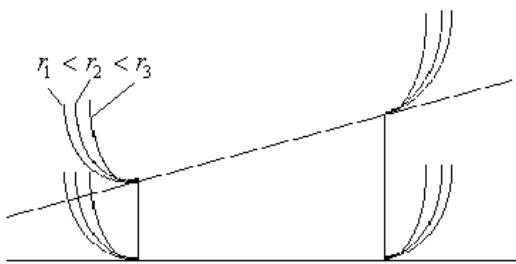


Рис. 6.1. Формирование штрафа при нарушении ограничений

В методе **барьерных функций** [6] функции $H \cdot$ и $G \cdot$ в допустимой области выбираются отличными от нуля, причем такими, чтобы при приближении к границе допустимой области (изнутри) они возрастили, препятствуя выходу при поиске за границу области (рис. 6.2). В этом случае эти функции должны принимать малые (*положительные* или *отрицательные*) значения внутри допустимой области и большие *положительные* вблизи границы (внутри области). Например, для ограничений неравенств:

$$G_j \left[g_j(\bar{x}) \right] \rightarrow \infty \text{ при } g_j(\bar{x}) \rightarrow 0^-.$$

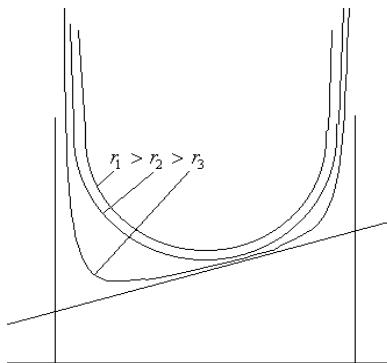


Рис. 6.2. Формирование барьёров вблизи границы

Такие методы называют еще *методами внутренней точки*. В алгоритмах, использующих функции штрафа данного типа (барьерные функции), необходимо, чтобы в процессе поиска точка \bar{x} всегда оставалась внутренней точкой допустимой области. Приближенное решение задачи (1) получается в результате решения последовательности задач вида (3) при $r_j \rightarrow 0$, $j = 1, k$.

При выборе функций штрафов для ограничений равенств обычно требуется, чтобы

$$H_j[h_j(\bar{x})] \rightarrow 0 \text{ при } h_j(\bar{x}) \rightarrow 0.$$

Это могут быть, например, функции следующего вида:

$$1) H_j[h_j(\bar{x})] = |h_j(\bar{x})|,$$

$$2) H_j[h_j(\bar{x})] = |h_j(\bar{x})|^2,$$

$$3) H_j[h_j(\bar{x})] = |h_j(\bar{x})|^\alpha \text{ при четном } \alpha.$$

Для ограничений неравенств функции штрафа подбирают таким образом, чтобы

$$G_j[g_j(\bar{x})] = 0 \text{ при } g_j(\bar{x}) \leq 0;$$

$$G_j[g_j(\bar{x})] > 0 \text{ при } g_j(\bar{x}) > 0.$$

Этому требованию отвечают, например, функции вида:

$$1) G_j[g_j(\bar{x})] = \frac{1}{2} g_j(\bar{x}) + |g_j(\bar{x})|,$$

$$2) G_j[g_j(\bar{x})] = \left[\frac{1}{2} g_j(\bar{x}) + |g_j(\bar{x})| \right]^2,$$

$$3) G_j[g_j(\bar{x})] = \left[\frac{1}{2} g_j(\bar{x}) + |g_j(\bar{x})| \right]^\alpha \text{ при четной степени } \alpha.$$

Барьерными функциями для ограничений неравенств могут служить, например, функции вида:

$$1) G_j[g_j(\bar{x})] = -\frac{1}{g_j(\bar{x})},$$

$$2) G_j[g_j(\bar{x})] = -\ln[-g_j(\bar{x})].$$

Последовательность действий при реализации методов штрафных или барьерных функций выглядит следующим образом.

1. На основании задачи (1) строим функцию (2). Выбираем начальное приближение \bar{x} и начальные значения коэффициентов штрафа r_j .

2. Решаем задачу (3).

3. Если полученное решение не удовлетворяет системе ограничений, то в случае использования метода штрафных функций увеличиваем значения коэффициентов штрафа r_j и снова решаем задачу (3). В случае метода барьерных функций, чтобы можно было получить решение на границе, значения коэффициентов r_j уменьшаются.

4. Процесс прекращается, если найденное решение удовлетворяет системе ограничений с определенной точностью.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем заключается идея метода штрафных функций? Какая последовательность задач решается в этом случае? Как меняются r_j в процессе поиска?

2. В чем отличие метода барьерных функций? Какая последовательность задач решается в этом случае? Как меняются r_j в данном случае?

3. Виды штрафных функций .

4. Виды барьерных функций.

7. СТАТИСТИЧЕСКИЕ МЕТОДЫ ПОИСКА

7.1. ВВЕДЕНИЕ

Статистические методы, или **методы случайного поиска**, получили достаточно широкое распространение при построении оптимальных решений в различных приложениях. Это объясняется в первую очередь тем, что с ростом размерности задач резко снижается эффективность регулярных методов поиска (детерминированных): так называемое «проклятие размерности». Во-вторых, зачастую информация об оптимизируемом объекте слишком мала для того, чтобы можно было применить детерминированные методы. Достаточно часто статистические алгоритмы используют при поиске оптимального решения в сис-

темах управления (рис. 7.1), когда отклик системы можно получить только при задании управляющих воздействий \bar{x} на ее входах. В таких ситуациях статистические алгоритмы могут оказаться значительно эффективнее детерминированных.

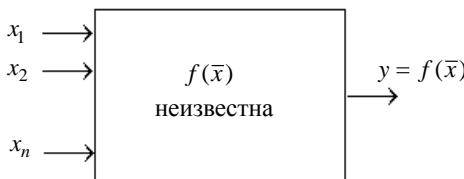


Рис. 7.1. Оптимальное управление системой

Наибольший эффект применение статистических методов приносит при решении задач большой размерности или при поиске глобального экстремума.

Под случайными, или статистическими, методами поиска будем понимать методы, использующие элемент случайности либо при сборе информации о целевой функции при пробных шагах, либо для улучшения значений функции при рабочем шаге. Случайным образом можно выбирать направление спуска, длину шага, величину штрафа при нарушении ограничения и т.д.

Статистические алгоритмы обладают рядом достоинств:

- простотой реализации и отладки программ;
- надежностью и помехоустойчивостью;
- универсальностью;
- возможностью введения операций обучения в алгоритм поиска;
- возможностью введения операций прогнозирования оптимальной точки (оптимального решения).

Основными недостатками являются большое количество вычислений минимизируемой функции и медленная сходимость в районе экстремума.

Принято считать, что преимущество статистических методов проявляется с ростом размерности задач, так как вычислительные затраты в детерминированных методах поиска в этом случае растут быстрее, чем в статистических алгоритмах.

7.2. ПРОСТОЙ СЛУЧАЙНЫЙ ПОИСК

Пусть нам необходимо решить задачу минимизации функции $f(\bar{x})$ при условии, что $\bar{x} \in [\bar{A}, \bar{B}]$.

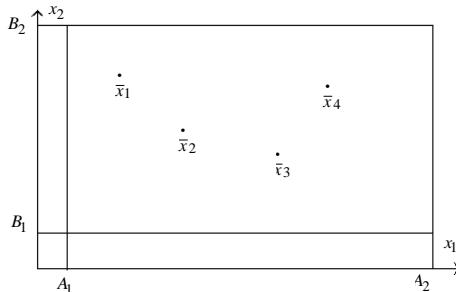


Рис. 7.2. Простой случайный поиск

В данной области по равномерному закону выбираем случайную точку \bar{x}_1 (рис. 7.2) и вычисляем в ней значение функции $y_1 = f(\bar{x}_1)$. Затем выбираем таким же образом случайную точку \bar{x}_2 и вычисляем $y_2 = f(\bar{x}_2)$. Запоминаем минимальное из этих значений и точку, в которой значение функции минимально. Далее генерируем новую точку. Проводим N экспериментов, после чего лучшую точку берем в качестве решения задачи (точку, в которой функция имеет минимальное значение среди всех «случайно» сгенерированных).

Оценим число экспериментов, необходимое для определения решения (точки минимума) с заданной точностью. Пусть n – размерность вектора переменных. Объем n -мерного прямоугольника, в котором ведется поиск минимума,

$$V = \prod_{i=1}^n B_i - A_i .$$

Если необходимо найти решение с точностью ε_i , $i = \overline{1, n}$, по каждой из переменных, то мы должны попасть в окрестность оптимальной точки с объемом

$$V_\varepsilon = \prod_{i=1}^n \varepsilon_i .$$

Вероятность попадания в эту окрестность при одном испытании равна $P_\varepsilon = \frac{V_\varepsilon}{V}$. Вероятность непопадания равна $1 - P_\varepsilon$. Испытания независимы, поэтому вероятность непопадания за N экспериментов равна $(1 - P_\varepsilon)^N$.

Вероятность того, что мы найдем решение за N испытаний:

$$P = 1 - (1 - P_\varepsilon)^N.$$

Отсюда нетрудно получить оценку необходимого числа испытаний N для определения минимума с требуемой точностью:

$$N \geq \frac{\ln 1 - P}{\ln 1 - P_\varepsilon}.$$

Опираясь на заданную точность ε_i , $i = \overline{1, n}$, и величину V , можно определить P_ε и, задаваясь вероятностью P , посмотреть, как меняется требуемое количество экспериментов N в зависимости от P_ε и P (табл. 7.1).

При решении экстремальных задач на областях со сложной геометрией обычно вписывают эту область в n -мерный параллелепипед (рис. 7.3). А далее генерируют в этом n -мерном параллелепипеде случайные точки по равномерному закону, оставляя только те, которые попадают в допустимую область.

Т а б л и ц а 7.1

Требуемое количество экспериментов N

P_ε	P				
	0.8	0.9	0.95	0.99	0.999
0.1	16	22	29	44	66
0.025	64	91	119	182	273
0.01	161	230	299	459	688
0.005	322	460	598	919	1379
0.001	1609	2302	2995	4603	6905

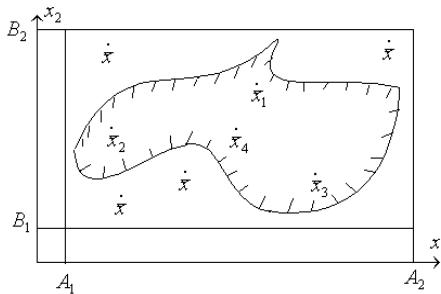


Рис. 7.3. Простой случайный поиск на сложной области

Различают направленный и ненаправленный случайный поиск.

Ненаправленный случайный поиск. При таком поиске все последующие испытания проводят совершенно независимо от результатов предыдущих. Сходимость такого поиска очень мала, но имеется важное преимущество, связанное с возможностью решения многоэкстремальных задач (искать глобальный экстремум). Примером ненаправленного поиска является рассмотренный простой случайный поиск.

Направленный случайный поиск. В этом случае отдельные испытания связаны между собой. Результаты проведенных испытаний используются для формирования последующих. Как правило, случайность используется при формировании направления спуска. Сходимость таких методов, как правило, выше, но сами методы обычно приводят только к локальным экстремумам.

7.3. ПРОСТЕЙШИЕ АЛГОРИТМЫ НАПРАВЛЕННОГО СЛУЧАЙНОГО ПОИСКА

7.3.1. АЛГОРИТМ ПАРНОЙ ПРОБЫ

В данном алгоритме четко разделены пробный и рабочий шаги.

Пусть \bar{x}^k – найденное на k -м шаге наименьшее значение минимизируемой функции $f(\bar{x})$. По равномерному закону генерируется случайный единичный вектор $\bar{\xi}$ и по обе стороны от исходной точки \bar{x}^k

делаются две пробы, т. е. проводим вычисление функции в точках $\bar{x}_{1,2}^k = \bar{x}^k \pm g\bar{\xi}$, где g – величина пробного шага.

Рабочий шаг делается в направлении наименьшего значения целевой функции (рис. 7.4). Очередное приближение определяется соотношением

$$\bar{x}^{k+1} = \bar{x}^k + \Delta\bar{x}^k = \bar{x}^k + a\bar{\xi} \operatorname{sign} f(\bar{x}^k - g\bar{\xi}) - f(\bar{x}^k + g\bar{\xi}) .$$

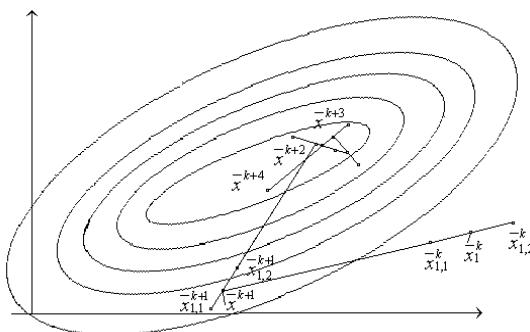


Рис. 7.4. Алгоритм парной пробы

Особенностью данного алгоритма является его повышенная тенденция к «блужданию». Даже при выходе в район экстремума алгоритм может увести процесс поиска в сторону.

7.3.2. АЛГОРИТМ НАИЛУЧШЕЙ ПРОБЫ

На k -м шаге мы имеем точку \bar{x}^k . Генерируется m случайных единичных векторов $\bar{\xi}_1, \dots, \bar{\xi}_m$. Делаются пробные шаги в направлениях $g\bar{\xi}_1, \dots, g\bar{\xi}_m$ и в точках $\bar{x}^k + g\bar{\xi}_1, \dots, \bar{x}^k + g\bar{\xi}_m$ вычисляются значения функции. Выбирается то направление, которое приводит к наибольшему уменьшению функции: $\bar{\xi}^* = \arg \min_{i=1,m} f(\bar{x}^k + g\bar{\xi}_i)$. И в данном направлении делается шаг (рис. 7.5)

$$\Delta\bar{x}^k = \lambda \bar{\xi}^* .$$

Параметр λ может находиться как результат минимизации по направлению, определяемому наилучшей пробой, или выбираться по определенному правилу.

С увеличением числа проб выбранное направление приближается к направлению $-\nabla f(\bar{x})$.

Если функция $f(\bar{x})$ близка к линейной, то имеется возможность ускорить поиск, рассматривая вместе с наилучшей и наихудшую пробу. Тогда рабочий шаг можно делать или в направлении наилучшей пробы, или в направлении, противоположном наихудшой пробе.

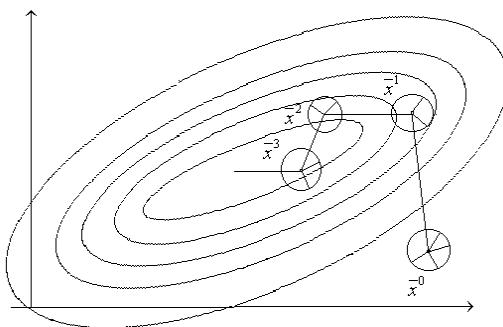


Рис. 7.5. Алгоритм наилучшей пробы

7.3.3. МЕТОД СТАТИСТИЧЕСКОГО ГРАДИЕНТА

Из исходного состояния \bar{x}^k в m случайных направлениях делается m независимых проб $g\bar{\xi}_1, \dots, g\bar{\xi}_m$, а затем вычисляются значения минимизируемой функции в соответствующих точках. Для каждой пробы запоминаем приращения функции

$$\Delta f_j = f(\bar{x}^k) + g\bar{\xi}_j - f(\bar{x}^k).$$

После этого формируем векторную сумму

$$\Delta \bar{f} = \sum_{j=1}^m \bar{\xi}_j \Delta f_j.$$

В пределе при $m \rightarrow \infty$ направление $\Delta \bar{f}$ совпадает с направлением градиента целевой функции. При конечном m вектор $\Delta \bar{f}$ представляет собой статистическую оценку направления градиента. В направлении $\Delta \bar{f}$ делается рабочий шаг (рис. 7.6). В результате очередное приближение определяется соотношением

$$\bar{x}^{k+1} = \bar{x}^k - \lambda \frac{\Delta \bar{f}}{\|\Delta \bar{f}\|}.$$

При выборе оптимального значения λ , которое минимизирует функцию в заданном направлении, мы получаем статистический вариант метода наискорейшего спуска. Существенное преимущество перед детерминированными алгоритмами заключается в возможности принятия решения о направлении рабочего шага при $m < n$. При $m = n$ и неслучайных ортогональных рабочих шагах, направленных вдоль осей координат, алгоритм вырождается в градиентный метод.

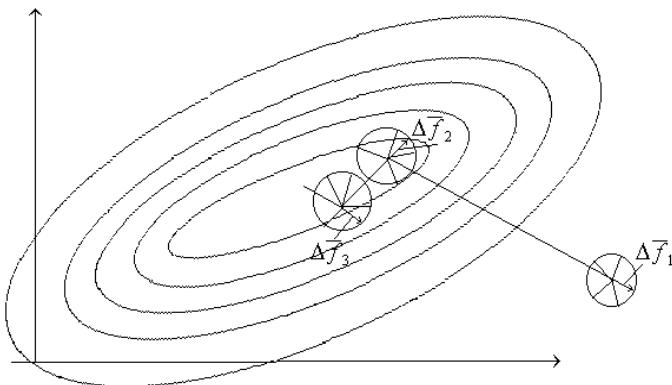


Рис. 7.6. Алгоритм статистического градиента

7.3.4. АЛГОРИТМ НАИЛУЧШЕЙ ПРОБЫ С НАПРАВЛЯЮЩИМ ГИПЕРКВАДРАТОМ

Внутри допустимой области строится гиперквадрат. В этом гиперквадрате случайным образом разбрасывается m точек $\bar{x}_1, \dots, \bar{x}_m$, в которых вычисляются значения функции. Среди построенных точек выбираем наилучшую. Таким образом, на 1-м этапе координаты случайных точек удовлетворяют неравенствам $a_i^1 \leq x_i \leq b_i^1$, $i = \overline{1, n}$, и $\bar{x}^1 = \arg \min_{j=1, m} f(\bar{x}_j)$ – точка с минимальным значением целевой функции (рис. 7.7).

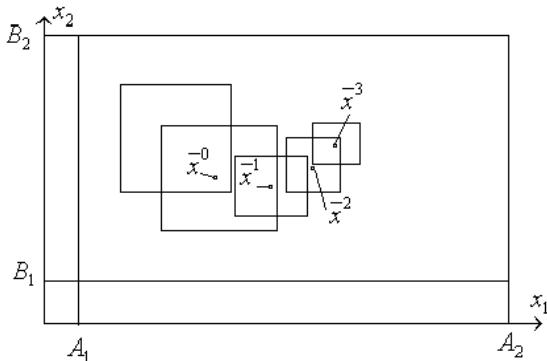


Рис. 7.7. Алгоритм наилучшей пробы с направляющим гиперквадратом

Опираясь на эту точку, строим новый гиперквадрат. Точка, в которой достигается минимум функции на k -м этапе, берется в качестве центра нового гиперквадрата на $(k+1)$ -м этапе.

Координаты вершин гиперквадрата на $(k+1)$ -м этапе определяются соотношениями

$$a_i^{k+1} = x_i^{k+1} - \frac{b_i^k - a_i^k}{2}, \quad b_i^{k+1} = x_i^{k+1} + \frac{b_i^k - a_i^k}{2},$$

где \bar{x}^k – наилучшая точка в гиперквадрате на k -м этапе.

В новом гиперквадрате выполняем ту же последовательность действий, случайным образом разбрасывая m точек. В результате осуществляется направленное перемещение гиперквадрата в сторону уменьшения функции.

В алгоритме с обучением стороны гиперквадрата могут регулироваться в соответствии с изменением по некоторому правилу параметра α , определяющему стратегию изменения стороны гиперквадрата. В этом случае координаты вершин гиперквадрата на $(k+1)$ -м этапе будут определяться соотношениями

$$a_i^{k+1} = x_i^{k+1} - \frac{b_i^k - a_i^k}{2\alpha}, \quad b_i^{k+1} = x_i^{k+1} + \frac{b_i^k - a_i^k}{2\alpha}.$$

Хорошо выбранное правило регулирования стороны гиперквадрата приводит к достаточно эффективному алгоритму поиска.

В алгоритмах случайного поиска вместо направляющего гиперквадрата могут использоваться направляющие гиперсфера, направляющие гиперконусы.

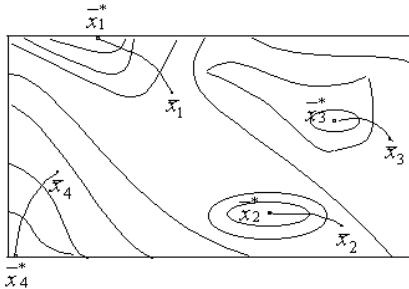
7.4. АЛГОРИТМЫ ГЛОБАЛЬНОГО ПОИСКА

Случайный поиск приобретает решающее значение при решении многоэкстремальных задач и оптимизации сложных объектов. В общем случае решение многоэкстремальных задач без элемента случайности практически невозможно.

Рассмотрим некоторые подходы к поиску глобального экстремума.

Алгоритм 1. В допустимой области D случайным образом выбирают точку $\bar{x}_1 \in D$. Приняв эту точку за исходную и используя некоторый детерминированный метод или алгоритм направленного случайного поиска, осуществляют спуск в точку локального минимума $\bar{x}_1^* \in D$, в области притяжения которого оказалась точка \bar{x}_1 .

Затем выбирается новая случайная точка $\bar{x}_2 \in D$ и по той же схеме осуществляется спуск в точку локального минимума $\bar{x}_2^* \in D$, и т. д. (рис. 7.8).



Rис. 7.8. Алгоритм 1

Поиск прекращается, как только некоторое заданное число m раз не удается найти точку локального экстремума со значением функции, меньшим предыдущих.

Алгоритм 2. Пусть получена некоторая точка локального экстремума $\bar{x}_1^* \in D$. После этого переходим к *ненаправленному случайному* поиску до получения точки \bar{x}_2 такой, что $f(\bar{x}_2) < f(\bar{x}_1^*)$.

Из точки \bar{x}_2 с помощью детерминированного алгоритма или направленного случайного поиска получаем точку локального экстремума \bar{x}_2^* , в которой заведомо выполняется неравенство $f(\bar{x}_2^*) < f(\bar{x}_1^*)$.

Далее с помощью случайного поиска определяем новую точку \bar{x}_3 , для которой справедливо неравенство $f(\bar{x}_3) < f(\bar{x}_2^*)$, и снова – спуск в точку локального экстремума \bar{x}_3^* , и т. д.

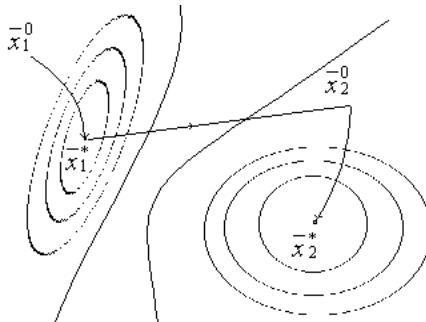
Поиск прекращается, если при генерации некоторого предельного числа новых случайных точек не удается найти лучшей, в которой значение функции меньше, чем предыдущий локальный экстремум; он тогда и принимается в качестве решения.

Алгоритм 3 (рис.7.9). Пусть \bar{x}_1^0 – некоторая исходная точка поиска в области D , из которой осуществляется спуск в точку локального экстремума \bar{x}_1^* со значением $f(\bar{x}_1^*)$. Далее из точки \bar{x}_1^* движемся либо в случайном направлении, либо в направлении $\bar{x}_1^* - \bar{x}_1^0$ до тех пор,

пока функция снова не станет убывать (выходим из области притяжения \bar{x}_1^*).

Полученная точка \bar{x}_2^0 принимается за начало следующего спуска. В результате находим новый локальный экстремум \bar{x}_2^* со значением функции $f(\bar{x}_2^*)$.

Если $f(\bar{x}_2^*) < f(\bar{x}_1^*)$, точка \bar{x}_1^* забывается и ее место занимает точка \bar{x}_2^* . Если $f(\bar{x}_2^*) \geq f(\bar{x}_1^*)$, то возвращаемся в точку \bar{x}_1^* и движемся из нее в новом случайном направлении.



Rис. 7.9. Алгоритм 3

Процесс прекращается, если не удается найти лучший локальный минимум после заданного числа попыток или не удается найти «случайное» направление, в котором функция снова начинает убывать.

Такой подход позволяет найти глобальный экстремум в случае многосвязных допустимых областей.

Алгоритм 4. В допустимой области D разбрасываем m случайных точек и выбираем из них наилучшую, т. е. ту, в которой значение функции минимально. Из выбранной точки осуществляем локальный спуск, а далее вокруг траектории спуска образуем запретную область.

В оставшейся области случайным образом разбрасываем новую совокупность случайных точек и из лучшей точки осуществляем спуск в точку локального экстремума. Вокруг новой траектории также строим запретную область и т. д. (рис. 7.10)

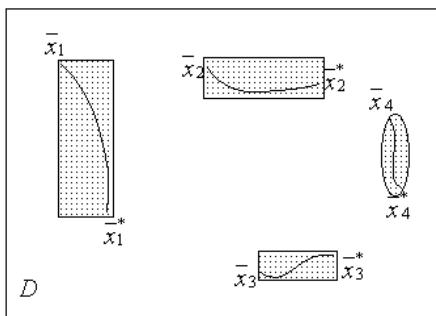


Рис. 7.10. Алгоритм 4

Поиск прекращается, если в течение заданного числа попыток не удается найти лучший локальный экстремум.

Замечание. Комбинация случайного поиска с детерминированными методами применяется не только для решения многоэкстремальных задач. Часто к такой комбинации прибегают в ситуациях, когда детерминированные методы сталкиваются с теми или иными трудностями (застревают на дне узкого оврага, в седловой точке и т.п.). Шаг в случайном направлении порой позволяет преодолеть такую ситуацию, тупиковую для детерминированного алгоритма.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Простой случайный поиск.
2. Направленный случайный поиск и ненаправленный. В чем различие?
3. Примеры направленного случайного поиска.
4. Примеры ненаправленного случайного поиска.
5. Алгоритм метода статистических градиентов.
6. Примеры построения алгоритмов глобального поиска.

8. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

8.1. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ И ТЕОРЕМЫ

Определение 1. Задача, в которой требуется минимизировать (или максимизировать) линейную форму

$$\sum_{i=1}^n c_i x_i \rightarrow \min (\max)$$

при условии, что

$$\sum_{j=1}^n a_{ij} x_i \leq b_j, \quad j = \overline{1, m},$$

или

$$\sum_{i=1}^n a_{ij} x_i = b_j, \quad j = \overline{m+1, p},$$

и

$$x_i > 0, \quad i = \overline{1, n},$$

называется задачей линейного программирования в произвольной форме записи [7, 8].

Определение 2. Задача в матричной форме вида

$$\left. \begin{array}{l} \bar{c}^T \bar{x} \rightarrow \min \max \\ A \bar{x} \leq \bar{b}, \\ \bar{x} \geq \bar{0} \end{array} \right\} \quad (1)$$

называется симметричной формой записи задачи линейного программирования.

Определение 3. Задача линейного программирования вида

$$\left. \begin{array}{l} \bar{c}^T \bar{x} \rightarrow \min \max \\ A \bar{x} = \bar{b}, \\ \bar{x} \geq \bar{0} \end{array} \right\} \quad (2)$$

называется канонической формой записи задачи линейного программирования.

Любую задачу линейного программирования можно привести к канонической форме. Например, если система ограничений задана в виде

$$A\bar{x} \leq \bar{b},$$

то можно, введя дополнительные переменные, привести ее к виду

$$A\bar{x} + E\bar{y} = \bar{b}, \quad \bar{x} \geq \bar{0}, \quad \bar{y} \geq \bar{0},$$

где $\bar{y} = [x_{n+1}, \dots, x_{n+m}]^T$. Если же ограничения в задаче заданы в виде

$$A\bar{x} \geq \bar{b},$$

то

$$A\bar{x} - E\bar{y} = \bar{b}, \quad \bar{x} \geq \bar{0}, \quad \bar{y} \geq \bar{0}.$$

Рассмотрим задачу с ограничениями $A\bar{x} \leq \bar{b}$. Этую систему ограничений можно представить в виде системы

$$\left. \begin{array}{l} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n + x_{n+1} = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,n}x_n + x_{n+2} = b_2 \\ \dots \\ a_{m,1}x_1 + a_{m,2}x_2 + \dots + a_{m,n}x_n + x_{n+m} = b_m \end{array} \right\}.$$

Введем следующие обозначения:

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}, \quad \bar{A}_1 = \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ \dots \\ a_{m,1} \end{bmatrix}, \dots, \quad \bar{A}_n = \begin{bmatrix} a_{1,n} \\ a_{2,n} \\ \dots \\ a_{m,n} \end{bmatrix}, \quad \bar{A}_{n+1} = \begin{bmatrix} 1 \\ 0 \\ \dots \\ 0 \end{bmatrix}, \dots, \quad \bar{A}_{n+m} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \end{bmatrix},$$

$$\bar{A}_0 = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix}.$$

Тогда задачу линейного программирования можно записать:

$$\sum_{i=1}^n c_i x_i \rightarrow \min (\max),$$

$$x_1 \bar{A}_1 + x_2 \bar{A}_2 + \dots + x_n \bar{A}_n + x_{n+1} \bar{A}_{n+1} + \dots + x_{n+m} \bar{A}_{n+m} = A_0,$$

$$\bar{x} \geq \bar{0}.$$

Векторы \bar{A}_i называются векторами условий, а сама задача линейного программирования называется расширенной по отношению к исходной.

Пусть D и D_1 – допустимые множества решений исходной и расширенной задач линейного программирования соответственно. Тогда любой точке множества D_1 соответствует единственная точка множества D и наоборот. В общем случае допустимое множество D исходной задачи есть проекция множества D_1 расширенной задачи на подпространство исходных переменных.

Определение 4. Набор чисел $\bar{x} = x_1, x_2, \dots, x_n$, удовлетворяющий ограничениям задачи линейного программирования, называется ее **планом**.

Определение 5. Решением задачи линейного программирования называется ее план, минимизирующий (или максимизирующий) линейную форму.

Введем понятие базисного решения. Из матрицы расширенной задачи $A_p = [\bar{A}_1, \bar{A}_2, \dots, \bar{A}_{n+m}]$ выберем m линейно независимых векторов-столбцов, которые обозначим как матрицу $B_{m \times m}$, а через $D_{m \times n}$ – обозначим матрицу из оставшихся столбцов. Тогда $A_p = B, D$, и ограничения расширенной задачи линейного программирования можно записать в виде

$$A_p \bar{x} = B \bar{x}_B + D \bar{x}_D = \bar{A}_0. \quad (3)$$

Очевидно, что столбцы матрицы B образуют базис m -мерного пространства. Поэтому вектор \bar{A}_0 и любой столбец матрицы D можно представить в виде линейной комбинации столбцов матрицы B .

Умножим (3) на B^{-1} слева:

$$B^{-1}B\bar{x}_B + B^{-1}D\bar{x}_D = B^{-1}\bar{A}_0 \quad (4)$$

и найдем отсюда \bar{x}_B :

$$\bar{x}_B = B^{-1}\bar{A}_0 - B^{-1}D\bar{x}_D. \quad (5)$$

Придавая \bar{x}_D различные значения, будем получать различные решения \bar{x}_B .

Если положить $\bar{x}_D = \bar{0}$, то

$$\bar{x}_B = B^{-1}\bar{A}_0. \quad (6)$$

Решение (6) называют **базисным решением** системы из m уравнений с $m+n$ неизвестными.

Если полученное решение содержит только положительные компоненты, то оно называется **базисным допустимым**.

Особенность допустимых базисных решений состоит в том, что они являются крайними точками допустимого множества D_1 расширенной задачи.

Если среди компонент \bar{x}_B нет нулевых, то базисное допустимое решение называется **невырожденным**.

Определение 6. План \bar{x} задачи линейного программирования будем называть опорным, если векторы условий \bar{A}_i с положительными коэффициентами линейно независимы.

То есть опорный план – это базисное допустимое решение расширенной системы, угловая точка многогранника решений.

Определение 7. Опорное решение называется **невырожденным**, если оно содержит m положительных компонент (по числу ограничений).

Невырожденный опорный план образуется пересечением n гиперплоскостей из образующих допустимую область. В случае вырожденности в угловой точке многогранника решений пересекается более чем n гиперплоскостей.

8.2. ОСНОВНАЯ ТЕОРЕМА ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

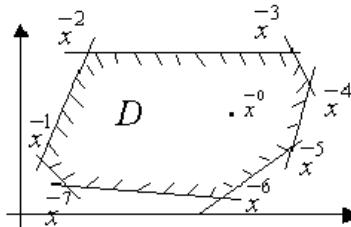
Теорема 1 [8]

1. Линейная форма $z = \bar{c}^T \bar{x}$ достигает своего минимума в угловой точке многогранника решений.

2. Если она принимает минимальное решение более чем в одной угловой точке, то она достигает того же самого значения в любой точке, являющейся выпуклой комбинацией этих угловых точек.

Доказательство. Доказательство теоремы основано на следующей лемме.

Лемма. Если D – замкнутое, ограниченное, выпуклое множество, имеющее конечное число крайних (угловых) точек, то любая точка $\bar{x} \in D$ может быть представлена в виде выпуклой комбинации крайних точек D (рис. 8.1).



Rис. 8.1. Допустимая область

1) Пусть \bar{x}^0 – некоторая внутренняя точка. Многогранник ограниченный, замкнутый, имеет конечное число угловых точек. D – допустимое множество.

Предположим, что точка \bar{x}^0 является оптимальной точкой, т. е. $z(\bar{x}^0) \leq z(\bar{x})$, $\forall \bar{x} \in D$. Предположим, что точка \bar{x}^0 не является угловой. Тогда на основании леммы точку \bar{x}^0 можно выразить через угловые точки многогранника \bar{x}^i , т. е.

$$\bar{x}^0 = \sum_{i=1}^p \alpha_i \bar{x}^i, \quad \forall \alpha_i \geq 0, \quad \sum_{i=1}^p \alpha_i = 1.$$

Так как функция $z(\bar{x})$ линейна, то

$$z(\bar{x}^0) = \sum_{i=1}^p \alpha_i z(\bar{x}^i). \quad (*)$$

Выберем среди точек \bar{x}^i ту, в которой линейная форма $z(\bar{x})$ принимает наименьшее значение. Пусть это будет точка \bar{x}^k . Обозначим минимальное значение функции в угловой точке через z^* :

$$z^* = z(\bar{x}^k) = \min_{1 \leq i \leq p} z(\bar{x}^i), \quad z(\bar{x}^1), z(\bar{x}^2), \dots, z(\bar{x}^p).$$

Подставим данное значение функции в линейную форму (*) вместо $z(\bar{x}^i)$ и получим

$$z(\bar{x}^0) \geq \sum_{i=1}^p \alpha_i z^* = z^* \sum_{i=1}^p \alpha_i = z^*.$$

Так как \bar{x}^0 – оптимальная точка, то получили противоречие: $z(\bar{x}^0) \geq z^*$ (!). Следовательно, $z(\bar{x}^0) = z(\bar{x}^k)$, $\bar{x}^0 = \bar{x}^k$ – угловая точка.

2) Предположим, что линейная форма $z(\bar{x})$ принимает минимальное значение более чем в одной угловой точке, например, в угловых точках $\bar{x}^1, \bar{x}^2, \dots, \bar{x}^q$ имеем $z(\bar{x}^1) = z(\bar{x}^2) = \dots = z(\bar{x}^q) = z^*$. Тогда если \bar{x} является выпуклой комбинацией этих точек, т. е.

$$\bar{x} = \sum_{i=1}^q \alpha_i \bar{x}^i, \quad \sum_{i=1}^q \alpha_i = 1 \quad \text{и} \quad \forall i \quad \alpha_i \geq 0,$$

$$\text{то } z(\bar{x}) = z\left(\sum_{i=1}^q \alpha_i \bar{x}^i\right) = z^* \sum_{i=1}^q \alpha_i = z^*.$$

Таким образом, если минимальное значение достигается более чем в одной угловой точке, то того же самого значения линейная форма

достигает в любой точке, являющейся выпуклой комбинацией этих угловых точек.

Теорема 2. Если известно, что система векторов условий $\bar{A}_1, \dots, \bar{A}_m$, ($m \leq n$) линейно независима и такова, что

$$x_1\bar{A}_1 + \dots + x_m\bar{A}_m = \bar{A}_0,$$

где все $x_j > 0$, то точка $\bar{x} = [x_1, \dots, x_m, 0, \dots, 0]^T$ является угловой точкой многогранника решений.

Теорема 3. Если вектор \bar{x} является угловой точкой многогранника решений, то векторы условий, соответствующие положительным компонентам вектора \bar{x} , являются линейно независимыми.

Следствия:

1) угловая точка многогранника решений имеет не более m положительных компонент вектора \bar{x} ;

2) каждой угловой точке многогранника решений соответствует m линейно независимых векторов из данной системы: $\bar{A}_1, \dots, \bar{A}_n$.

8.3. СИМПЛЕКС-МЕТОД

8.3.1. ВВЕДЕНИЕ В СИМПЛЕКС-МЕТОД

Этот метод называют еще **методом последовательного улучшения плана** [8]. Метод предназначен для решения общей задачи линейного программирования.

Пусть имеем следующую задачу:

$$Q \bar{x} = c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \min, \quad (7)$$

с системой ограничений вида

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n = b_1 \\ \dots \\ a_{m,1}x_1 + a_{m,2}x_2 + \dots + a_{m,n}x_n = b_m \end{cases} . \quad (8)$$

Разрешим эту систему относительно переменных x_1, \dots, x_m :

$$\left\{ \begin{array}{l} x_1 = a'_{1,m+1}x_{m+1} + \dots + a'_{1,n}x_n + b'_1 \\ \dots \\ x_m = a'_{m,m+1}x_{m+1} + \dots + a'_{m,n}x_n + b'_m \end{array} \right. . \quad (9)$$

Векторы условий, соответствующие x_1, \dots, x_m , образуют базис. Переменные x_1, \dots, x_m назовем базисными переменными. Остальные переменные задачи – небазисные.

Целевую функцию можно выразить через небазисные переменные:

$$Q \bar{x} = c'_{m+1}x_{m+1} + c'_{m+2}x_{m+2} + \dots + c'_n x_n + c'_0 \rightarrow \min.$$

Если приравнять небазисные переменные нулю

$$x_{m+1} = 0, x_{m+2} = 0, \dots, x_n = 0,$$

то соответствующие базисные переменные примут значения

$$x_1 = b'_1, x_2 = b'_2, \dots, x_m = b'_m.$$

Вектор \bar{x} с такими компонентами представляет собой угловую точку многогранника решений (допустимую) при условии, что $b'_i \geq 0$ (опорный план).

Теперь необходимо перейти к другой угловой точке с меньшим значением целевой функции. Для этого следует выбрать некоторые небазисную и базисную переменные так, чтобы после того как мы «поменяли их местами», значение целевой функции уменьшилось. Такой направленный перебор в конце концов приведет нас к решению задачи.

Пример 1. Пусть

$$Q \bar{x} = x_4 - x_5 \rightarrow \min$$

$$\left. \begin{array}{l} x_1 + x_4 - 2x_5 = 1, \\ x_2 - 2x_4 + x_5 = 2, \\ x_3 + 3x_4 + x_5 = 3. \end{array} \right\}$$

Выберем в качестве базисных следующие переменные: x_1, x_2, x_3 и разрешим систему относительно этих переменных. Система ограничений примет вид

$$\left. \begin{array}{l} x_1 = 1 - x_4 + 2x_5, \\ x_2 = 2 + 2x_4 - x_5, \\ x_3 = 3 - 3x_4 - x_5. \end{array} \right\}$$

Переменные x_4, x_5 являются небазисными. Если взять $x_4 = 0$ и $x_5 = 0$, то получим угловую точку (опорный план)

$$\bar{x}^1 = \begin{pmatrix} 1 & 2 & 3 & 0 & 0 \end{pmatrix}^T,$$

которому соответствует $Q \bar{x}^1 = 0$.

Значение целевой функции можно уменьшить за счет увеличения x_5 . При увеличении x_5 x_1 также увеличивается, а x_2 и x_3 – уменьшаются. Причем x_2 может стать отрицательной раньше. Поэтому, вводя в базис переменную x_5 , одновременно x_2 исключаем из базиса. В результате после очевидных преобразований получим следующие выражения для новой системы базисных переменных и целевой функции:

$$\left. \begin{array}{l} x_5 = 2 - x_2 + 2x_4, \\ x_1 = 5 - 2x_2 + 3x_4, \\ x_3 = 1 + x_2 - 5x_4, \end{array} \right\}$$

$$Q \bar{x} = -2 - x_4 + x_2 \rightarrow \min.$$

Соответствующий опорный план $\bar{x}^2 = \begin{pmatrix} 5 & 0 & 1 & 0 & 2 \end{pmatrix}^T$ и $Q \bar{x}^2 = -2$.

Целевую функцию можно уменьшить за счет увеличения x_4 . Увеличение x_4 приводит к уменьшению только x_3 . Поэтому вводим в

базис переменную x_4 , а x_3 исключаем из базиса. В результате получим следующие выражения для новой системы базисных переменных и целевой функции:

$$\left. \begin{aligned} x_4 &= \frac{1}{5} + \frac{1}{5}x_2 - \frac{1}{5}x_3, \\ x_1 &= \frac{28}{5} - \frac{7}{5}x_2 - \frac{3}{5}x_3, \\ x_5 &= \frac{12}{5} - \frac{3}{5}x_2 - \frac{2}{3}x_3, \end{aligned} \right\}$$

$$Q \bar{x} = -\frac{11}{5} + \frac{4}{5}x_2 + \frac{1}{5}x_3 \rightarrow \min.$$

Соответствующий опорный план $\bar{x}^3 = \left[\frac{28}{5} \quad 0 \quad 0 \quad \frac{1}{5} \quad \frac{12}{5} \right]^T$ и значение целевой функции $Q \bar{x}^3 = -\frac{11}{5}$. Так как все коэффициенты при небазисных переменных в целевой функции неотрицательны, то нельзя уменьшить целевую функцию за счет увеличения x_2 или x_3 . Следовательно, полученный план \bar{x}^3 является оптимальным.

Пример 2. Пусть имеем задачу

$$\left. \begin{aligned} Q \bar{x} &= -x_1 - x_2 \rightarrow \min \\ x_3 &= 1 + x_1 - x_2 \\ x_4 &= 2 - x_1 + 2x_2 \\ \bar{x} &\geq 0 \end{aligned} \right\}.$$

Переменные x_3, x_4 – базисные, а x_1, x_2 – небазисные переменные.

Опорный план $\bar{x}^0 = [0 \quad 0 \quad 1 \quad 2]^T$, $Q \bar{x}^0 = 0$.

Теперь вводим в базис переменную x_1 , а x_4 исключаем из базиса. В результате получим следующие выражения для базисных переменных и целевой функции:

$$\left. \begin{array}{l} x_1 = 2 + 2x_2 - x_4, \\ x_3 = 3 + x_2 - x_4, \end{array} \right\}$$

$$Q \bar{x} = -2 - 3x_2 + x_4.$$

Опорный план $\bar{x}^1 = [2 \ 0 \ 3 \ 0]^T$, значение целевой функции $Q \bar{x}^1 = -2$.

Теперь можно заметить, что при увеличении x_2 значения переменных x_1 и x_3 также возрастают, т. е. при $x_2 \rightarrow \infty$ в допустимой области $Q \bar{x} \rightarrow -\infty$ (задача не имеет решения).

Замечание. В процессе поиска допустимого плана может быть выявлена противоречивость системы ограничений.

8.3.2. АЛГОРИТМ СИМПЛЕКС-МЕТОДА

Формализованный алгоритм симплекс-метода состоит из двух основных этапов [8]:

1) построение опорного плана;

2) построение оптимального плана.

Проиллюстрируем алгоритм на рассмотренном ранее примере:

$$Q \bar{x} = x_4 - x_5 \rightarrow \min$$

$$\left. \begin{array}{l} x_1 + x_4 - 2x_5 = 1, \\ x_2 - 2x_4 + x_5 = 2, \\ x_3 + 3x_4 + x_5 = 3, \end{array} \right\}$$

$$\bar{x} \geq \bar{0}.$$

В случае базисных переменных x_1, x_2, x_3 начальная симплексная таблица для данного примера будет выглядеть следующим образом:

	$-x_4$	$-x_5$	1
$x_1 =$	1	-2	1
$x_2 =$	-2	1	2
$x_3 =$	3	1	3
$Q \bar{x} =$	-1	1	0

Она уже соответствует опорному плану $\bar{x}^1 = 1 \ 2 \ 3 \ 0 \ 0^T$ (столбец свободных членов).

Построение оптимального плана. Для того чтобы опорный план был оптимальен при минимизации целевой функции, необходимо, чтобы коэффициенты в строке целевой функции были неположительными (в случае максимизации – неотрицательными), т. е. при поиске минимума мы должны освободиться от положительных коэффициентов в строке $Q \bar{x}$.

Выбор разрешающего элемента. Если при поиске минимума в строке целевой функции есть коэффициенты больше нуля, то выбираем столбец с положительным коэффициентом в строке целевой функции в качестве разрешающего. Пусть это столбец с номером l .

Для выбора разрешающей строки (разрешающего элемента) среди положительных коэффициентов разрешающего столбца выбираем тот (ту строку), для которого отношение коэффициента в столбце свободных членов к коэффициенту в разрешающем столбце минимально:

$$\frac{b_r}{a_{rl}} = \min \left\{ \frac{b_i}{a_{il}} \mid a_{il} \geq 0 \right\}.$$

Тогда a_{rl} – разрешающий (направляющий) элемент, строка r – разрешающая.

Для перехода к следующей симплексной таблице (следующему опорному плану с меньшим значением целевой функции) делается шаг модифицированного жорданова исключения с разрешающим элементом a_{rl} .

Если в разрешающем столбце нет положительных коэффициентов, то целевая функция не ограничена снизу (при максимизации – не ограничена сверху).

Шаг модифицированного жорданова исключения над симплексной таблицей:

- 1) на месте разрешающего элемента ставится 1 и делится на разрешающий элемент;
- 2) остальные элементы разрешающего столбца меняют знак на противоположный и делятся на разрешающий элемент;
- 3) остальные элементы разрешающей строки делятся на разрешающий элемент;
- 4) все остальные элементы симплексной таблицы вычисляются по следующей формуле:

$$a_{ij} = \frac{a_{ij}a_{rl} - a_{rj}a_{il}}{a_{rl}} = a_{ij} - \frac{a_{rj}a_{il}}{a_{rl}}.$$

	$-x_4$	$-x_5$	1	Разрешающий элемент, соответствующий замене базисной переменной x_2 на небазисную переменную x_5
$x_1 =$	1	-2	1	
$x_2 =$	-2	1	2	
$x_3 =$	3	1	3	
$Q \bar{x} =$	-1	1	0	

	$-x_4$	$-x_2$	1	Разрешающий элемент, соответствующий замене базисной переменной x_3 на небазисную переменную x_4
$x_1 =$	-3	2	5	
$x_5 =$	-2	1	2	
$x_3 =$	5	-1	1	
$Q \bar{x} =$	1	-1	-2	

	$-x_3$	$-x_2$	1	Все коэффициенты в строке целевой функции отрицательны, т. е. мы нашли оптимальное решение
$x_1 =$	3/5	7/5	28/5	
$x_5 =$	2/5	3/5	12/5	
$x_4 =$	1/5	-1/5	1/5	
$Q \bar{x} =$	-1/5	-4/5	-11/5	

Построение опорного плана. Пусть необходимо решить задачу:

$$Q \bar{x} = c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \min \max$$

$$\left\{ \begin{array}{l} a_{1,1}x_1 + \dots + a_{1,n}x_n = b_1 \\ \dots \\ a_{m,1}x_1 + \dots + a_{m,n}x_n = b_m \\ a_{m+1,1}x_1 + \dots + a_{m+1,n}x_n \leq b_{m+1} \\ \dots \\ a_{m+p,1}x_1 + \dots + a_{m+p,n}x_n \leq b_{m+p}. \end{array} \right.$$

Введем дополнительные переменные, чтобы преобразовать ограничения-неравенства к равенствам. В ограничениях-равенствах дополнительные переменные должны быть нулевыми. Тогда система ограничений принимает вид

$$\left\{ \begin{array}{l} 0 = b_1 - a_{1,1}x_1 - \dots - a_{1,n}x_n \\ \dots \\ 0 = b_m - a_{m,1}x_1 - \dots - a_{m,n}x_n \\ x_{n+1} = b_{m+1} - a_{m+1,1}x_1 - \dots - a_{m+1,n}x_n \\ \dots \\ x_{n+p} = b_{m+p} - a_{m+p,1}x_1 - \dots - a_{m+p,n}x_n, \end{array} \right.$$

где $x_{n+i} \geq 0 \quad \forall i = 1, p$.

В качестве базисных переменных будем брать систему дополнительно введенных переменных. Тогда симплексная таблица для преобразованной задачи будет иметь следующий вид:

	$-x_1$	$-x_2$	$-x_S$	$-x_n$	1
$0 =$	$a_{1,1}$	$a_{1,2}$	$a_{1,S}$	$a_{1,n}$	b_1
....
$0 =$	$a_{m,1}$	$a_{m,2}$	$a_{m,S}$	$a_{m,n}$	b_m
$x_{m+1} =$	$a_{m+1,1}$	$a_{m+1,2}$	$a_{m+1,S}$	$a_{m+1,n}$	b_{m+1}
....
$x_{m+p} =$	$a_{m+p,1}$	$a_{m+p,2}$	$a_{m+p,S}$	$a_{m+p,n}$	b_{m+p}
$Q \bar{x} =$	$-c_1$	$-c_2$	$-c_S$	$-c_n$	0

Правила выбора разрешающего элемента при поиске опорного плана

1. При условии отсутствия «0-строк» (ограничений-равенств) и «свободных» переменных (т.е. переменных, на которые не наложено требование неотрицательности).

- Если в столбце свободных членов симплексной таблицы нет отрицательных элементов, то опорный план найден.

- Есть отрицательные элементы в столбце свободных членов, например $b_i < 0$. В такой строке ищем отрицательный коэффициент a_{il} , и этим самым определяем разрешающий столбец l . Если не найдем отрицательный a_{il} , то система ограничений несовместна (противоречива).

- В качестве разрешающей выбираем строку, которой соответствует минимальное отношение

$$\frac{b_r}{a_{rl}} = \min_i \left\{ \frac{b_i}{a_{il}} \middle| \frac{b_i}{a_{il}} > 0 \right\},$$

где r – номер разрешающей строки. Таким образом, a_{rl} – разрешающий элемент.

- После того как разрешающий элемент найден, делаем шаг модифицированного жорданова исключения с направляющим элементом a_{rl} и переходим к следующей симплексной таблице.

2. В случае присутствия ограничений-равенств и «свободных» переменных поступают следующим образом.

- Выбирают разрешающий элемент в «0-строке» и делают шаг модифицированного жорданова исключения, после чего вычеркивают этот разрешающий столбец. Данную последовательность действий продолжают до тех пор, пока в симплексной таблице остается хотя бы одна «0-строка» (при этом таблица сокращается).

- Если же присутствуют и свободные переменные, то необходимо данные переменные сделать базисными. И после того как свободная переменная станет базисной, далее в процессе определения разрешающего элемента при поиске опорного и оптимального планов данная строка не учитывается (но преобразуется).

8.3.3. ВЫРОЖДЕННОСТЬ В ЗАДАЧАХ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Рассматривая симплекс-метод, мы предполагали, что задача линейного программирования является невырожденной, т. е. каждый опорный план содержит ровно m положительных компонент, где m – число ограничений в задаче. В вырожденном опорном плане число положительных компонент оказывается меньше числа ограничений: некоторые базисные переменные, соответствующие данному опорному плану, принимают нулевые значения.

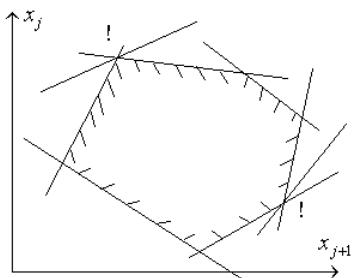


Рис. 8.2. Допустимая область с вырожденными опорными планами

Используя геометрическую интерпретацию для простейшего случая, когда $n - m = 2$ (число небазисных переменных равно 2), легко отличить вырожденную задачу от невырожденной. В вырожденной задаче в одной вершине многогранника условий пересекается более двух прямых, описываемых уравнениями вида $x_i = 0$ (рис. 8.2). Это значит, что одна или несколько сторон многоугольника, определяющего допустимую область, стягиваются в точку.

Аналогично при $n - m = 3$ в вырожденной задаче в одной вершине пересекается более трех плоскостей $x_i = 0$.

В предположении о невырожденности задачи находилось только одно значение, соответствующее минимальному значению

$\theta = \min_i \left\{ \frac{b_i}{a_{il}} \mid \frac{b_i}{a_{il}} > 0 \right\}$, по которому определялся индекс выводимого из базиса вектора условий (переменной, выводимой из числа базисных).

В вырожденной задаче $\min_i \left\{ \frac{b_i}{a_{il}} \mid \frac{b_i}{a_{il}} > 0 \right\}$ может достигаться на нескольких индексах сразу (для нескольких строк). В этом случае в находящемся опорном плане несколько базисных переменных будут нулевыми.

Если задача линейного программирования оказывается вырожденной, то при плохом выборе вектора условий, выводимого из базиса, может возникнуть бесконечное движение по базисам одного и того же опорного плана. Это так называемое явление зацикливания. Хотя в практических задачах линейного программирования зацикливание – явление крайне редкое, возможность его не исключена.

Один из приемов борьбы с вырожденностью состоит в преобразовании задачи путем «незначительного» изменения вектора правых частей системы ограничений на величины ε_i , таким образом, чтобы задача стала невырожденной и в то же время, чтобы это изменение не повлияло реально на оптимальный план задачи.

Чаще реализуемые алгоритмы включают в себя некоторые простые правила, снижающие вероятность возникновения зацикливания или его преодоления.

Пусть переменную x_j необходимо сделать базисной. Рассмотрим множество индексов E_0 , состоящее из тех i , для которых достигается

$\theta_0 = \min_i \left\{ \frac{b_i}{a_{il}} \middle| \frac{b_i}{a_{il}} > 0 \right\}$. Множество индексов i , для которых выполняется данное условие, обозначим через E_0 . Если E_0 состоит из одного элемента, то из базиса исключается вектор условий A_i (переменная x_i делается небазисной).

Если E_0 состоит более чем из одного элемента, то составляется множество E_1 , которое состоит из $i \in E_0$, на которых достигается

$\theta_1 = \min_{i \in E_0} \left\{ \frac{a_{il}}{a_{il}} \right\}$. Если E_1 состоит из одного индекса k , то из базиса вы-

водится переменная x_k . В противном случае составляется множество E_2 и т. д.

Практически правилом надо пользоваться, если зацикливание уже обнаружено.

8.4. ДВОЙСТВЕННОСТЬ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

8.4.1. ПОНЯТИЕ ДВОЙСТВЕННОЙ ЗАДАЧИ

Рассмотрим задачу максимизации линейной формы и одновременно задачу минимизации:

$$\left. \begin{array}{l} Q \bar{x} = \bar{p}^T \bar{x} \rightarrow \max, \\ A\bar{x} \leq \bar{b}, \\ \bar{x} \geq 0. \end{array} \right\} \quad (10)$$

$$\left. \begin{array}{l} W \bar{u} = \bar{b}^T \bar{u} \rightarrow \min, \\ A^T \bar{u} \geq \bar{p}, \\ \bar{u} \geq 0. \end{array} \right\} \quad (11)$$

Задача (11) называется двойственной по отношению к прямой (10) (и наоборот!) [8].

Пример. Предприятие выпускает три вида продукции. Каждая продукция требует обработки на трех различных типах установок. Ресурс времени каждого типа установок ограничен. Известна прибыль от единицы каждого вида продукции p_1, p_2, p_3 . Если количество выпускаемой продукции каждого вида x_1, x_2, x_3 , тогда необходимо максимизировать прибыль

$$Q \bar{x} = p_1 x_1 + p_2 x_2 + p_3 x_3 \rightarrow \max$$

при ограничениях следующего вида:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \leq b_1,$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \leq b_2,$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \leq b_3,$$

$$\bar{x} \geq \bar{0},$$

где b_1, b_2, b_3 – ресурсы времени установок первого, второго и третьего типов. Величины a_{ij} определяют количество ресурса времени уста-

новки i -го типа, которое необходимо для выпуска одной единицы продукции j -го вида.

Двойственная к ней задача будет иметь вид

$$W \bar{u} = b_1 u_1 + b_2 u_2 + b_3 u_3 \rightarrow \min$$

при ограничениях:

$$\begin{cases} a_{11}u_1 + a_{21}u_2 + a_{31}u_3 \geq p_1, \\ a_{12}u_1 + a_{22}u_2 + a_{32}u_3 \geq p_2, \\ a_{13}u_1 + a_{23}u_2 + a_{33}u_3 \geq p_3, \\ \bar{u} \geq \bar{0}. \end{cases}$$

Здесь u_1 – это оценка (цена), соответствующая одной единице ограниченного ресурса по первой установке. И она равна величине, на которую могла бы возрасти суммарная прибыль, если бы количество этого ограниченного ресурса увеличилось на единицу, и если это увеличение было бы использовано оптимально. Иными словами, u_1 – это количество прибыли, недополученной из-за нехватки единицы ограниченного ресурса b_1 . Аналогичным образом можно интерпретировать смысл величин u_2 и u_3 .

8.4.2. ПРЕОБРАЗОВАНИЯ ПРИ РЕШЕНИИ ПРЯМОЙ И ДВОЙСТВЕННОЙ ЗАДАЧ

Пусть имеются прямая и двойственная задачи следующего вида:

Прямая задача:

$$\begin{aligned} Q \bar{x} &= \bar{p}^T \bar{x} \rightarrow \max \\ A\bar{x} &\leq \bar{b} \\ \bar{x} &\geq \bar{0} \end{aligned}$$

Двойственная к ней задача:

$$\begin{aligned} W \bar{u} &= \bar{b}^T \bar{u} \rightarrow \min \\ A^T \bar{u} &\geq \bar{p} \\ \bar{u} &\geq \bar{0} \end{aligned}$$

Представим ограничения в виде

$$\begin{aligned} \bar{y} &= -A\bar{x} + \bar{b} \geq \bar{0} \\ \bar{x} &\geq \bar{0} \end{aligned}$$

$$\begin{aligned} \bar{v} &= A^T \bar{u} - \bar{p} \geq 0 \\ \bar{u} &\geq \bar{0} \end{aligned}$$

Для ограничений прямой задачи симплексная таблица имеет вид

	$-x_1$...	$-x_s$...	$-x_n$	1
$y_1 =$	a_{11}	...	a_{1s}	...	a_{1n}	b_1
...
$y_r =$	a_{r1}	...	a_{rs}	...	a_{rn}	b_r
...
$y_m =$	a_{m1}	...	a_{ms}	...	a_{mn}	b_m
$Q \bar{x} =$	$-p_1$...	$-p_s$...	$-p_n$	0

Пусть a_{rs} – разрешающий элемент. Сделаем шаг модифицированного жорданова исключения и получим таблицу:

	$-x_1$...	$-y_r$...	$-x_n$	1
$y_1 =$	b_{11}	...	$-a_{1s}$...	b_{1n}	$b_{1,n+1}$
...
$x_s =$	a_{r1}	...	1	...	a_{rn}	b_r
...
$y_m =$	b_{m1}	...	$-a_{ms}$...	b_{mn}	$b_{m,n+1}$
$Q \bar{x} =$	$b_{m+1,1}$...	p_s	...	$b_{m+1,n}$	$b_{m+1,n+1}$,

где $b_{ij} = a_{ij}a_{rs} - a_{rj}a_{is}$, и всю данную таблицу следует разделить еще на a_{rs} .

Симплексную таблицу для двойственной задачи запишем, развернув ее на 90° . Получаем:

	$v_1 =$...	$v_s =$...	$v_n =$	$W =$
u_1	a_{11}	...	a_{1s}	...	a_{1n}	b_1
...
u_r	a_{r1}	...	a_{rs}	...	a_{rn}	b_r
...
u_m	a_{m1}	...	a_{ms}	...	a_{mn}	b_m
1	$-p_1$...	$-p_s$...	$-p_n$	0

Пусть a_{rs} – направляющий элемент. Сделаем шаг обыкновенного жорданова исключения (отличие от модифицированного состоит в том, что элементы в разрешающей строке меняют знаки, а в столбце знаки сохраняются; в остальном преобразование остается тем же):

	$v_1 =$	\dots	$u_r =$	\dots	$v_n =$	$W =$
u_1	b_{11}	\dots	$-a_{1s}$	\dots	b_{1n}	$b_{1,n+1}$
\dots	\dots	\dots	\dots	\dots	\dots	\dots
v_s	a_{r1}	\dots	1	\dots	a_{rn}	b_r
\dots	\dots	\dots	\dots	\dots	\dots	\dots
u_m	b_{m1}	\dots	$-a_{ms}$	\dots	b_{mn}	$b_{m,n+1}$
1	$b_{m+1,1}$	\dots	p_s	\dots	$b_{m+1,n}$	$b_{m+1,n+1}$

Здесь $b_{ij} = a_{ij}a_{rs} - a_{rj}a_{is}$, и всю данную таблицу также следует разделить еще на a_{rs} .

Замечание. Не следует забывать при преобразованиях, что в данном случае у нас таблица развернута.

Таким образом, нетрудно заметить, что шаг модифицированного жорданова исключения над симплексной таблицей прямой задачи соответствует шагу обыкновенного жорданова исключения над симплексной таблицей двойственной задачи. Эти взаимно двойственные задачи можно совместить в одной симплексной таблице:

	$v_1 = -x_1$	\dots	$v_s = -x_s$	\dots	$v_n = -x_n$	$W = 1$
$u_1 \ y_1 =$	a_{11}	\dots	a_{1s}	\dots	a_{1n}	b_1
\dots	\dots	\dots	\dots	\dots	\dots	\dots
$u_r \ y_r =$	a_{r1}	\dots	a_{rs}	\dots	a_{rn}	b_r
\dots	\dots	\dots	\dots	\dots	\dots	\dots
$u_m \ y_m =$	a_{m1}	\dots	a_{ms}	\dots	a_{mn}	b_m
1 $Q \bar{x} =$	$-p_1$	\dots	$-p_s$	\dots	$-p_n$	0.

Можно показать, что, решая основную задачу линейного программирования, решаем и двойственную к ней. И наоборот. Причем $\max Q = \min W$.

8.4.3. ТЕОРЕМЫ ДВОЙСТВЕННОСТИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Основная теорема [8]. Пусть рассматривается пара двойственных задач:

$$\left. \begin{array}{l} Q \bar{x} = \bar{p}^T \bar{x} \rightarrow \max, \\ A\bar{x} \leq \bar{b}, \\ \bar{x} \geq \bar{0}. \end{array} \right\} \quad (12)$$

$$\left. \begin{array}{l} W \bar{u} = \bar{b}^T \bar{u} \rightarrow \min, \\ A^T \bar{u} \geq \bar{p}, \\ \bar{u} \geq \bar{0}. \end{array} \right\} \quad (13)$$

Если одна из этих задач обладает оптимальным решением, то и двойственная к ней задача также имеет оптимальное решение. Причем экстремальные значения соответствующих линейных форм равны:
 $\max Q = \min W$.

Если же у одной из этих задач линейная форма не ограничена, то двойственная к ней задача противоречива.

Доказательство. Пусть основная задача (12) имеет конечное решение и получена окончательная симплексная таблица:

		$u_1 = \dots$	$u_s = \dots$	$v_{s+1} = \dots$	$v_n = \dots$	$W =$
v_1	$x_1 =$	$b_{1,1} \dots$	$b_{1,s} \dots$	$b_{1,s+1} \dots$	$b_{1,n} \dots$	1
...
v_s	$x_s =$	$b_{s,1} \dots$	$b_{s,s} \dots$	$b_{s,s+1} \dots$	$b_{s,n} \dots$	$b_{s,n+1}$
u_{s+1}	$y_{s+1} =$	$b_{s+1,1} \dots$	$b_{s+1,s} \dots$	$b_{s+1,s+1} \dots$	$b_{s+1,n} \dots$	$b_{s+1,n+1}$
...
u_m	$y_m =$	$b_{m,1} \dots$	$b_{m,s} \dots$	$b_{m,s+1} \dots$	$b_{m,n} \dots$	$b_{m,n+1}$
1	$Q =$	$q_1 \dots$	$q_s \dots$	$q_{s+1} \dots$	$q_n \dots$	q_0

Так как данная таблица, по предположению, соответствует оптимальному решению задачи (12), то $b_{1,n+1} \geq 0, \dots, b_{m,n+1} \geq 0$ и $q_1 \geq 0, \dots, q_n \geq 0$. При этом $\max Q = q_0$ достигается при $y_1 = \dots = y_s = x_{s+1} = \dots = x_n = 0$.

Рассмотрим полученную таблицу двойственной задачи. Полагая значения переменных слева (небазисных) равными нулю.

$$v_1 = \dots = v_s = u_{s+1} = \dots = u_m = 0,$$

найдем $u_1 = q_1 \geq 0, \dots, u_s = q_s \geq 0, v_{s+1} = q_{s+1} \geq 0, \dots, v_n = q_n \geq 0$. Следовательно, получено опорное решение:

$$u_1 = q_1, \dots, u_s = q_s, u_{s+1} = 0, \dots, u_m = 0.$$

Из последнего столбца

$$W = b_{1,n+1}v_1 + \dots + b_{s,n+1}v_s + b_{s+1,n+1}u_{s+1} + \dots + b_{m,n+1}u_m + q_0$$

в точке

$$v_1 = \dots = v_s = u_{s+1} = \dots = u_m = 0$$

будет минимальным в силу того, что $b_{i,n+1} \geq 0 \quad \forall i, i = \overline{1, m}$. Следовательно, $\max Q = \min W$.

Пусть теперь линейная форма прямой задачи не ограничена, т. е. для некоторой верхней переменной, например y_s , соответствующий коэффициент $q_s < 0$, а все коэффициенты этого столбца симплексной таблицы неположительны: $b_{1,s} \leq 0, b_{2,s} \leq 0, \dots, b_{m,s} \leq 0$. Тогда из таблицы для двойственной задачи:

$$u_s = b_{1,s}v_1 + \dots + b_{s,s}v_s + b_{s+1,s}u_{s+1} + \dots + b_{m,s}u_m + q_s \leq q_s < 0,$$

т. е. система ограничений двойственной задачи противоречива, поскольку из неотрицательности $v_1, \dots, v_s, u_{s+1}, \dots, u_m$ следует неположительность u_s (нельзя сделать ее положительной), т. е. система несовместна.

Теорема доказана.

Вторая теорема двойственности [8]. Если хотя бы одно оптимальное решение одной из двойственных задач обращает i -е ограничение этой задачи в строгое неравенство, то i -я компонента (т. е. x_i или u_i) каждого оптимального решения второй двойственной задачи равна нулю.

Если же i -я компонента хотя бы одного оптимального решения одной из двойственных задач положительна, то каждое оптимальное решение другой двойственной задачи обращает i -е ограничение в строгое равенство, т. е. оптимальные решения \bar{x}^* и \bar{u}^* пары двойственных задач удовлетворяют условиям

$$x_j^* \left[\sum_{i=1}^m a_{ij} u_i^* - p_j \right] = 0, \quad j = \overline{1, n}, \quad (14)$$

$$u_i^* \left[\sum_{j=1}^n a_{ij} x_j^* - b_i \right] = 0, \quad i = \overline{1, m}. \quad (15)$$

Доказательство. Пусть \bar{x}^* и \bar{u}^* – оптимальные решения пары двойственных задач. Тогда для

$$Q(\bar{x}) = \sum_{j=1}^n p_j x_j \rightarrow \max,$$

$$W(\bar{u}) = \sum_{i=1}^m b_i u_i \rightarrow \min$$

они удовлетворяют следующим ограничениям:

$$\left. \begin{array}{l} a_{i1} x_1^* + a_{i2} x_2^* + \dots + a_{in} x_n^* \leq b_i, \quad i = \overline{1, m}, \\ x_j^* \geq 0, \quad j = \overline{1, n}, \\ a_{1j} u_1^* + a_{2j} u_2^* + \dots + a_{mj} u_m^* \geq p_j, \quad j = \overline{1, n}, \\ u_i^* \geq 0, \quad i = \overline{1, m}. \end{array} \right\} \quad (16)$$

Умножим (16) соответственно на u_i^* и x_j^* и просуммируем полученные выражения:

$$\sum_{j=1}^n p_j x_j^* \leq \sum_{i=1}^m \sum_{j=1}^n a_{ij} u_i^* x_j^* \leq \sum_{i=1}^m b_i u_i^*. \quad (17)$$

Из основной теоремы двойственности следует

$$\sum_{j=1}^n p_j x_j^* = \sum_{i=1}^m b_i u_i^*. \quad (18)$$

И с учетом (17) получаем:

$$\begin{aligned} \sum_{j=1}^n p_j x_j^* &= \sum_{j=1}^n \sum_{i=1}^m a_{ij} u_i^* x_j^*, \\ \sum_{i=1}^m b_i u_i^* &= \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_j^* u_i^*. \end{aligned}$$

Первое из этих выражений можем переписать в виде

$$\sum_{j=1}^n x_j^* \left(\sum_{i=1}^m a_{ij} u_i^* - p_j \right) = 0,$$

и так как все x_j^* и выражения в скобках неотрицательны, то, опуская \sum , получим

$$x_j^* \left(\sum_{i=1}^m a_{ij} u_i^* - p_j \right) = 0, \quad j = \overline{1, n}.$$

Аналогично получим

$$u_i^* \left(\sum_{j=1}^n a_{ij} x_j^* - b_i \right) = 0, \quad i = \overline{1, m}.$$

Что и требовалось доказать.

Справедлива и обратная теорема.

8.4.4. МЕТОД ПОСЛЕДОВАТЕЛЬНОГО УТОЧНЕНИЯ ОЦЕНОК

Иногда его называют еще двойственным симплекс-методом [8]. Ранее говорилось, что одновременно с решением прямой задачи решается и двойственная задача. Если проследить за получающимися преобразованиями двойственной таблицы и переменными u_i и x_j , записав таблицу для двойственной задачи в обычном виде, то получим описание нового метода – метода последовательного уточнения оценок.

Пусть дана задача:

$$\begin{aligned} W \bar{u} &= b_1 u_1 + \dots + b_r u_r + \dots + b_m u_m \rightarrow \min \\ v_1 &= a_{11} u_1 + \dots + a_{r1} u_r + \dots + a_{m1} u_m - p_1 \geq 0, \\ &\dots \\ v_s &= a_{1s} u_1 + \dots + a_{rs} u_r + \dots + a_{ms} u_m - p_r \geq 0, \\ &\dots \\ v_n &= a_{1n} u_1 + \dots + a_{rn} u_2 + \dots + a_{m,n} u_m - p_n \geq 0, \\ \bar{u} &\geq \bar{0}. \end{aligned}$$

Симплексная таблица, построенная для данной задачи, будет иметь вид:

	u_1	...	u_r	...	u_m	1
$v_1 =$	a_{11}	...	a_{1r}	...	a_{1m}	$-p_1$
...
$v_s =$	a_{s1}	...	a_{sr}	...	a_{sm}	$-p_s$
...
$v_n =$	a_{n1}	...	a_{nr}	...	a_{nm}	$-p_n$
$W =$	b_1	...	b_r	...	b_m	0

В методе последовательного уточнения оценок сначала избавляются от отрицательности в W -строке (получают псевдоплан), а затем, перебирая псевдопланы, ищут оптимальный план (первый найденный опорный).

Правило выбора разрешающего элемента для избавления от отрицательности в W -строке.

1. Если все коэффициенты W -строки неотрицательны, то 0 является оценкой снизу для целевой функции W и можно переходить к отысканию оптимального решения. Иначе выбираем некоторый $b_r < 0$ и рассматриваем r -й столбец.

2. Находим в r -м столбце какой-нибудь из отрицательных элементов, например, $a_{rs} < 0$. Тогда строку с номером s , содержащую a_{rs} , выбираем в качестве разрешающей строки. Если все коэффициенты r -го столбца неотрицательны, то либо W неограничена снизу ($W \rightarrow -\infty$), либо система ограничений противоречива. (Из противоречивости двойственной не следует неограниченность прямой задачи.)

3. Находим неотрицательные отношения коэффициентов W -строки к коэффициентам разрешающей (s -й) строки. В качестве разрешающего берем тот элемент разрешающей s -й строки, для которого это отношение положительно и минимально, т. е. выбираем некоторый коэффициент a_{sk} , для которого

$$\frac{b_k}{a_{sk}} = \min_j \left\{ \frac{b_j}{a_{sj}} \middle| \frac{b_j}{a_{sj}} > 0 \right\}.$$

Выбрав разрешающий элемент, делаем шаг обыкновенного жорданова исключения. Указанная последовательность действий выполняется до тех пор, пока все коэффициенты W -строки не станут неотрицательными. Например, будет получена следующая таблица:

	v_1	v_s	u_{s+1}	u_m	1
$u_1 =$	b_{11}	b_{s1}	$b_{s+1,1}$	b_{m1}	q_1
....
$u_s =$	b_{1s}	b_{ss}	$b_{s+1,s}$	b_{ms}	q_s
$v_{s+1} =$	$b_{1,s+1}$	$b_{s,s+1}$	$b_{s+1,s+1}$	$b_{m,s+1}$	q_{s+1}
....
$v_n =$	$b_{1,n}$	$b_{s,n}$	$b_{s+1,n}$	b_{mn}	q_n
$W =$	b_1	b_s	b_{s+1}	b_m	q_0

Если все q_1, \dots, q_n – неотрицательны, то таблица соответствует оптимальному решению и $q_0 = \min W$, иначе, q_0 – оценка снизу для W .

Правило выбора разрешающего элемента при поиске оптимального решения

1. В качестве разрешающей строки берем строку, содержащую отрицательный коэффициент, например, $q_s < 0$, и строка с номером s будет разрешающей.

2. В качестве разрешающего выбираем тот положительный коэффициент b_{ks} строки s , для которого

$$\frac{b_k}{b_{sk}} = \min_j \left\{ \frac{b_j}{b_{sj}} \middle| \frac{b_j}{b_{sj}} > 0 \right\}.$$

Если в строке с номером s нет положительных коэффициентов, то *ограничения задачи противоречивы*.

После выбора разрешающего элемента делаем шаг обыкновенного жорданова исключения.

После конечного числа шагов либо найдем оптимальное решение, либо убедимся в противоречивости ограничений задачи.

Замечание. Если в симплекс-методе мы приближаемся к оптимальному решению при поиске минимума *сверху*, передвигаясь по опорным планам, то в методе последовательного уточнения оценок при поиске минимума приближаемся к оптимальному решению *снизу*, причем промежуточные планы (*псевдопланы*) не являются опорными (лежат вне многогранника решений). Первое же допустимое решение (опорный план) будет оптимальным.

Пример. Решить следующую задачу методом последовательного уточнения оценок:

$$L(\bar{x}) = -2x_1 - x_2 \rightarrow \min,$$

$$\begin{aligned} x_1 - 2x_2 + 3 &\geq 0, \\ -3x_1 - 7x_2 + 21 &\geq 0, \\ -x_1 + x_2 + 2 &\geq 0, \\ -5x_1 - 4x_2 + 20 &\geq 0, \\ x_i &\geq 0, \quad i = 1, 2. \end{aligned}$$

	x_1	x_2	1
$y_1 =$	1	-2	3
$y_2 =$	-3	-7	21
$y_3 =$	-1	1	2
$y_4 =$	-5	-4	20
$L =$	-2	-1	0

	y_3	x_2	1
$y_1 =$	-1	-1	5
$y_2 =$	3	-10	15
$x_1 =$	-1	1	2
$y_4 =$	5	-9	10
$L =$	2	-3	-4

	y_3	y_1	1
$x_2 =$	-1	-1	5
$y_2 =$	13	10	-35
$x_1 =$	-2	-1	7
$y_4 =$	14	9	-35
$L =$	5	3	-19

	y_3	y_2	1
$x_2 =$	0,3	-0,1	1,5
$y_1 =$	-1,3	0,1	3,5
$x_1 =$	-0,7	-0,1	3,5
$y_4 =$	2,3	0,9	-3,5
$L =$	1,1	0,3	-8,5

	y_3	y_4	1
$x_2 =$	5/9	-1/9	10/9
$y_1 =$	14/9	1/9	35/9
$x_1 =$	4/9	-1/9	28/9
$y_2 =$	-23/9	10/9	35/9
$L =$	1/3	1/3	-22/3

Ответ:
 $L_{\min} = -7 \frac{1}{3}; \quad \bar{x} = \left(3 \frac{1}{9}, 1 \frac{1}{9} \right).$

КОНТРОЛЬНЫЕ ВОПРОСЫ

- Что такое задача линейного программирования? В канонической форме?
- Сформулируйте основную теорему линейного программирования.
- Метод последовательного улучшения плана. Модифицированные жордановы исключения. Поиск опорного плана. Поиск оптимального плана.
- Что собой представляет пара двойственных задач линейного программирования?
- Первая теорема двойственности.
- Вторая теорема двойственности.
- Что такое псевдоплан?
- Метод последовательного уточнения оценок.
- Вырожденность в задачах линейного программирования. Какой план называется вырожденным?

9. МЕТОДЫ РЕШЕНИЯ ТРАНСПОРТНОЙ ЗАДАЧИ

9.1. ФОРМУЛИРОВКА КЛАССИЧЕСКОЙ ТРАНСПОРТНОЙ ЗАДАЧИ

Транспортная задача линейного программирования формулируется следующим образом [9]. Необходимо минимизировать транспортные расходы

$$Q(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$$

при ограничениях

$$\left. \begin{array}{l} \sum_{i=1}^m x_{ij} = b_j, \quad j = \overline{1, n}, \\ \sum_{j=1}^n x_{ij} = a_i, \quad i = \overline{1, m}, \\ x_{ij} \geq 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}, \end{array} \right\}$$

где c_{ij} – стоимость перевозки единицы продукции из пункта i в пункт j ; x_{ij} – планируемая величина перевозок из пункта i в пункт j (план перевозок X – матрица размерности $m \times n$); b_j – потребности в продукте в пункте j ; a_i – запасы в пункте i .

Предполагается, что модель *закрытого* типа, т. е. $\sum_{j=1}^n b_j = \sum_{i=1}^m a_i$.

Если модель *открытого* типа $\left(\sum_{j=1}^n b_j \neq \sum_{i=1}^m a_i \right)$, то ее всегда можно привести к закрытому типу введением фиктивного пункта производства или фиктивного пункта потребления.

1. Если $\sum_{j=1}^n b_j < \sum_{i=1}^m a_i$, то $b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$, тогда $\sum_{j=1}^{n+1} b_j = \sum_{i=1}^m a_i$, причем $c_{i,n+1} = 0 \quad \forall i$.
2. Если $\sum_{j=1}^n b_j > \sum_{i=1}^m a_i$, то $a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$, $\sum_{j=1}^n b_j = \sum_{i=1}^{m+1} a_i$ и $c_{m+1,j} = 0 \quad \forall j$.

Транспортная задача представляет собой задачу линейного программирования, и, естественно, ее можно решить с использованием метода последовательного улучшения плана или метода последовательного уточнения оценок. В этом случае основная трудность бывает связана с числом переменных задачи ($m \times n$) и числом ограничений ($m + n$). Поэтому специальные алгоритмы оказываются более эффективными. К таким алгоритмам относятся *метод потенциалов* и *венгерский метод*.

Алгоритм метода потенциалов, его называют еще модифицированным распределительным алгоритмом, начинает работу с некоторого опорного плана транспортной задачи (допустимого плана перевозок). Для построения опорного плана обычно используется один из двух методов: *метод северо-западного угла* или *метод минимального элемента*.

9.2. МЕТОД СЕВЕРО-ЗАПАДНОГО УГЛА

Он позволяет найти некоторый допустимый план перевозок. Составим транспортную таблицу некоторой задачи.

a_i					
	30	80	20	30	90
120	2 30	4 80	2 10	3	8
30	3	5	6 10	6 20	2
40	6	8	7 10	4 10	5 30
60	3	4	2	1 	4 60

В данном случае имеем задачу закрытого типа, так как

$$\sum_{i=1}^4 a_i = 250 = \sum_{j=1}^5 b_j.$$

При построении плана следует учитывать, что сумма перевозок в столбце должна оказаться равной потребностям в данном пункте, а сумма перевозок в строке – запасу в пункте, соответствующем данной строке.

Заполнение начинается с верхнего левого угла таблицы. Величина перевозки устанавливается равной минимальной: из величины остатка запасов в пункте i или величины еще неудовлетворенного спроса в пункте j . Далее:

- если ресурс в данной строке исчерпан, то переходим к перевозке в следующей строке текущего столбца (на одну строку вниз);
- если потребности для данного пункта (столбца) удовлетворены, то переходим к следующей перевозке текущей строки в следующем столбце.

Затраты на перевозку по построенному плану равны

$$Q = 30 \times 2 + 4 \times 80 + 2 \times 10 + 6 \times 10 + 6 \times 20 + 4 \times 10 + 5 \times 30 + 4 \times 60 = 1010.$$

Естественно, что найденный план далек от оптимального.

9.3. МЕТОД МИНИМАЛЬНОГО ЭЛЕМЕНТА

В таблице отыскиваем $\min c_{ij}$ и в первую очередь заполняем соответствующую клетку: $x_{ij} = \min a_i, b_j$. Затем вычеркиваем остаток соответствующей строки, если $a_i < b_j$, или столбца, если $a_i > b_j$, и корректируем остатки запасов и неудовлетворенного спроса. В оставшихся клетках таблицы снова отыскиваем минимальную стоимость перевозки и заполняем соответствующую клетку и т. д.

a_i	b_j				
	30	80	20	30	90
120	2 30	4 80	2	3	8 10
30	3	5	6	6	2 30
40	6	8	7	4	5 40
60	3	4	20	30	10 4

Затраты на перевозку по построенному плану равны

$$Q = 30 \times 2 + 4 \times 80 + 8 \times 10 + 2 \times 30 + 5 \times 40 + 2 \times 20 + 1 \times 30 + 4 \times 10 = 830.$$

Этот план лучше, но утверждать, что он оптимален, нельзя.

Определение 1. Набором называется произвольная совокупность перевозок транспортной таблицы.

Определение 2. Цепью называют такие наборы, когда каждая пара соседних клеток в цепи расположены либо в одном столбце, либо в одной строке.

Определение 3. Циклом называется цепь, крайние элементы которой находятся либо в одной строке, либо в одном столбце.

9.4. ТЕОРЕМА, ЛЕЖАЩАЯ В ОСНОВЕ МЕТОДА ПОТЕНЦИАЛОВ

Метод потенциалов позволяет находить оптимальный план перевозок транспортной таблицы. В его основе лежит следующая теорема.

Теорема [9]. Для того чтобы некоторый план $X = [x_{ij}]_{m \times n}$ транспортной задачи был оптимальным, необходимо и достаточно, чтобы ему соответствовала такая система $m + n$ чисел $u_1, u_2, \dots, u_m; v_1, v_2, \dots, v_n$, для которой выполняются условия:

$$v_j - u_i \leq c_{ij}, \quad i = \overline{1, m}, \quad j = \overline{1, n}, \quad (1)$$

$$v_j - u_i = c_{ij}, \quad \forall x_{ij} > 0. \quad (2)$$

Величины u_i и v_j называются потенциалами соответствующих пунктов отправления и пунктов назначения. Условия (1) – (2) называются условиями потенциальности.

План X будем называть потенциальным, если для него существует система u_i и v_j , удовлетворяющая (1) – (2). Тогда теорема коротко формулируется следующим образом.

Теорема. Для оптимальности плана транспортной задачи необходимо и достаточно, чтобы он был потенциален.

Доказательство. Достаточность. Пусть план X потенциален, так что существует система u_i и v_j , удовлетворяющая (1) – (2). Тогда для любого допустимого плана $X' = [x'_{ij}]_{m \times n}$

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x'_{ij} &\geq \sum_{i=1}^m \sum_{j=1}^n v_j - u_i \quad x'_{ij} = \sum_{j=1}^n v_j \sum_{i=1}^m x'_{ij} - \sum_{i=1}^m u_i \sum_{j=1}^n x'_{ij} = \\ &= \sum_{j=1}^n v_j b_j - \sum_{i=1}^m u_i a_i = \text{для потенциального плана} \\ &= \sum_{j=1}^n v_j \sum_{i=1}^m x_{ij} - \sum_{i=1}^m u_i \sum_{j=1}^n x_{ij} = \\ &= \sum_{j=1}^n \sum_{i=1}^m v_j x_{ij} - \sum_{i=1}^m \sum_{j=1}^n u_i x_{ij} = \sum_{j=1}^n \sum_{i=1}^m v_j - u_i \quad x_{ij} = \sum_{j=1}^n \sum_{i=1}^m c_{ij} x_{ij}, \end{aligned}$$

т. е. стоимость перевозок по любому плану X' не меньше стоимости перевозок по потенциальному плану X . Следовательно, план X оптимален.

Необходимость. Будем рассматривать транспортную задачу как задачу линейного программирования с минимизацией линейной формы

$$Q(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$$

при соответствующих ограничениях. Заполним симплексную таблицу и рассмотрим двойственную к ней задачу, что легко получить из таблицы. Прямую таблицу будем заполнять, повернув.

	$0 =$ $-u_1$	\dots	$0 =$ $-u_i$	\dots	$0 =$ $-u_m$	$0 =$ $-v_1$	\dots	$0 =$ $-v_j$	\dots	$0 =$ $-v_n$	$Q =$ 1
$x_{11} y_{11} =$	-1	\dots	0	\dots	0	1	\dots	0	\dots	0	c_{11}
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
$x_{1n} y_{1n} =$	-1	\dots	0	\dots	0	0	\dots	0	\dots	1	c_{1n}
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
$x_{i1} y_{i1} =$	0	\dots	-1	\dots	0	1	\dots	0	\dots	0	c_{i1}
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
$x_{ij} y_{ij} =$	0	\dots	-1	\dots	0	0	\dots	1	\dots	0	c_{ij}
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
$x_{in} y_{in} =$	0	\dots	-1	\dots	0	0	\dots	0	\dots	1	c_{in}
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
$x_{m1} y_{m1} =$	0	\dots	0	\dots	-1	1	\dots	0	\dots	0	C_{m1}
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
$x_{mn} y_{mn} =$	0	\dots	0	\dots	-1	0	\dots	0	\dots	1	C_{mn}
$1 \quad w =$	a_1	\dots	a_i	\dots	a_n	b_1	\dots	b_j	\dots	b_n	0

Примечание. Прямую задачу заполняем, как для решения задачи методом последовательного уточнения оценок, с обыкновенными жордановыми исключениями, двойственную – как для решения симплекс-методом).

Получаем, что двойственная задача имеет вид

$$w = \sum_{j=1}^n b_j v_j - \sum_{i=1}^m a_i u_i \rightarrow \max$$

при ограничениях

$$y_{ij} = u_i - v_j + c_{ij} \geq 0, \quad i = \overline{1, m}, \quad j = \overline{1, n},$$

$$\text{т. е. } v_j - u_i \leq c_{ij}, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Пусть $X = [x_{ij}]_{m \times n}$ – оптимальное решение транспортной задачи. Тогда на основании первой теоремы двойственности двойственная задача имеет оптимальное решение

$$u_1^*, \dots, u_m^*; v_1^*, \dots, v_n^*.$$

Убедимся, что эти числа являются потенциалами соответствующих пунктов транспортной задачи. Действительно, все u_i^*, v_j^* как опорное решение двойственной задачи удовлетворяют неравенствам (1).

Если $x_{ij}^* > 0$, то по второй теореме двойственности соответствующее ограничение двойственной задачи

$$y_{ij}^* = u_i^* - v_j^* + c_{ij} \geq 0$$

обращается в строгое равенство

$$v_j^* - u_i^* = c_{ij}.$$

Теорема доказана.

9.5. АЛГОРИТМ МЕТОДА ПОТЕНЦИАЛОВ

Алгоритм метода потенциалов состоит из предварительного этапа и повторяющегося основного этапа [9].

Предварительный этап

1. Каким-либо способом ищется допустимый план X (методом северо-западного угла или минимального элемента).
2. Для полученного плана строится система $m + n$ чисел $u_1, \dots, u_m, v_1, \dots, v_n$, таких, что $v_j - u_i = c_{ij}, \forall x_{ij} > 0$.
3. Построенная система u_i и v_j исследуется на потенциальность, т. е. план X исследуется на оптимальность. Для этого проверяется $v_j - u_i \leq c_{ij}, \forall x_{ij} = 0$.

Если система непотенциальная, то переходят к основному этапу (так как план не оптimalен), иначе оптимальный план найден.

Основной этап

1. Улучшаем план, т. е. от плана X переходим к плану X' такому, что $Q(X) \geq Q(X')$.

2. Для плана X' строим новую систему $u'_i, v'_j, i = \overline{1, m}, j = \overline{1, n}$, такую, что $v'_j - u'_j = c_{ij}, \forall x_{ij} > 0$.

3. Исследуем систему u'_i, v'_j на потенциальность. Если система не-потенциальная, то переходим на п. 1. Иначе – найден оптимальный план.

Найдем оптимальное решение задачи методом потенциалов, взяв в качестве опорного план, построенный методом северо-западного угла (1-й шаг предварительного этапа).

u_i						2
	v_1	:	:	:	:	
u_1	2 30	4 80	2 10	3	8	
u_2	3	5	6 10	- 20	+ 2	
u_3	6	8	7	+ 4 10	- 5 30	
u_4	3	4	2	1	4 60	

. Строим систему потенциалов:

$$v_1 - u_1 = 2, \quad v_2 - u_1 = 4, \quad v_3 - u_1 = 2,$$

$$v_3 - u_2 = 6, \quad v_4 - u_2 = 6, \quad v_4 - u_3 = 4,$$

$$v_5 - u_3 = 5, \quad v_5 - u_4 = 4.$$

Число неизвестных больше числа уравнений, поэтому можем взять, например, $u_1 = 0$ и найти значения остальных потенциалов, $u_2 = -4$, $u_3 = -2$, $u_4 = -1$, $v_1 = 2$, $v_2 = 4$, $v_3 = 2$, $v_4 = 2$, $v_5 = 3$.

3. Проверяем систему на потенциальность:

$$v_1 - u_2 = 6 \not\leq 3, \quad v_1 - u_3 = 4 \leq 6, \quad v_1 - u_4 = 3 \leq 3,$$

$$v_2 - u_2 = 8 \not\leq 5, \quad v_2 - u_3 = 6 \leq 8, \quad v_2 - u_4 = 5 \not\leq 4,$$

$$v_3 - u_3 = 4 \leq 7, \quad v_3 - u_4 = 3 \not\leq 2, \quad v_4 - u_1 = 2 \leq 3,$$

$$v_4 - u_4 = 3 \not\leq 1, \quad v_5 - u_1 = 3 \leq 8, \quad v_5 - u_2 = 7 \not\leq 2.$$

Система непотенциальна. Переходим к общему этапу.

1. Выбираем клетку, для которой неравенство вида $v_j - u_i \leq c_{ij}$ нарушается в наибольшей степени, т. е. находится число

$$\alpha_{i_0 j_0} = \max_{i,j} \alpha_{ij} = v_j - u_i - c_{ij} > 0$$

среди тех клеток, для которых условие (1) не выполняется: $\alpha_{i_0 j_0} = \alpha_{25} = 5$.

Начиная с клетки $i_0 j_0$, в направлении (для определенности) против часовой стрелки строится цепь из заполненных клеток таблицы (цикл). Совершая обход по цепи, помечаем клетки, начиная с $i_0 j_0$, попеременно знаками «+» и «-». Клетки со знаками «+» образуют положительную полуцепь, а со знаками «-» – отрицательную полуцепь. В клетках отрицательной полуцепи ищем минимальную перевозку

$$\theta = \min x_{ij}^- .$$

Теперь улучшаем план следующим образом: перевозки отрицательной полуцепи уменьшаем на величину θ , а перевозки положительной полуцепи увеличиваем на θ . Новые перевозки

$$x'_{ij} = \begin{cases} x_{ij}^- - \theta, \\ x_{ij}^+ + \theta, \\ x_{ij}. \end{cases}$$

В нашем примере $\theta = \min x_{ij}^- = 20$.

1. Новому плану соответствует таблица

u_i						
	v_1	2				
u_1	30	2	80	4	2 10	3 8
u_2		3		5	- 10	6 0 20
u_3		6		8		4 30 5 10
u_4		3		4	+	2 1 4 60

Затраты на перевозку по построенному плану равны

$$Q = 30 \times 2 + 4 \times 80 + 2 \times 10 + 6 \times 10 + 4 \times 30 + 2 \times 20 + 5 \times 10 + 4 \times 60 = 910.$$

2. Строим систему потенциалов:

$$v_1 - u_1 = 2, \quad v_2 - u_1 = 4, \quad v_3 - u_1 = 2,$$

$$v_3 - u_2 = 6, \quad v_5 - u_2 = 2, \quad v_4 - u_3 = 4,$$

$$v_5 - u_3 = 5, \quad v_5 - u_4 = 4.$$

Полагаем $u_1 = 0$ и находим значения остальных потенциалов:

$$u_2 = -4, \quad u_3 = -7, \quad u_4 = -6, \quad v_1 = 2, \quad v_2 = 4, \quad v_3 = 2, \quad v_4 = -3, \quad v_5 = -2.$$

3. Проверяем систему на потенциальность:

$$v_1 - u_2 = 6 \not\leq 3, \quad v_1 - u_3 = 9 \not\leq 6, \quad v_1 - u_4 = 8 \not\leq 3,$$

$$v_2 - u_2 = 8 \not\leq 5, \quad v_2 - u_3 = 11 \not\leq 8, \quad v_2 - u_4 = 10 \not\leq 4,$$

$$v_3 - u_3 = 9 \not\leq 7, \quad v_3 - u_4 = 8 \not\leq 2, \quad v_4 - u_1 = -3 \leq 3,$$

$$v_4 - u_4 = 3 \not\leq 1, \quad v_5 - u_1 = -2 \leq 8, \quad v_4 - u_2 = 1 \leq 6.$$

Система непотенциальна.

1. Находим $\alpha_{i_0 j_0} = \alpha_{43} = 6$, строим цикл, $\theta = \min x_{ij}^- = 10$. Улучшаем план. Новому плану соответствует таблица

u_i						
	v_1	v_2	v_3	v_4	v_5	
u_1	2 30	4 80	2 10	3 0	6 7	3 0
u_2	3 6	5 8	6 7	0 + 4	6 + 1	2 5
u_3	6 3	8 4	7 2	- 30	+ 10	5 4
u_4	3 10	4 2	7 + 1	- 30	+ 10	5 4

Затраты на перевозку по построенному плану равны

$$Q = 30 \times 2 + 4 \times 80 + 2 \times 10 + 2 \times 10 + 4 \times 30 + 2 \times 30 + 5 \times 10 + 4 \times 50 = 850.$$

2. Строим систему потенциалов:

$$v_1 - u_1 = 2, \quad v_2 - u_1 = 4, \quad v_3 - u_1 = 2,$$

$$v_3 - u_4 = 2, \quad v_5 - u_2 = 2, \quad v_4 - u_3 = 4,$$

$$v_5 - u_3 = 5, \quad v_5 - u_4 = 4.$$

Полагаем $u_1 = 0$ и находим значения остальных потенциалов: $u_2 = 2$, $u_3 = -1$, $u_4 = 0$, $v_1 = 2$, $v_2 = 4$, $v_3 = 2$, $v_4 = 3$, $v_5 = 4$.

3. Проверяем систему на потенциальность:

$$v_1 - u_2 = 0 \leq 3, \quad v_1 - u_3 = 3 \leq 6, \quad v_1 - u_4 = 2 \leq 3,$$

$$v_2 - u_2 = 2 \leq 5, \quad v_2 - u_3 = 5 \leq 8, \quad v_2 - u_4 = 4 \leq 4,$$

$$v_3 - u_3 = 3 \leq 7, \quad v_3 - u_2 = 0 \leq 6, \quad v_4 - u_1 = 3 \leq 3,$$

$$v_4 - u_4 = 3 \not\leq 1, \quad v_5 - u_1 = 4 \leq 8, \quad v_4 - u_2 = 1 \leq 6.$$

Система непотенциальна.

1. Находим $\alpha_{i_0 j_0} = \alpha_{44} = 2$, строим цикл, $\theta = \min x_{ij}^- = 30$. Улучшаем план. Новому плану соответствует таблица

u_i	v_j				
	v_1	v_2	v_3	v_4	v_5
u_1	2 30	4 80	2 10	3	8
u_2	3	5	6	6	2 30
u_3	6	8	7	4 0	5 40
u_4	3	4	2 10	1 30	4 20

Затраты на перевозку по построенному плану равны

$$Q = 30 \times 2 + 4 \times 80 + 2 \times 10 + 2 \times 10 + 1 \times 30 + 2 \times 30 + 5 \times 40 + 4 \times 20 = 790.$$

2. Строим систему потенциалов:

$$v_1 - u_1 = 2, \quad v_2 - u_1 = 4, \quad v_3 - u_1 = 2,$$

$$v_3 - u_4 = 2, \quad v_5 - u_2 = 2, \quad v_4 - u_4 = 1,$$

$$v_5 - u_3 = 5, \quad v_5 - u_4 = 4.$$

Полагаем $u_1 = 0$ и находим значения остальных потенциалов: $u_2 = 2$, $u_3 = -1$, $u_4 = 0$, $v_1 = 2$, $v_2 = 4$, $v_3 = 2$, $v_4 = 1$, $v_5 = 4$.

3. Проверяем систему на потенциальность:

$$\begin{array}{lll} v_1 - u_2 = 0 \leq 3, & v_1 - u_3 = 3 \leq 6, & v_1 - u_4 = 2 \leq 3, \\ v_2 - u_2 = 2 \leq 5, & v_2 - u_3 = 5 \leq 8, & v_2 - u_4 = 4 \leq 4, \\ v_3 - u_3 = 3 \leq 7, & v_3 - u_2 = 0 \leq 6, & v_4 - u_1 = 1 \leq 3, \\ v_4 - u_4 = 1 \leq 1, & v_5 - u_1 = 4 \leq 8, & v_4 - u_2 = -1 \leq 6. \end{array}$$

Система потенциальна, следовательно, план оптимален и окончательные затраты $Q_{\min} = 790$.

9.6. О ВЫРОЖДЕННОСТИ ТРАНСПОРТНОЙ ЗАДАЧИ

Определение 4. Допустимый опорный план транспортной задачи называется невырожденным, если число заполненных клеток транспортной таблицы, т. е. число положительных перевозок $x_{ij} > 0$, равно $m + n - 1$, где m – число пунктов отправления, n – число пунктов назначения.

Определение 5. Если допустимый опорный план содержит менее $m + n - 1$ элементов $x_{ij} > 0$, то он называется вырожденным, а транспортная задача – вырожденной транспортной задачей.

Следующая теорема позволяет определить вырожденность задачи до ее решения.

Теорема. Для невырожденной транспортной задачи необходимо и достаточно отсутствие такой неполной группы пунктов производства, суммарный объем производства которой точно совпадает с суммарными потребностями некоторой группы пунктов потребления.

Другими словами, это условие означает, что для любых двух систем индексов i_1, i_2, \dots, i_t , j_1, j_2, \dots, j_s , где $t + s < n + m$, имеет место неравенство

$$\sum_{k=1}^t a_{i_k} \neq \sum_{k=1}^s b_{j_k}. \quad (\text{Доказательство не сложно, от противного.})$$

Для решения транспортной задачи методом потенциалов строится система потенциалов $v_j - u_i = c_{ij}$, $\forall x_{ij} > 0$. Если опорное решение невырожденно, то число уравнений равно $m + n - 1$, а число неизвестных на 1 больше числа уравнений. При вырожденном опорном решении

количество положительных перевозок меньше $m + n - 1$. По аналогии симплекс-методом в невырожденном решении $x_{ij} > 0$ представляют собой базисные переменные, а $x_{ij} = 0$ – небазисные. Если опорное решение вырожденно, то часть базисных переменных принимает нулевые значения.

Пусть первое опорное решение, найденное методом северо-западного угла или методом минимального элемента, является вырожденным. Тогда, чтобы решать задачу методом потенциалов, необходимо выбрать в качестве базисных переменных некоторые перевозки $x_{ij} = 0$ и для них также составить уравнения $v_j - u_i = c_{ij}$ по условию (2) теоремы. Какие перевозки вида $x_{ij} = 0$ включать в базисные? Выбираются такие клетки таблицы с $x_{ij} = 0$, чтобы из базисных переменных нельзя было организовать ни одного цикла!

При переходе к новому улучшенному плану задачи в небазисные переменные переводится перевозка из отрицательной полуцепи, которая находится следующим образом: $\theta = \min x_{ij}^-$. В вырожденной задаче это значение может достигаться на нескольких перевозках x_{ij} отрицательной полуцепи. В этом случае на каждом шаге в небазисные переменные переводится та минимальная перевозка отрицательной полуцепи, которая связана с пунктом производства, имеющим меньший номер. Это правило уменьшает вероятность возникновения зацикливания, что само по себе достаточно редкое явление.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Сформулируйте транспортную задачу линейного программирования.
2. Что такое транспортная задача закрытого типа?
3. Методы построения опорного плана транспортной задачи линейного программирования.
4. Сформулируйте теорему, на которую опирается алгоритм метода потенциалов.
5. В чем заключается проверка оптимальности опорного плана транспортной задачи? Что такое условие потенциальности?
6. Каким образом улучшается опорный план транспортной задачи?
7. Как можно проверить, является ли транспортная задача вырожденной?
8. К чему может привести вырожденность транспортной задачи?

10. ТРАНСПОРТНАЯ ЗАДАЧА С ОГРАНИЧЕНИЯМИ

10.1. ПОСТАНОВКА ЗАДАЧИ

Транспортная задача линейного программирования с ограничениями на пропускные способности путей сообщения может быть сформулирована следующим образом [9]. Необходимо минимизировать транспортные расходы:

$$Q(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$$

при ограничениях

$$\left. \begin{array}{l} \sum_{i=1}^m x_{ij} = b_j, \quad j = \overline{1, n}, \\ \sum_{j=1}^n x_{ij} = a_i, \quad i = \overline{1, m}, \\ x_{ij} \geq 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}, \\ x_{ij} \geq d_{ij}, \quad i = \overline{1, m}, \quad j = \overline{1, n}, \end{array} \right\}$$

где c_{ij} – стоимость перевозки единицы продукции из пункта i в пункт j ; x_{ij} – планируемая величина перевозок из пункта i в пункт j (план перевозок X – матрица размерности $m \times n$); b_j – потребности в продукте в пункте j ; a_i – запасы в пункте i ; d_{ij} – ограничение на величину планируемой перевозки из пункта i в пункт j .

Алгоритм состоит из двух этапов:

- определение опорного плана;
- определение оптимального плана методом потенциалов.

10.2. МЕТОД ПОТЕНЦИАЛОВ ДЛЯ ОПРЕДЕЛЕНИЯ ОПТИМАЛЬНОГО ПЛАНА

1. Пусть у нас есть опорный план. Для перевозок вида

$$0 < x_{ij} < d_{ij}, \quad (1)$$

для определения потенциалов u_i и v_j соответствующих пунктов отправления и пунктов назначения составляется система уравнений

$$v_j - u_i = c_{ij} \quad (2)$$

и находятся потенциалы u_i и v_j .

2. Для остальных клеток транспортной таблицы вычисляются значения

$$c_{ij} - (v_j - u_i). \quad (3)$$

Если для $\forall x_{ij} = 0$

$$\alpha_{ij} = c_{ij} - (v_j - u_i) \geq 0 \quad (4)$$

и для $\forall x_{ij} = d_{ij}$

$$\beta_{ij} = c_{ij} - v_j + u_i \leq 0, \quad (4')$$

то опорный план транспортной задачи *оптимален*.

Если эти условия не выполняются, то среди отрицательных чисел α_{ij} и β_{ij} выбираем наименьшее (максимальное по модулю). Пусть это наименьшее из чисел соответствует перевозке с индексами i_0 и j_0 .

Возможно два варианта:

а) наименьшее из чисел соответствует $x_{i_0 j_0} = 0$;

б) наименьшее из чисел соответствует $x_{i_0 j_0} = d_{i_0 j_0}$.

Начиная с перевозки $x_{i_0 j_0}$, строим замкнутый цикл из базисных перевозок транспортной задачи [соответствующих (1)]. В зависимости от случаев а) и б) дальнейшие действия различаются.

а) В этом случае для улучшения плана мы должны ввести в базис перевозку $x_{i_0 j_0}$. Пометим перевозки цикла, начиная с i_0 , j_0 , поочередно знаками «+» и «-» ($x_{i_0 j_0}$ помечаем знаком «+»).

Определим для отрицательной полуцепи

$$\theta' = \min x_{ij}^- ,$$

для положительной полуцепи

$$\theta'' = \min d_{ij}^+ - x_{ij}^+ .$$

и возьмем

$$\theta'' = \min \theta', \theta'', \theta_{i_0, j_0} .$$

Для получения более экономичного плана перевозки положительной полуцепи увеличиваем на θ , а отрицательной – уменьшаем на θ .

После этого переходим к п. 1.

б) В этом случае перевозку $x_{i_0 j_0}$ помечаем знаком «-», а остальные перевозки цикла помечаем последовательно «+» и «-».

Определим для отрицательной полуцепи

$$\theta' = \min x_{ij}^- ,$$

для положительной полуцепи

$$\theta'' = \min d_{ij}^+ - x_{ij}^+ .$$

Определяем

$$\theta = \min \theta', \theta'', d_{i_0, j_0} .$$

Увеличиваем перевозки положительной полуцепи на θ , а перевозки отрицательной полуцепи уменьшаем на θ . Переходим к п. 1.

10.3. ПОСТРОЕНИЕ ОПОРНОГО ПЛАНА

Построение опорного плана состоит из двух этапов: предварительного этапа, напоминающего метод минимального элемента, и ряда этапов метода потенциалов, применяемого к *расширенной* задаче.

Предварительный этап разбивается на несколько однотипных шагов.

Первый шаг. Среди элементов c_{ij} матрицы C находим минимальный. Если этим элементом является $c_{i_1 j_1}$, то находим

$$x_{i_1 j_1} = \min \{a_{i_1}, b_{j_1}, d_{i_1 j_1}\}.$$

Возможны три случая:

$$x_{i_1 j_1} = a_{i_1}; \quad x_{i_1 j_1} = b_{j_1}; \quad x_{i_1 j_1} = d_{i_1 j_1}.$$

В первом случае все остальные перевозки строки $x_{i_1 j} = 0 \quad \forall j \neq j_1$, во втором – все остальные перевозки столбца $x_{i_1 j} = 0 \quad \forall i \neq i_1$. В третьем случае заполняется только $x_{i_1 j_1}$. Далее вычеркиваем из матрицы C либо строку, либо столбец, либо элемент $c_{i_1 j_1}$. Преобразуем величины в таблице:

$$a_i^1 = \begin{cases} a_i, & i \neq i_1, \\ a_{i_1} - x_{i_1 j_1}, & i = i_1; \end{cases} \quad b_j^1 = \begin{cases} b_j, & j \neq j_1, \\ b_{j_1} - x_{i_1 j_1}, & j = j_1. \end{cases}$$

Второй шаг состоит в проведении тех же операций применительно к невычеркнутым элементам матрицы C , не заполненным позициям матрицы X и величинам a_i^1, b_j^1 .

Шаги предварительного этапа следуют до полного заполнения матрицы X . Согласно процессу формирования матрицы X ее элементы удовлетворяют условиям

$$\left. \begin{array}{l} \sum_{i=1}^m x_{ij} \leq b_j, \quad j = \overline{1, n}, \\ \sum_{j=1}^n x_{ij} \leq a_i, \quad i = \overline{1, m}, \\ 0 \leq x_{ij} \geq d_{ij}, \quad i = \overline{1, m}, \quad j = \overline{1, n}. \end{array} \right\}$$

Положим

$$\left. \begin{aligned} x_{m+1,j} &= b_j - \sum_{i=1}^m x_{ij}, \quad j = \overline{1,n}, \\ x_{i,n+1} &= a_i - \sum_{j=1}^n x_{ij}, \quad i = \overline{1,m}, \\ \varepsilon &= \sum_{i=1}^m x_{i,n+1} = \sum_{j=1}^n x_{m+1,j}. \end{aligned} \right\}$$

Если $\varepsilon = 0$, то очевидно, что матрица X является (опорным) планом задачи, которую обозначим T_d .

Однако в общем случае $\varepsilon > 0$, и для получения искомого опорного плана задачи T_d необходимо провести еще несколько итераций методом потенциалов.

Введем расширенную задачу $T_d(M)$, которую образуем из T_d следующим образом. Присоединим к пунктам производства задачи T_d фиктивный пункт A_{m+1} с объемом производства $a_{m+1} = \varepsilon$, а к пунктам потребления пункт B_{n+1} с $b_{n+1} = \varepsilon$. Пусть стоимости перевозок $c_{i,n+1}$, $i = 1, m$ и $c_{m+1,j}$, $i = 1, n$ равны M (максимально большое число), а $c_{m+1,n+1} = 0$.

Для этой задачи легко образовать опорный план, введя перевозки из i в $n+1$ и из $m+1$ в j , равные $x_{i,n+1}$ и $x_{m+1,j}$, и взяв $x_{m+1,n+1} = 0$.

К задаче $T_d(M)$ применяем метод потенциалов. При ее решении возможны два случая.

1. После ряда итераций строится опорный план X_1 задачи $T_d(M)$, согласно которому перевозка между A_{m+1} и B_{n+1} равна ε . В этом случае множество перевозок между пунктами a_i и b_j , $i = \overline{1,m}$, $j = \overline{1,n}$, составят опорный план исходной задачи.

2. В оптимальном плане задачи $T_d(M)$, который определяется за несколько итераций метода потенциалов, перевозка между пунктами A_{m+1} и B_{n+1} меньше ε . В этом случае задача T_d не имеет ни одного плана, т. е. неразрешима.

Пример

	v_1	v_2	v_3	v_4	v_5	
γ_i	15	30	35	20	1	
	15) 14 12	35) 10 23	14) + 2 14	10) 5		\bar{M}
u_1	50					$\uparrow 1$
u_2	20	8) 11 4) 5	20) 4 20	12) 11		M
u_3	30	10) 9 3 7	32) 12 20	20) 1 20		M
u_4	1	M 0	M 0	M 1	M 0	$+ 0$

Должно быть $5 + 4 - 1 = 8$ уравнений.

$$v_1 - u_1 = 14, \quad v_2 - u_1 = 10, \quad v_1 - u_3 = 9,$$

$$v_5 - u_1 = M, \quad v_3 - u_4 = M.$$

Задача вырожденная. Необходимо ввести в базис три перевозки:

x_{13} , нет циклов $x_{13} = d_{13} = 14$;

x_{23} , нет циклов $x_{23} = d_{23} = 20$;

$x_{33} = 0$, вводить нельзя, так как образуется цикл;

x_{34} , нет циклов $x_{34} = d_{34} = 20$;

нельзя x_{21} , (так как x_{23}), x_{22} , x_{24} , x_{25} , x_{35} , x_{41} , x_{42} ...

Добавляем уравнения:

$$v_3 - u_1 = 2, \quad v_3 - u_2 = 4, \quad v_4 - u_3 = 1.$$

Вычисляем:

$$u_1 = 0, \quad u_4 = 2 - M, \quad v_3 = 2,$$

$$u_2 = -2, \quad v_1 = 14, \quad v_4 = 6,$$

$$u_3 = 5, \quad v_2 = 10, \quad v_5 = M.$$

Проверяем условия потенциальности для $x_{ij} = 0$:

$$v_4 - u_1 \neq 6 \leq 5, \quad v_1 - u_2 = 16 \not\leq 11, \quad v_2 - u_2 = 12 \not\leq 5,$$

$$v_4 - u_2 = 8 \leq 11, \quad v_5 - u_2 = M + 2 \not\leq M, \quad v_3 - u_3 = -3 \leq 12,$$

$$v_5 - u_3 = M - 5 \leq M, \quad v_1 - u_4 = M + 12 \not\leq M, \quad v_2 - u_4 = M + 8 \not\leq M,$$

$$v_4 - u_4 = M + 4 \not\leq M, \quad v_5 - u_4 = 2M - 2 \not\leq M.$$

Максимальное $-\alpha_{45} = 2M - 2$.

Проверяем для небазисных перевозок вида $x_{ij} = d_{ij}$:

$$v_2 - u_3 = 5 \not\geq 8. \text{ Следовательно, } \beta_{32} = 3.$$

$-\alpha_{45} > \beta_{32}$, поэтому помечаем x_{45} (+) и строим цикл,

$$\theta = \min \theta', \theta'', d_{i_0, j_0} = \min 0, 1, \infty = 0.$$

Перевозку x_{13} выводим из базиса, перевозку x_{45} – в базис. Перевозка x_{45} вводится в базис, но $x_{45} = 0$.

Ищем новую систему потенциалов:

$$v_1 - u_1 = 14, \quad v_2 - u_1 = 10, \quad v_1 - u_3 = 9,$$

$$v_5 - u_1 = M, \quad v_3 - u_4 = M, \quad v_3 - u_2 = 4,$$

$$v_4 - u_3 = 1, \quad v_5 - u_4 = 0,$$

$$u_1 = 0, \quad u_4 = M, \quad v_3 = 2M,$$

$$u_2 = 2M - 4, \quad v_1 = 14 \quad v_4 = 6,$$

$$u_3 = 5, \quad v_2 = 10, \quad v_5 = M.$$

Проверяем условия потенциальности для $x_{ij} = 0$:

$$v_4 - u_1 = 6 \not\leq 5, \quad v_1 - u_2 = 18 - 2M \leq 11, \quad v_2 - u_2 = 14 - 2M \leq 5,$$

$$v_4 - u_2 = 10 - 2M \leq 11, \quad v_5 - u_2 = 4 - M \leq M, \quad v_3 - u_3 = 2M - 5 \not\leq 12,$$

$$v_5 - u_3 = M - 5 \leq M, \quad v_1 - u_4 = 14 - M \leq M, \quad v_2 - u_4 = 10 - M \leq M.$$

$$-\alpha_{33} = 2M - 17.$$

Проверяем для небазисных перевозок $x_{ij} = d_{ij}$:

$$v_2 - u_3 = 5 \not\geq 8, v_3 - u_1 = 2M \geq 2.$$

Следовательно, x_{33} в базис.

$$\theta = \min \theta', \theta'', d_{i_0, j_0} = \min 1, 3, 32 = 1.$$

15 +	14	35	10	14	2	10	5	M
12	6	23	6	14			M	6
8	11	4	5	20	4	12	11	M
				20	6		0	
10-	9	7	8	32	12	20	1	M
3	6	7		+		20	6	0
	M	M			M	M	+	0
0	0	1			6	0	0	6

Есть опорный план! Ищем оптимальное решение: $m = 3$, $n = 4$,
 $m + n - 1 = 6$.

15– 13	$\frac{14}{6}$	$35 +$ 23	$\frac{10}{6}$	14 14	2	10 12	5 11
8	11	4	5	20 20	4 $\frac{6}{6}$	12 20	11 $\frac{6}{6}$
10 + 2	$\frac{9}{6}$	7 7	–	8	32 1	12 $\frac{6}{6}$	20 20 $\frac{1}{6}$

$$v_1 - u_1 = 14, \quad v_2 - u_1 = 10, \quad v_1 - u_3 = 9,$$

$$v_3 - u_3 = 12, \quad v_3 - u_2 = 4, \quad v_4 - u_3 = 1,$$

$$u_1 = 0 \quad v_1 = 14 \quad v_4 = 6$$

$$u_2 = 13 \quad v_2 = 10$$

$$u_3 = 5 \quad v_3 = 17$$

$$v_4 - u_1 = 6 \not\leq 5, \quad v_1 - u_2 = 1 \leq 11, \quad v_2 - u_2 = -3 \leq 5,$$

$$v_4 - u_2 = -7 \leq 11.$$

$$v_3 - u_1 = 17 \geq 2, \quad v_2 - u_3 = 5 \not\geq 8.$$

Максимальное $\beta_{32} = 3$, x_{32} помечаем (–) и строим цикл, $\theta = 7$. Система потенциалов та же, потенциалы те же. x_{14} в базис с (+).

15– 6	$\frac{14}{6}$	35 30	$\frac{10}{6}$	14 14	2	10 + 12	5 11
8	11	4	5	20 20	4 $\frac{6}{6}$	12 20	11 $\frac{6}{6}$
10 + 9	$\frac{9}{6}$	7	8	32 1	12 $\frac{6}{6}$	20 20 $\frac{1}{6}$	– $\frac{6}{6}$

$\theta = \min \theta', \theta'', d_{i_0, j_0} = \min 13, 10 - 9, 10 = 1$, т. е. x_{31} из базиса.

15 5	$\frac{14}{6}$	35 30	$\frac{10}{6}$	14 14	2	10 1	5 $\frac{6}{6}$
8 11		4	5	20 20	4 $\frac{6}{6}$	12 19	11 $\frac{6}{6}$
10 10	9	7	8	32 1	12 $\frac{6}{6}$	20 19	1 $\frac{6}{6}$

Система уравнений:

$$v_1 - u_1 = 14, \quad v_2 - u_1 = 10, \quad v_4 - u_1 = 5,$$

$$v_3 - u_3 = 12, \quad v_3 - u_2 = 4, \quad v_4 - u_3 = 1,$$

$$u_1 = 0, \quad v_1 = 14, \quad v_4 = 5,$$

$$u_2 = 12, \quad v_2 = 10,$$

$$u_3 = 4, \quad v_3 = 16.$$

Проверяем условия потенциальности для $x_{ij} = 0$:

$$v_1 - u_2 = 2 \leq 11, \quad v_2 - u_2 = -2 \leq 5,$$

$$v_4 - u_2 = -7 \leq 11, \quad v_2 - u_3 = 6 \leq 8$$

для $x_{ij} = d_{ij}$:

$$v_3 - u_1 = 16 \geq 2, \quad v_1 - u_3 = 10 \geq 9.$$

Все условия выполняются, план оптимален.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Сформулируйте транспортную задачу линейного программирования с ограничениями.
2. К чему приводит наличие ограничений на пропускные способности?
3. Метод потенциалов для определения оптимального плана транспортной задачи с ограничениями.
4. Метод потенциалов для определения опорного плана транспортной задачи с ограничениями.
5. К чему может привести вырожденность в такой транспортной задаче?

11. ТРАНСПОРТНАЯ ЗАДАЧА ПО КРИТЕРИЮ ВРЕМЕНИ

В такой транспортной задаче решающую роль играет не стоимость перевозок, а время, которое затрачивается на доставку груза. Оптимальным планом считается план, который минимизирует время перевозок [9]. Подобные задачи возникают при перевозках скоропортящихся продуктов, в военном деле, где зачастую стоимость перевозок играет второстепенную роль. Как и в предыдущей задаче, имеется m пунктов отправления с запасами однородного продукта a_i , n пунктов назначения с потребностями b_j .

Задача закрытого типа, т. е. $\sum_{j=1}^n b_j = \sum_{i=1}^m a_i$.

Задана матрица $T = [t_{ij}]_{m \times n}$, где t_{ij} – время, необходимое для перевозки груза из пункта i в пункт j .

Необходимо выбрать среди допустимых такой план $X = [x_{ij}]_{m,n}$, что

$$\left. \begin{array}{l} \sum_{i=1}^m x_{ij} = b_j, \forall j \in \overline{1, n} \\ \sum_{j=1}^n x_{ij} = a_i, \forall i \in \overline{1, m} \end{array} \right\}$$

и грузы будут доставляться по этому плану за минимальное время T_{\min} .

Каждому допустимому плану $X = [x_{ij}]_{m,n}$ соответствует некоторый набор $t_{ij}|_X$, состоящий из элементов матрицы $T = [t_{ij}]_{m \times n}$, соответствующих положительным компонентам x_{ij} плана X , т. е. t_{ij} включается в набор, если производится перевозка из пункта i в пункт j .

Время t_X , необходимое для выполнения плана X , определяется следующим образом:

$$t_X = \max\{t_{ij}\}_X.$$

Тогда время, необходимое для реализации оптимального плана X^* :

$$t_{X^*} = \min_X t_X = \min_X (\max\{t_{ij}\}_X).$$

Алгоритм отыскания оптимального решения. Данный алгоритм состоит из двух этапов.

1. Предварительный шаг. Строим допустимый план по методу северо-западного угла или минимального элемента X_0 .
2. Общий шаг. Просматриваем все t_{ij} , соответствующие положительным x_{ij} , и выбираем из них наибольшее

$$t_{ij}' = \max_{x_{ij} > 0} \{t_{ij}\}$$

и вычеркиваем все клетки, для которых $t_{ij} \geq t_{ij}'$, $\forall x_{ij} = 0$.

Далее исправляем план X_0 , для чего стремимся обратить в 0 перевозку x_{ij}' , соответствующую $t_{ij}' = \max_{x_{ij} > 0} \{t_{ij}\}$ (в той же клетке). Если это удается, то, естественно, уменьшается время, необходимое на реализацию нового допустимого плана X_1 . Для построения плана X_1 строится цикл, как и в методе потенциалов.

В качестве первой клетки отрицательной полуцепи берем клетку с t_{ij}' , в качестве остальных – клетки с $x_{ij} > 0$, клетками положительной полуцепи считаем клетки с $t_{ij} < t_{ij}'$.

Затем перемещаем минимальный элемент θ отрицательной полуцепи в положительную. Если удается обратить x_{ij}' в 0, то реализация нового плана требует меньшего времени.

Общий шаг продолжаем повторять до тех пор, пока не станет невозможным обращение в 0 всей перевозки x_{ij}' из клетки с максимальным временем t_{ij}' .

Пример 1

a_i					b_j
	7	13	9	11	
10	7	8 → 3 → 10	8		2
18		12 ↓ 10 → 7 → 8	3		1
12		5 → 8 → 1 ↓ 11	1	6	

a_i					b_j
	7	13	9	11	
10	7	8 × 10 × 8			2
18	×	12 13 7 5 3			1
12	5	4	1	6	8

a_i	b_j			
	7	13	9	11
10	8 ×	10 ×	8 ×	2 10
18	12 ×	13 7	3 5	1
12	5 7	4	1	6 1

a_i	b_j			
	7	13	9	11
10	8 ×	10 ×	8 ×	2 10
18	12 ×	8 7	3 9	1
12	5 7	4 5	1	6

$$t_{\min} = 7$$

Пример 2

a_i	b_j				
	30	80	20	30	90
120	6	8	2	3	5
30	8	5	6	6	2
40	2	4	7	4	8
60	3	4	2	1	4

a_i	b_j				
	30	80	20	30	90
120	6 ×	8 ×	2 20	3 30	5 70
30	8 30	5	6 ×	6 ×	2
40	2	4 40	7 ×	4	8 ×
60	3	4 40	2	1	4 20

$$t_{\min} = 5$$

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как построить допустимый план транспортной задачи по критерию времени?
2. Как ищется оптимальный план?

12. ЗАДАЧА О МАКСИМАЛЬНОМ ПОТОКЕ В ТРАНСПОРТНОЙ СЕТИ

12.1. ПОСТАНОВКА ЗАДАЧИ

Определение 1. Транспортной сетью называется конечный граф G , состоящий из $(n+1)$ вершин P_0, P_1, \dots, P_n и из дуг (P_i, P_j) , соединяющих некоторые пары этих вершин, причем каждой дуге поставлено в соответствие число $c_{ij} \geq 0$, называемое пропускной способностью дуги (P_i, P_j) .

Вершина P_0 называется входом сети, а P_n – выходом транспортной сети.

Будем считать, что граф симметрический, т. е. если в него входит дуга (P_i, P_j) , то входит и дуга (P_j, P_i) .

c_{ij} – определяет количество вещества (машин и т. п.), которое может протекать по дуге в единицу времени.

Например (рис. 12.1): $c_{ij} = 0$, если P_i и P_j не соединены дугой.

По путям μ $(P_0, P_{i_1}, P_{i_2}, \dots, P_{i_k}, P_n)$, составленным из дуг сети $(P_0, P_{i_1}), (P_{i_1}, P_{i_2}), \dots, (P_{i_k}, P_n)$, направляется транспорт из P_0 в P_n .

Потоком x_{ij} по дуге (P_i, P_j) ($i, j = 0, \dots, n, i \neq j$) называется количество вещества, проходящее через эту дугу в единицу времени.

Потоком по сети или просто потоком будем называть совокупность потоков $\{x_{ij}\}$ по всем дугам сети.

Потоки должны удовлетворять следующим ограничениям:

$$0 \leq x_{ij} \leq c_{ij} \quad (i, j = 0, \dots, n, i \neq j), \quad (1)$$

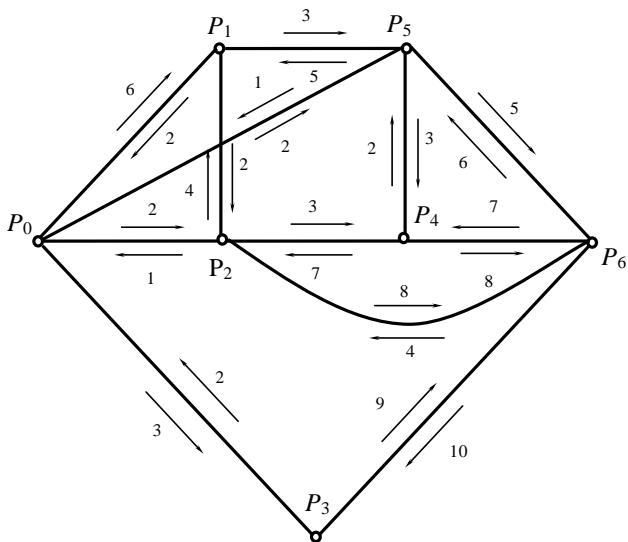


Рис. 12.1. Пример транспортной сети

$$\sum_{i=0}^{n-1} x_{ik} = \sum_{j=1}^n x_{kj}, \quad (k = 1, \dots, n-1). \quad (2)$$

Равенство (2) означает, что количество вещества, притекающее в вершину сети, равно количеству вещества, вытекающего из него (кроме P_0 и P_n).

Поток, удовлетворяющий ограничениям (1) и (2), будем называть допустимым.

Из (2) видно, что общее количество вещества $\sum_{j=1}^n x_{0j}$, вытекающего из P_0 , совпадает с общим количеством вещества $\sum_{i=0}^{n-1} x_{in}$, притекающего в P_n , т. е.

$$\sum_{i=0}^{n-1} x_{in} = \sum_{j=1}^n x_{0j} = Q. \quad (3)$$

Линейная форма Q называется потоком по сети.

Задача о максимальном потоке в транспортной сети заключается в отыскании такого решения x_{ij}^* ($i, j = \overline{0, n}$) системы (1) – (2), т. е. такого допустимого потока, который максимизирует Q . Это решение x_{ij}^* называется максимальным потоком сети.

Рассмотрим такой простейший пример.

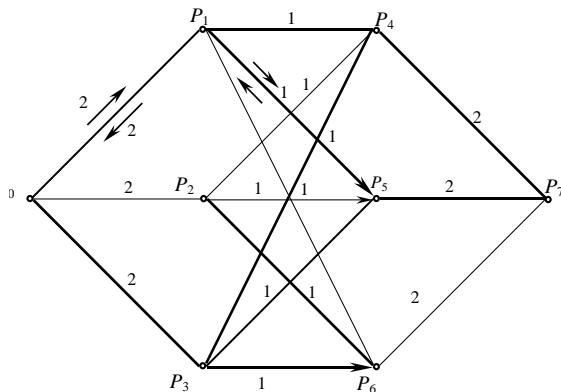


Рис. 12.2. Пример простейшей транспортной сети

Найти максимальный поток из P_0 в P_7 . Толстые дуги насыщены, т. е. $x_{ij} = c_{ij}$, полный поток $\sum_{j=1}^n x_{0j} = 5$, а максимальный поток равен 6.

Разобьем множество всех вершин G на два подмножества U и V так, что $P_0 \in U$ и $P_7 \in V$. Сечением (U, V) сети G назовем совокупность всех дуг (P_i, P_j) , концы которых принадлежат разным подмножествам.

Каждому сечению поставим в соответствие неотрицательное число $C(U, V)$ – пропускную способность сечения, равную сумме c_{ij} всех дуг сечения, начинающихся в U и кончающихся в V , т. е.

$$C(U, V) = \sum_{\substack{P_i \in U \\ P_j \in V}} c_{ij}.$$

Любой путь из P_0 в P_n непременно содержит хоть одну дугу сечения (U, V) , которая начинается в U и заканчивается в V . Ясно, что пропускная способность пути не превышает пропускной способности каждой его дуги. Поэтому величина любого потока из P_0 в P_n , которая является суммарной величиной пропускной способности всех путей из P_0 в P_n , не может превысить пропускной способности любого сечения (U, V) , т. е. всегда $Q \leq C(U, V)$.

Теорема Форда–Фалкерсона утверждает следующее.

Теорема [9]. Для заданной транспортной сети наибольшая величина потока равна наименьшей пропускной способности сечения, т. е.

$$\max Q \leq \min_{(U,V)} C(U, V) = C(U^*, V^*)$$

по всем возможным сечениям.

Таким образом, если удастся построить такой поток x_{ij}^* , что $Q^* = C(U^*, V^*)$, то этот поток будет максимальным, а сечение $C(U^*, V^*)$ – обладать минимальной пропускной способностью.

Естественно, что каждую транспортную сеть стремится использовать оптимально, т. е. организовать перевозки по транспортным путям таким образом, чтобы поток перевозимых грузов был максимальным.

12.2. АЛГОРИТМ ПОСТРОЕНИЯ МАКСИМАЛЬНОГО ПОТОКА В ТРАНСПОРТНОЙ СЕТИ

Предварительный шаг. Условия (1) записываются в виде следующей таблицы. Если $c_{ij} > 0$, $c_{ji} = 0$, то в клетке ji ставим 0. Если же $c_{ij} = c_{ji} = 0$, то клетки ij и ji не заполняем.

Начальная таблица будет выглядеть следующим образом:

	P_0	...	P_i	...	P_j	...	P_n
P_0			c_{0i}		c_{0j}		c_{0n}
...							
P_i	$c_{i0} = 0$				c_{ij}		c_{in}
...							
P_j	$c_{j0} = 0$		c_{ji}				c_{jn}
...							
P_n	$c_{n0} = 0$		$c_{ni} = 0$		$c_{nj} = 0$		

(4)

Будем рассматривать алгоритм на примере следующей транспортной сети (рис. 12.3)

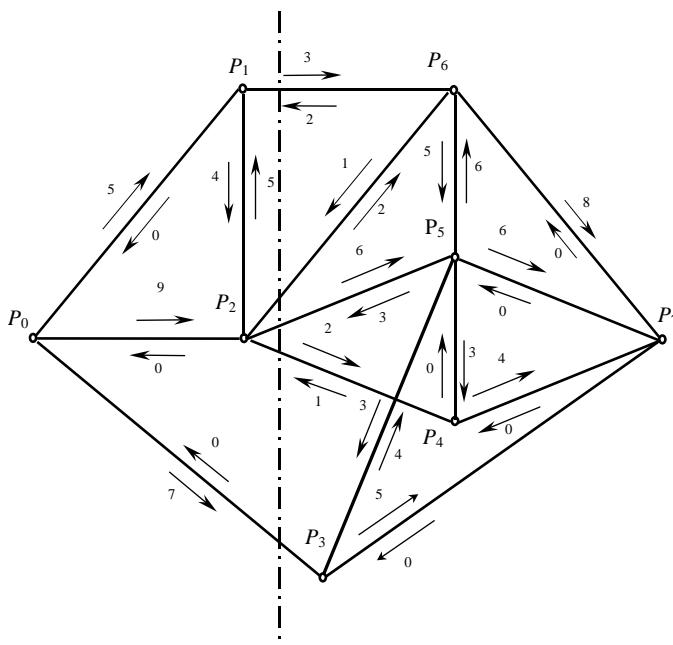


Рис. 12.3. Транспортная сеть

Построенная начальная таблица примет вид

	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_0		5	9	7				
P_1	0		4				3	
P_2	0	5			2	6	2	
P_3	0					4		5
P_4			1			0		4
P_5			3	3	3		6	6
P_6		2	1			5		8
P_7				0	0	0	0	

Общий шаг состоит из трех действий.

1. *Отыскивается по таблице новый путь из P_0 в P_n .*

Сначала отмечаем P_0 -й столбец *. Затем отыскиваем в строке P_0 все положительные c_{0i} и содержащие их столбцы отмечаем сверху числом 0 (номером вершины P_0), т. е. выделили все дуги (P_0, P_i) , которые могут быть первыми дугами различных путей из P_0 в P_n .

Просматриваем затем строки, имеющие те же номера, что и отмеченные столбцы.

В каждой такой строке (например, P_i) отыскиваем все неотрицательные c_{ij} , расположенные в неотмеченных столбцах, и отмечаем эти столбцы номером рассматриваемой строки (например i).

Этим самым будут выделены дуги (P_i, P_j) с положительной пропускной способностью, которые могут служить вторыми дугами различных путей из P_0 в P_n , т. е. уже выделены (P_0, P_i) и (P_i, P_j) .

Продолжаем аналогичный просмотр строк с номерами отмеченных столбцов. Процесс оканчивается, если:

а) отмечен P_n -й столбец, т. е. удалось выделить дугу (P_k, P_n) с $c_{kn} > 0$, которая служит последней дугой некоторого пути из P_0 в P_n ;

б) просмотрены все строки и нельзя отметить новых столбцов (т. е. в неотмеченных столбцах нет $c_{ij} > 0$), это означает отсутствие пути из P_0 в P_n , все дуги которого обладают положительными пропускными способностями (алгоритм закончен).

В случае (а) искомый новый путь из P_0 в P_n , начиная от P_n , отыскиваем следующим образом. Пусть столбец P_n был отмечен номером k , т. е. предшествующая вершина в пути, соединяющем P_0 с P_n , — P_k . (Столбец P_n был отмечен при просмотре строки P_k .) Число $c_{kn} > 0$ помечаем знаком «—» c_{kn}^- . Число c_{nk} , расположенное симметрично к диагонали, помечаем знаком «+» c_{nk}^+ . Так как рассматривалась P_k -я строка, значит, перед этим был отмечен столбец P_k номером например, l . По столбцу P_k двигаемся вверх до P_l -й строки. c_{lk} отмечаем «—» c_{lk}^- , а c_{kl} знаком «+». Этот процесс продолжаем до тех пор, пока не придем к P_0 -й строке и не отметим элемент этой строки и соответствующий симметричный элемент.

2. Определяется пропускная способность найденного пути:

$$\theta = \min c_{ij}^- .$$

3. Вычисляются новые пропускные способности дуг найденного пути и симметричных с ними:

$$\forall c_{ij}^-, c_{ij} = c_{ij}^- - \theta ,$$

$$\forall c_{ij}^+, c_{ij} = c_{ij}^+ + \theta .$$

В результате получаем новую таблицу, с которой повторяем указанные три шага.

Величина θ представляет собой пропускную способность найденного пути. Если дуга (P_i , P_j) входила в предыдущий путь, то по ней уже пропущено θ вещества. Если (P_i , P_j) входит в один путь, то есте-

ственno, что в этом пути по ней нельзя пропустить больше чем $c_{ij} - \theta$ вещества в единицу времени.

После действия 3 получаем таблицу для сети G с новыми пропускными способностями. Все старые отметки убираются, и возвращаемся к действию 1 общего шага, который применяем до тех пор, пока не придем к окончательной таблице, в которой нет ни одного пути из P_0 в P_n .

По этой таблице легко определить любую дугу, по которой протекает поток. Пропускная способность дуг, по которым протекает поток, уменьшилась по сравнению с c_{ij} на величину x_{ij} , а пропускная способность противоположно направленной (симметричной) дуги увеличилась на x_{ij} , что свидетельствует об отсутствии потока по ней.

Обозначим через U множество вершин сети G , которые достижимы из P_0 по некоторому пути в последней таблице, а множество остальных вершин через V . Тогда увидим, что все дуги сечения (U, V) , направленные из U в V , загружены полностью до пропускных способностей (т. е. x_{ij} для дуги (P_i, P_j) , такой что $P_i \in U, P_j \in V$, $x_{ij} = c_{ij}$), а дуги (P_j, P_i) противоположного направления, идущие из V в U , в построенном потоке не используются (т. е. $x_{ij} = 0, P_i \in U, P_j \in V$), так что величина потока равна

$$Q = \sum_{\substack{P_i \in U \\ P_j \in V}} x_{ij} - \sum_{\substack{P_i \in U \\ P_j \in V}} x_{ji} = \sum_{\substack{P_i \in U \\ P_j \in V}} c_{ij} = C(U, V),$$

т. е. построенный поток максимален, а (U, V) – сечение с минимальной пропускной способностью.

Для определения полученного максимального потока x_{ij}^* , $i, j = \overline{0, n}$, вычитаем из всех элементов начальной таблицы (4) соответствующие элементы таблицы, полученной на последнем шаге. Положительные значения найденных разностей дают величины потоков x_{ij}^* по дугам (P_i, P_j) , а величина потока в сети вычисляется по следующей формуле:

$$Q = \sum_{i=0}^{n-1} x_{in}^* = \sum_{j=1}^n x_{0j}^*.$$

Построим максимальный поток для нашего примера.

*	0	0	0	2	2	1	3	
	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_0		5	9	7 ⁻				
P_1	0		4				3	
P_2	0	5			2	6	2	
P_3	0 ⁺					4		5 ⁻
P_4			1			0		4
P_5			3	3	3		6	6
P_6		2	1			5		8
P_7				0 ⁺	0	0	0	

$$\theta_{\min}\{c_{ij}^-\} = \min\{7, 5\} = 5.$$

Путь $\mu_1(P_0, P_3, P_7)$ состоит из дуг $(P_0, P_3), (P_3, P_7)$.

*	0	0	0	2	2	1	6	
	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_0		5 ⁻	9	2				
P_1	0 ⁺		4				3 ⁻	
P_2	0	5			2	6	2	
P_3	5					4		0
P_4			1			0		4
P_5			3	3	3		6	6
P_6		2 ⁺	1			5		8 ⁻
P_7				5	0	0	0 ⁺	

$$\theta = \min\{8, 3, 5\} = 3; \mu_2(P_0, P_1, P_6, P_7).$$

*	0	0	0	2	2	2	4	
	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_0		2	9 -	2				
P_1	3		4				0	
P_2	0 +	5			2 -	6	2	
P_3	5					4		0
P_4			1 +			0		4 -
P_5			3	3	3		6	6
P_6		5	1			5		5
P_7				5	0 +	0	3	

$$\theta = \min\{4, 2, 9\} = 2; \mu_3(P_0, P_2, P_4, P_7).$$

*	0	0	0	5	2	2	5	
	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_0		2	7 -	2				
P_1	3		4				0	
P_2	2 +	5			0	6 -	2	
P_3	5					4		0
P_4			3			0		2
P_5			3 +	3	3		6	6 -
P_6		5	1			5		5
P_7				5	2	0 +	3	

$$\theta = \min\{6, 6, 7\} = 6; \mu_4(P_0, P_2, P_5, P_7).$$

*	0	0	0	3	2	6		
	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_0		2	1 -	2				
P_1	3		4				0	
P_2	8 +	5			0	0	2 -	
P_3	5					4		0
P_4			3			0		2
P_5			9	3	3		6	0
P_6		5	1 +			5		5 -
P_7				5	2	6	3 +	

$$\theta = \min\{5, 2, 1\} = 1; \mu_5(P_0, P_2, P_6, P_7).$$

*	0	1	0	5	3	2	6	
	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_0		2 -	0	2				
P_1	3 +		4 -				0	
P_2	9	5 +			0	0	1 -	
P_3	5					4		0
P_4			3			0		2
P_5			9	3	3		6	0
P_6		5	2 +			5		4 -
P_7				5	2	6	4 +	

$$\theta = \min\{4, 1, 4, 2\} = 1; \mu_6(P_0, P_1, P_2, P_6, P_7).$$

*	0	1	0	5	3	2	6	
	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_0		1	0	2 -				
P_1	4		3				0	
P_2	9	6			0	0	0	
P_3	5 +					4 -		0
P_4			3			0		2
P_5			9	3 +	3		6 -	0
P_6		5	3			5 +		3 -
P_7				5	2	6	5 +	

$$\theta = \min\{3, 6, 4, 2\} = 2; \mu_7(P_0, P_3, P_5, P_6, P_7).$$

*	0	1						
	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_0		1	0	0				
P_1	4		3				0	
P_2	9	6			0	0	0	
P_3	7					2		0
P_4			3			0		2
P_5			9	5	3		4	0
P_6		5	3			7		1
P_7				5	2	6	7	

Больше нельзя построить нового пути из P_1 в P_n . Вычитаем эту таблицу из первоначальной и получаем:

	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_0		4	9	7				
P_1	-4		1				3	
P_2	-9	-1			2	6	2	
P_3	-7					2		5
P_4			-2			0		2
P_5			-6	-2	0		2	6
P_6		-3	-2			-2		7
P_7				-5	-2	-6	-7	

Положительные величины в этой таблице показывают величины потока по соответствующим дугам. Полученная величина максимального потока

$$Q = \sum_{i=0}^{n-1} x_{in}^* = \sum_{i=1}^n x_{0j}^* = 4 + 9 + 7 = 5 + 2 + 6 + 7 = 20.$$

В последней таблице из P_0 можно построить пути только в P_1 и P_2 , т. е. имеем множество

$$U = \{P_0, P_1, P_2\}$$

и сечение с минимальной пропускной способностью

$$U, V = P_0, P_3, P_1, P_6, P_2, P_6, P_2, P_5, P_2, P_4.$$

Так как

$$c_{03} = 7, c_{16} = 3, c_{26} = 2, c_{25} = 6, c_{24} = 2,$$

то минимальная пропускная способность транспортной сети:

$$C(U, V) = 7 + 3 + 2 + 6 + 2 = 20.$$

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое транспортная сеть? Поток по дуге? Поток в сети?
2. О чём говорит теорема Форда–Фалкерсона?
3. Как строится первоначальная таблица?
4. Как отыскивается новый путь в транспортной сети?
5. Как определить, что нельзя найти нового пути с положительной пропускной способностью?
6. Как найдутся потоки по дугам, соответствующие максимальному потоку в транспортной сети?
7. Как найти сечение с минимальной пропускной способностью и величину максимального потока?

13. ПАРАМЕТРИЧЕСКОЕ ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

13.1. ПОСТАНОВКА ЗАДАЧИ

В практических задачах, как правило, ряд исходных параметров имеет неточные значения, а может пробегать некоторый диапазон изменения. Поэтому для обоснования решения задачи необходимо изучить зависимость оптимального ее решения от вариации некоторых параметров модели.

При проектировании систем, при планировании разработки некоторые параметры могут выбираться с известной свободой. В этом случае полезные рекомендации могут быть получены при использовании аппарата параметрического программирования. Оно также позволяет оценить устойчивость решения по отношению к случайным погрешностям в исходных данных.

Рассмотрим частный случай, когда от параметра t зависят только коэффициенты целевой функции [8]:

$$\bar{P} = \bar{P}_1 + t\bar{P}_2,$$

а вся задача выглядит следующим образом:

$$Q(x) = \bar{P}^T \bar{x} \rightarrow \max,$$

т. е.

$$Q(\bar{x}) = (\bar{P}_1 + t\bar{P}_2)^T \bar{x} \rightarrow \max,$$

$$\begin{cases} A\bar{x} \leq \bar{b}; \\ \bar{x} \geq \bar{0}. \end{cases} \quad (1)$$

Рассмотрим геометрическую интерпретацию такой модели.

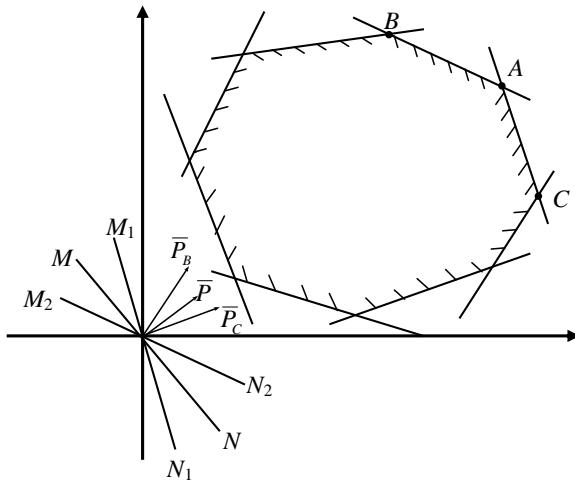


Рис. 13.1. Геометрическая интерпретация задачи параметрического линейного программирования

Для $t = t_0$ линии уровня целевой функции параллельны MN . При $t = \underline{t}$ линии уровня параллельны M_2N_2 , а при $t = \bar{t} - M_1N_1$.

Изменению t от \underline{t} до \bar{t} соответствует поворот MN по часовой стрелке. При $t = t_0$ оптимальное решение соответствует точке A . При $t < \underline{t}$ решение в точке B , при $t > \bar{t}$ – в точке C .

Продолжая рассматривать задачу таким образом, можно разбить заданный диапазон изменения t на конечное число частей, каждой из которых соответствует свой оптимальный план.

Совокупность значений параметра t , при которых данный опорный план оптimalен, называют множеством оптимальности этого плана.

Для исследования параметрической модели воспользуемся алгоритмом метода последовательного улучшения плана.

13.2. АЛГОРИТМ

Задаемся каким-либо t_0 . Если в модели область изменения параметра t ограничена, т. е. $t \in [t_1, t_2]$, то в качестве t_0 можно взять одну из границ.

После конечного числа шагов алгоритма либо придем к оптимальному плану задачи при t_0 (случай 1⁰), либо убедимся, что целевая функция при данном t_0 не ограничена на допустимой области (задача неразрешима) (случай 2⁰).

Рассмотрим сначала случай 1⁰.

Случай 1⁰. Если мы ищем **max** линейной формы, то признаком оптимальности опорного плана является *неотрицательность* коэффициентов P'_i строки критерия.

Эти коэффициенты можно представить в виде следующей суммы:

$$P'_i = P'_{i_1} + tP'_{i_2}, \quad i = 1, \dots, n.$$

Так как план оптimalен для $t = t_0$, то

$$P'_i(t_0) \geq 0, \quad i = 1, \dots, n$$

и, следовательно, совместна система неравенств (из неотрицательности коэффициентов)

$$P'_{i_1} + tP'_{i_2} \geq 0, \quad i = 1, \dots, n. \quad (2)$$

Для всех $P'_{i_2} < 0$ неравенства этой системы можно переписать в виде

$$t \leq -\frac{P'_{i_1}}{P'_{i_2}},$$

а для всех $P'_{i_2} > 0$

$$t \geq -\frac{P'_{i_1}}{P'_{i_2}}.$$

Введем следующие обозначения:

$$\underline{t} = \begin{cases} \max_{P'_{i_2} > 0} \left(-\frac{P'_{i_1}}{P'_{i_2}} \right) \\ -\infty, \text{ если все } P'_{i_2} \leq 0; \end{cases} \quad (3)$$

$$\bar{t} = \begin{cases} \min_{P'_{i_2} < 0} \left(-\frac{P'_{i_1}}{P'_{i_2}} \right) \\ \infty, \text{ если все } P'_{i_2} \geq 0. \end{cases}$$

Таким образом, можно утверждать, что если

$$\underline{t} \leq t \leq \bar{t}, \quad (4)$$

то найденный оптимальный план для t_0 будет оставаться оптимальным для всех t , удовлетворяющих неравенству (4).

Если область изменения $[t_1, t_2]$ параметра t , заданная в технических условиях, не накрывает отрезком $[\underline{t}, \bar{t}]$, то возникает необходимость исследования параметрической модели для

$$t < \underline{t} \text{ и } t > \bar{t}.$$

Это в том случае, если хотя бы $\bar{t} < \infty$ или $\underline{t} > -\infty$.

Исследуем задачу на области $t > \bar{t}$. Пусть

$$\bar{t} = \min_{P'_{i_2} < 0} \left(-\frac{P'_{i_1}}{P'_{i_2}} \right) = -\frac{P'_{k_1}}{P'_{k_2}}.$$

Тогда в опорный план (в базис) необходимо ввести переменную, соответствующую столбцу k (x_k).

Просматривается столбец коэффициентов в таблице. Если среди них нет положительных, то при $t > \bar{t}$ линейная форма не ограничена на допустимом множестве. Если есть положительные коэффициенты, то среди них выбираем тот, для которого отношение свободного члена к соответствующему положительному коэффициенту минимально. Он и берется в качестве разрешающего элемента.

Для нового плана получаем, что $\underline{t}' = \bar{t}$, т. е. наше \bar{t} становится левой границей нового интервала.

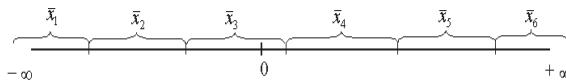
Находится правая граница

$$\bar{t}' = \min_{P'_{i_2} < 0} \left(-\frac{P'_{i_1}}{P'_{i_2}} \right).$$

Если $\bar{t}' = \infty$ или правая граница исходного интервала $[t_1, t_2]$, $t_2 < \bar{t}'$, то исследование в этом направлении прекращается.

Аналогично проводится исследование параметрической модели для $t < \underline{t}$. В этом случае в базис вводят переменную, соответствующую \underline{t} .

В результате исследования за конечное число итераций ось t $(-\infty, \infty)$ разбивается на множества оптимальности, каждому из которых соответствует свой оптимальный план.



Случай 2⁰. Необходимо специально остановиться на этом случае, когда в результате предварительного анализа при $t = t_0$ обнаружено, что целевая функция не ограничена.

Это соответствует тому, что коэффициент в строке целевой функции

$$P'_k = P'_{k_1} + t_0 P'_{k_2} < 0 \quad (5)$$

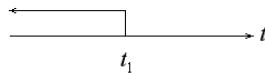
и все коэффициенты в k -м столбце *неположительны*.

При $P'_{k_2} = 0$ условие (5) соблюдается для любого значения параметра, а значит, задача неразрешима на всей оси t .

Если $P'_{k_2} > 0$, то (5) выполняется для всех значений

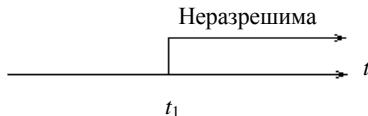
$$t < t_1 = -\frac{P'_{k_1}}{P'_{k_2}}.$$

Неразрешима



Если $P'_{k_2} < 0$, то (5) выполняется при

$$t > t_1 = -\frac{P'_{k_1}}{P'_{k_2}}.$$



Таким образом, в первом случае наша задача неразрешима слева от t_1 , а в другом – справа от t_1 .

Анализ параметрической задачи на луче $t \geq t_1$ начинается с решения задачи линейного программирования при $t = t_1$, отправляясь с имеющегося базиса. Если в этом случае в процессе решения будет найден оптимальный план при t_1 , то решение далее продолжается, как и случае 1⁰.

Если и сейчас процесс окончился выявлением неразрешимости задачи:

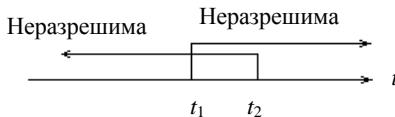
$$P'_S = P'_{S_1} + t_1 P'_{S_2} < 0$$

и в столбце коэффициентов $a_{jS} \leq 0$, $j = 1, \dots, n$, то дальнейший анализ зависит от знака P'_{S_2} . Если $P'_{S_2} = 0$, то задача неразрешима всюду.

Если $P'_{S_2} < 0$, то задача неразрешима при

$$t > t_2 = -\frac{P'_{S_1}}{P'_{S_2}}.$$

И если $t_1 \geq t_2$, то задача неразрешима на всей оси (при $t < t_1$ задача неразрешима и при $t > t_2$ неразрешима).



Если $P'_{S_2} > 0$, то задача неразрешима при

$$t < t_2 = -\frac{P'_{S_1}}{P'_{S_2}}.$$

И если $t_2 > t_1$, исследования продолжаются при $t > t_2$.



В конце концов либо будет найден оптимальный план, либо убедимся, что задача неразрешима на всей оси.

Аналогично исследования проводятся на лучше $t \leq t_1$.

Алгоритм метода последовательного улучшения плана для параметрической модели обладает некоторыми особенностями. Вместо одной строки критерия вводятся три дополнительные строки P'_{i_1} , P'_{i_2} и

$-\frac{P'_{i_1}}{P'_{i_2}}$ для случая 1⁰ и две строки $P'_{i_1} + t_S P'_{i_2}$ и P'_{i_2} в случае 2⁰.

Процесс решения начинается с анализа для некоторого $t = t_0$. После выявления случая Значение 1⁰ вводят строки P'_{i_1} , P'_{i_2} и $-\frac{P'_{i_1}}{P'_{i_2}}$. Значение t_0 стараются выбрать таким образом, чтобы при анализе движение по оси t происходило в одном фиксированном направлении.

Тогда при движении вправо строку с $-\frac{P'_{i_1}}{P'_{i_2}}$ заполняют лишь для позиций, соответствующих $P'_{i_2} < 0$. Если все позиции последней строки оказались незаполненными, то текущий опорный план оптимален для всех $t \geq t_0$, $[t_0, \infty)$. В противном случае индекс минимального элемента этой строки определит индекс переменной, которую надо сделать базисной, а значение этого элемента совпадет с правой границей множества оптимальности текущего опорного плана.

При движении влево заполняются лишь строки, соответствующие $P'_{i_2} > 0$. В этом случае, если последняя строка останется незаполненной, то текущий опорный план оптимален для всех $t \leq t_0$, $(-\infty, t_0]$. Незаполненность последней строки при движении в фиксированном направлении является признаком прекращения анализа в этом направлении, т. е. план остается оптимальным при стремлении t к $\pm\infty$.

Если в модели $t \in [t_1, t_2]$, то этот процесс может закончиться раньше, как только область анализа охватит этот интервал.

$$-\infty \leftarrow t \rightarrow +\infty$$

Пример. Для всех значений параметра t найти максимум линейной формы

$$L(\bar{x}) = (2+3t)x_1 + (-1+2t)x_2 + 3tx_3 + 4x_4 \rightarrow \max$$

при

$$\begin{aligned} x_1 + 2x_2 + x_3 + 3x_4 &\leq 7, \\ -3x_1 + 4x_2 + 3x_3 - x_4 &\leq 15, \\ 2x_1 - 5x_2 + 2x_3 + 2x_4 &\leq 2, \\ x_i &\geq 0, i=1,4. \end{aligned}$$

Решение начинаем при $t = 0$: P'_2

	$-x_1$	$-x_2$	$-x_3$	$-x_4$	1
$y_1 =$	1	2	1	3	7
$y_2 =$	-3	4	3	-1	15
$y_3 =$	2	-5	2	2	2
$P'_1 =$	-2	1	0	-4	0
$P'_2 =$	-3	-2	-3	0	0

	$-x_1$	$-x_2$	$-x_3$	$-y_3$	1
$y_1 =$	-2	19/2	-2	-3/2	4
$y_2 =$	-2	3/2	4	1/2	16
$x_4 =$	1	-5/2	1	1/2	1
$P'_1 =$	2	-9	4	2	4
$P'_2 =$	-3	-2	-3	0	0

	$-x_1$	$-y_1$	$-x_3$	$-y_3$	1
$x_2 =$	$-4/19$	$2/19$	$-4/19$	$-3/19$	$8/19$
$y_2 =$	$-32/19$	$-3/19$	$82/19$	$14/19$	$292/19$
$x_4 =$	$2/19$	$5/19$	$9/19$	$2/19$	$39/19$
$P'_1 =$	$2/19$	$18/19$	$40/19$	$11/19$	$148/19$
$P'_2 =$	$-65/19$	$4/19$	$-65/19$	$-6/19$	$16/19$
$-\frac{P'_1}{P'_2} =$	$2/65$	$-9/2$	$40/65$	$11/6$	$-$

План $\bar{x}_1 = (0, 8/19, 0, 39/19)^T$ оптimalен при $t \in [\underline{t}, \bar{t}]$, где

$$\underline{t} = \max_{P'_{i_2} > 0} \left(-\frac{P'_{i_1}}{P'_{i_2}} \right) = -\frac{9}{2}, \quad \bar{t} = \min_{P'_{i_2} < 0} \left(-\frac{P'_{i_1}}{P'_{i_2}} \right) = \frac{2}{65}.$$

Для того чтобы исследовать задачу при $t < -9/2$, надо ввести в базис y_1 , а при $t > 2/65$ – ввести в базис x_1 .

	$-x_1$	$-x_2$	$-x_3$	$-y_3$	1
$y_1 =$	-2	$19/2$	-2	$-3/2$	4
$y_2 =$	-2	$3/2$	4	$1/2$	16
$x_4 =$	1	$-5/2$	1	$1/2$	1
$P'_1 =$	2	-9	4	2	4
$P'_2 =$	-3	-2	-3	0	0
$-\frac{P'_1}{P'_2} =$					

Этой таблице соответствует оптимальный план $\bar{x}_2 = (0, 0, 0, 1)^T$. Так как все P'_{i_2} неположительны, и мы двигались влево по оси t , то последнюю строку не заполняем, так как полученный план будет опти-

мален для всех $t < -9/2$ и $\underline{t} = -\infty$: $[\underline{t}, \bar{t}] = -\infty, -9/2$. Исследуем модель при $t > 2/65$. Из базиса выводим переменную x_4 .

	$-x_4$	$-y_1$	$-x_3$	$-y_3$	1
$x_2 =$	2	12/19	14/19	1/19	86/19
$y_2 =$	16	77/19	226/19	46/19	916/19
$x_1 =$	19/2	5/2	9/2	1	39/2
$P' =$	-1	13/19	31/19	9/19	109/19
$P'_2 =$	65/2	333/38	455/38	59/19	2467/38
$-\frac{P'_1}{P'_2} =$					

Этой таблице соответствует оптимальный план $\bar{x}_3 = (39/2, 86/19, 0, 0)^T$. Так как все P'_{i_2} положительны, и мы двигались вправо по оси t , то последнюю строку не заполняем, и полученный план будет оптимален для всех $t > 2/65$ и $\bar{t} = \infty$: $\underline{t}, \bar{t} = 2/65, \infty$.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие задачи линейного программирования называются параметрическими?
2. Как ищется некоторый «начальный» оптимальный план? Как находится его множество оптимальности?
3. Как осуществляется дальнейший анализ, если множество оптимальности «уже» области возможных изменений параметра, заданных техническими условиями?
4. Как поступать, если при заданном первоначальном значении параметра столкнулись с неограниченностью целевой функции?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

ИСПОЛЬЗУЕМАЯ ЛИТЕРАТУРА

1. *Карманов В.Г.* Математическое программирование: учеб. пособие. – М.: Физматлит, 2001. – 263 с.
2. *Химмельблау Д.* Прикладное нелинейное программирование. – М.: Мир, 1975. – 534 с.
3. *Васильев О.В.* Методы оптимизации в задачах и упражнениях. – М.: Физматлит, 1999. – 207 с.
4. *Васильев Ф.П.* Численные методы решения экстремальных задач. – М.: Наука, 1980. – 520 с.
5. *Пищеничный Б. Н.* Выпуклый анализ и экстремальные задачи: курс лекций. – М.: Наука, 1980. – 319 с.
6. *Фиакко А., Мак-Кормик Г.* Нелинейное программирование: Методы последовательной безусловной оптимизации. – М.: Мир, 1972. – 240 с.
7. *Юдин Д.Б., Гольштейн Е.Г.* Задачи и методы линейного программирования). – М.: Сов. радио, 1964. – 736 с.
8. *Зуховицкий С.И., Авдеева Л.И.* Линейное и выпуклое программирование. – М.: Наука, 1967.
9. *Гольштейн Е.Г., Юдин Д.Б.* Задачи линейного программирования транспортного типа. – М.: Наука, 1969. – 382 с.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная

1. *Карманов В.Г.* Математическое программирование: учеб. пособие. – М.: Физматлит, 2001. – 263 с.
2. *Дегтярев Ю.И.* Методы оптимизации: учеб. пособие для спец. 0646; 0647. – М.: Сов. радио, 1980. – 270 с.
3. *Эльсгольц Л.Э.* Дифференциальные уравнения и вариационное исчисление. – М.: Наука, 1965. – 424 с.
4. *Аоки М.* Введение в методы оптимизации. – М.: Наука, 1977. – 344 с.
5. *Зангвилл У.И.* Нелинейное программирование. – М.: Сов. радио, 1973. – 312 с.

6. Гольштейн Е.Г., Юдин Д.Б. Задачи линейного программирования транспортного типа. – М.: Наука, 1969. – 382 с.
7. Юдин Д.Б., Гольштейн Е.Г. Задачи и методы линейного программирования. – М.: Сов. радио, 1964. – 736 с.
8. Беллман Р. Динамическое программирование и современная теория управления. – М.: Наука, 1969. – 118 с.
9. Вентцель Е.С. Исследование операций: задачи, принципы, методология. – М.: Дрофа, 2004. – 208 с.
10. Зайченко Ю.П. Исследование операций. – Киев: Вища школа, 1975. – 319 с.
11. Давыдов Э.Г. Исследование операций. – М.: Высшая школа, 1990. – 382 с.
12. Подиновский В.В., Ногин В.Д. Парето-оптимальные решения много критериальных задач. – М.: Наука, 1982. – 254 с.

Дополнительная

13. Васильев О.В. Методы оптимизации в задачах и упражнениях. – М.: Физматлит, 1999. – 207 с.
14. Лутманов С.В. Курс лекций по методам оптимизации. – Ижевск: Изд-во РХД, 2001. – 368 с.
15. Пшеничный Б.Н. Выпуклый анализ и экстремальные задачи: курс лекций. – М.: Наука, 1980. – 319 с.
16. Коршунов Ю.М. Математические основы кибернетики: учеб. пособие для вузов. – М.: Энергоатомиздат, 1987. – 494 с.
17. Акоф Р., Сасиени М. Основы исследования операций. – М.: Мир, 1971. – 533 с.
18. Вагнер Г. Основы исследования операций. – М.: Мир, Т.1, 1972; Т. 2, 1973; Т. 3, 1973.
19. Мусеев Н.Н. Методы оптимизации: учеб. пособие по специальности «Прикладная математика». – М.: Наука, 1978. – 351 с.
20. Численные методы условной оптимизации / ред. Ф. Гилл, У. Мюррей; пер. с англ. В. Ю. Лебедева; под ред. А. А. Петрова. – М.: Мир, 1977. – 299 с.
21. Сухарев А.Г. Курс методов оптимизации: учеб. пособие / А.Г. Сухарев, А. В. Тимохов, В. В. Федоров. – М.: Наука, 1986. – 326 с.
22. Лесин В.В. Основы методов оптимизации: учеб. пособие для втузов. – М.: МАИ, 1998. – 340 с.
23. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. – М.: Наука, 1965. – 457 с.
24. Кюнци Г.П., Крелле В. Нелинейное программирование. – М.: Сов. радио, 1965. – 303 с.

Очень полезная литература

25. Химмельблау Д. Прикладное нелинейное программирование. – М.: Мир, 1975. – 534 с.
26. Васильев Ф.П. Численные методы решения экстремальных задач. – М.: Наука, 1980. – 520 с.
27. Фиакко А., Мак-Кормик Г. Нелинейное программирование: Методы последовательной безусловной оптимизации. – М.: Мир, 1972. – 240 с.
28. Сеа Ж. Оптимизация: Теория и алгоритмы. – М.: Мир, 1973. – 244 с.
29. Полак Э. Численные методы оптимизации: Единый подход. – М.: Мир, 1974. – 376 с.
30. Пшеничный Б.Н., Данилин Ю.М. Численные методы в экстремальных задачах. – М: Наука, 1975. – 319 с.
31. Коробкин А.Д., Лемешко Б.Ю., Цой Е.Б. Математические методы оптимизации: учеб. пособие. – Новосибирск, 1977.
32. Зуховицкий С.И., Авдеева Л.И. Линейное и выпуклое программирование. – М.: Наука, 1967.
33. Саати Т.Л. Математические методы исследования операций. – М.: Мир, 1973.
34. Хедли Дж. Нелинейное и динамическое программирование. – М.: Мир, 1968.

Лемешко Борис Юрьевич

МЕТОДЫ ОПТИМИЗАЦИИ

Конспект лекций

Редактор *И.Л. Кескевич*
Выпускающий редактор *И.П. Брованова*
Корректор *И.Е. Семенова*
Дизайн обложки *А.В. Ладыжская*
Компьютерная верстка *Н.М. Шуваева*

Подписано в печать 27.07.2009. Формат 60 x 84 1/16. Бумага офсетная
Тираж 150 экз. Уч.-изд. л. 9,06. Печ. л. 9,75. Изд. № 20. Заказ №
Цена договорная

Отпечатано в типографии
Новосибирского государственного технического университета
630092, г. Новосибирск, пр. К. Маркса, 20