

КИБЕРНЕТИКА



В. И. МУДРОВ    А. А. ИВЛЕВ

# **МАЖОРАНТЫ НЬЮТОНА В ПРИКЛАДНЫХ ЗАДАЧАХ**

**Настоящая серия выпускается по рекомендации IX Международного совещания представителей научно-технических издательств социалистических стран (июнь 1975 г.)**

В. И. МУДРОВ  
А. А. ИВЛЕВ

# **МАЖОРАНТЫ НЬЮТОНА В ПРИКЛАДНЫХ ЗАДАЧАХ**

ТЕОРИЯ,  
АЛГОРИТМЫ,  
ПРОГРАММЫ



МОСКВА  
«РАДИО И СВЯЗЬ»  
1987

**Мудров В. И., Ивлев А. А.**

**М89** Мажоранты Ньютона в прикладных задачах. Теория, алгоритмы, программы. — М.: Радио и связь, 1987. — 144 с.: ил. — (Кибернетика).

Описывается один из новых методов отыскания абсолютного экстремума суммы слабо выпуклых и негладких функций. Использование метода демонстрируется на примерах решения возникающих при проектировании сложных радиотехнических систем задач безусловной и условной оптимизации: размещение точечных и линейных объектов, построение алгоритмов обработки измерительной информации, оценка технического уровня изделий, обработка экспертной информации и др. Приводятся алгоритмы решения некоторых практических задач, программы на языке ПЛ/1 и результаты числовых расчетов.

Для инженерно-технических работников, использующих математические методы в своей деятельности; может быть полезна студентам технических вузов.

М 1502000000-132  
046(01)-87 56-87

ББК 22.18

Рецензенты: д-р физ.-мат. наук В. А. Горелик  
и д-р техн. наук Л. С. Гурин

**Редакция литературы по кибернетике и вычислительной технике**

Производственное издание

**Владимир Иванович Мудров**  
**Алексей Алексеевич Ивлев**

**МАЖОРАНТЫ НЬЮТОНА В ПРИКЛАДНЫХ ЗАДАЧАХ.**  
**ТЕОРИЯ, АЛГОРИТМЫ, ПРОГРАММЫ**

Заведующая редакцией Г. И. Козырева  
Редактор Н. Г. Давыдова  
Художник К. М. Присолов  
Художественный редактор Н. С. Шейн  
Технический редактор Э. Н. Ратникова  
Корректор З. Г. Галушкина

**ИБ № 1481**

---

Сдано в набор 18.11.86. Подписано в печать 13.02.87. Т-08327 Формат 60×84<sup>1</sup>/<sub>16</sub>.  
Бумага офсетная № 2. Гарнитура литературная. Печать офсетная. Усл. печ. л. 8,37.  
Усл. кр.-отт. 8,951. Уч.-изд. л. 9,20. Тираж 7000 экз. Изд. № 21717. Зак. 2213. Цена 45 к.  
Издательство «Радио и связь», 101000 Москва, Почтамт, а/я 693

---

Московская типография № 4 «Союзполиграфпром» при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли.  
129041, Москва, Б. Переяславская, 46

© Издательство «Радио и связь», 1987

## ПРЕДИСЛОВИЕ

В задачах проектирования сложных технических систем часто необходимо отыскивать минимум суммы выпуклых вниз функций с плохими аналитическими свойствами. Например, при проектировании системы (сети) каналов связи возникает задача размещения коммутационных узлов, которая ставится так: учитывая расположение абонентских пунктов, характер каналов связи, обусловленный интенсивностью переговоров, дать рекомендации по размещению узлов, обеспечив минимум суммарной стоимости каналов. При этом в соответствующей математической модели каждый пункт может быть представлен точкой в некоторой системе координат (если внутри этого пункта сеть связи развита слабо) или объектом конечных размеров: линейным отрезком, кругом и т. д. (если этот пункт имеет определенную протяженность и достаточно подвести канал к любой точке этого объекта). Может возникнуть задача соединения абонентских пунктов с помощью центральной магистрали, к которой необходимо подвести линии от конечных пунктов, учитывая некоторые дополнительные ограничения, обусловленные спецификой местности (например, в черте города обычно они прокладываются только вдоль улиц).

Во всех перечисленных случаях разработчику приходится решать некоторую оптимизационную задачу по минимизации суммы взвешенных расстояний, где значение веса определяется спецификой канала. К сожалению, вид входящих в эту сумму слагаемых с математической точки зрения достаточно сложен и не всегда допускает эффективное применение общеизвестных приемов отыскания экстремума.

Подобные задачи возникают и при разработке схем соединения элементов с использованием печатного и объемного монтажа. Напомним, что задача последовательного соединения элементов формулируется как хорошо известная в математике задача о коммивояжере [20]. Задача же параллельного соединения элементов сводится к задаче Вебера — Штейнера, многочисленные разновидности которой рассматриваются в настоящей книге.

Другой тип оптимизационных задач, возникающих при проектировании радиотехнических систем, связан с обработкой измерений. Как известно, основным методом получения оценок неизвестных параметров является построение и минимизация некоторой функции этих параметров (например, минус логарифм функции правдоподобия). При засорении общего массива измерений помехами минимизируемая функция принимает вид, похожий на тот, который имеет место при решении задач размещения коммутационных узлов.

При проектировании радиотехнических систем широко распространены задачи, связанные с выбором базового объекта. В этих случаях каждая альтернатива может рассматриваться как некоторая точка в многомерном пространстве и, следовательно, соответствующие задачи выбора очень близки к задачам размещения объектов. Помимо этого, такие задачи, как правило, являются задачами многокритериальной или векторной оптимизации. Их редко удается решить без привлечения дополнительной информации, полученной от экспертов. Эта информация, первоначально представленная, как правило, в нечисловой форме (ранжировки по предпочтительности, матрицы парных сравнений, различные классификации), подвергается обработке с целью приведения в более удобную для использования «компактную» форму — в форму шкалы.

В задачах получения различных шкал (полезности, предпочтительности, перспективности и др.) обычно предполагается, что объективно существует некоторая неизвестная лицу, принимающему решение, величина, соответствующая оценке объекта на данной шкале. Решение задачи шкалирования в этом случае сводится к нахождению экстремума некоторого критерия соответствия искомой шкалы данным, полученным от экспертов. И здесь функция, подлежащая минимизации, не всегда обладает гладкостью, сильной выпуклостью и другими свойствами, допускающими применение стандартных алгоритмов.

Указанные задачи еще более усложняются, если на множество независимых переменных наложены ограничения.

Некоторые из этих задач успешно решаются при помощи метода вариационно-взвешенных квадратических мажорант (ВВКМ). В отечественной литературе его в неявной форме применял Е. Я. Ремез при построении чебышевских приближений [26], а впоследствии использовали В. И. Мудров и В. Л. Кушко при обработке измерений нетрадиционными способами [21], а также И. И. Дикин и В. И. Зоркальцев при решении задач математического программирования [12]. В иностранной литературе этот метод связывают с именем Е. Вейсфельда [41], который впервые применил его для решения одной из важных задач размещения (задачи Вебера—Штейнера). Статья Вейсфельда, опубликованная в 1937 г. на французском языке в японском журнале *Tôhoku Mathematical Journal*, долгое время оставалась в тени. Однако в последние годы интерес к методу ВВКМ резко возрос. Его возможности стали оживленно обсуждаться на страницах многих журналов по исследованию операций и вычислительной математике, а сам метод находит все новые практические приложения. Доказана линейная сходимость итерационного процесса, порождаемого этим методом, и прослежены его связи с другими методами поиска экстремума. Итеративные методы наименьших квадратов, к которым относится метод ВВКМ, распространены на задачи математического программирования с ограничениями [12] и т. д.

Хотя метод ВВКМ внешне похож на метод Ньютона, идейная основа этих методов различна: в методе Ньютона центральной является идея аппроксимации, в методе ВВКМ — идея мажорирования. При этом хорошая аппроксимация минимизируемой функции весьма желательна, но не обязательна. Главные преимущества метода ВВКМ заключаются в простоте машинной реализации, отсутствии затруднений, связанных с выбором начального приближения, а также в том, что он не столь сильно боится «проклятия» размерности как другие методы. Класс функций, к которому приложим ВВКМ, к сожалению, не очень широк, однако это компенсируется актуальностью и важностью задач, входящих в этот класс.

При описании прикладных задач авторы использовали терминологию, допускающую достаточно широкое толкование. Например, под термином «объект» в задачах размещения в конкретных случаях можно подразумевать и радиоэлемент, и ЭВМ, и пункт связи, и РЛС; в задачах обработки измерений не уточняется содержание термина «измеряемая величина», а в задачах шкалирования нет привязки к конкретной шкале. Использование более общей терминологии должно способствовать расширению круга лиц, которые могут с пользой применить предлагаемый метод в своей работе. По нашему убеждению, систематическое изложение основ метода ВВКМ и его основных приложений должно вызвать интерес у специалистов самого различного профиля, сталкивающихся с решением задач оптимизации.

## Глава 1.

### ЧИСЛЕННЫЕ МЕТОДЫ ОТЫСКАНИЯ АБСОЛЮТНОГО ЭКСТРЕМУМА

Алгоритмы безусловной оптимизации принято делить на классы в зависимости от порядка производных, используемых в ходе вычислений. Если при вычислениях используются значения только самой функции в различных точках, то соответствующие методы называются методами нулевого порядка (или прямыми методами); если кроме значений функции вычисляются и используются значения первых и вторых производных, то мы имеем дело соответственно с методами первого и второго порядков. По мере повышения порядка используемых производных естественно возрастают и ограничения, накладываемые на минимизируемую функцию. Достаточно сказать, что само существование вторых производных предполагает наличие и непрерывность первых производных минимизируемой функции.

Для того чтобы лучше «почувствовать» место, которое занимает метод ВВКМ среди других методов оптимизации, определить в чем сходство и в чем различие между ними, в данной главе мы более или менее подробно рассмотрим сущность этих методов. При описании методов будем часто ссылаться на достаточно популярные источники: учебник Н. Н. Моисеева, Ю. П. Иванилова и Е. М. Столяровой «Методы оптимизации» [19] и книгу Б. Т. Поляка «Введение в оптимизацию» [25].

#### 1.1. МЕТОДЫ НУЛЕВОГО ПОРЯДКА (ИЛИ ПРЯМЫЕ МЕТОДЫ)

Будем предполагать, что минимизируемая функция  $f(x)$  от скалярного аргумента унимодальна на отрезке  $[a, b]$ . Это означает, что она строго монотонно убывает на отрезке  $[a, x_{\min}]$  и строго монотонно возрастает на отрезке  $[x_{\min}, b]$ , где  $x_{\min} = \underset{x}{\operatorname{argmin}} f(x)$ . Никаких других ограничений на  $f(x)$  не налагается.

Методы нулевого порядка основаны на последовательном сокращении размеров области, в которой содержится единственный экстремум. Каждый из этих методов сводится к определению точек  $x_i$ , которые разбивают первоначальный отрезок на две или более непересекающиеся части и в которых вычисляется  $f(x)$ . В дальнейшем эти точки будем называть точками приложения измерительного прибора.

Учитывая, что один вычислительный эксперимент не в состоянии сузить интервал неопределенности, будем считать, что даны две точки  $x_1$  и

$x_2$  ( $x_1 < x_2$ ) на отрезке  $[a, b]$ . Если  $f(x_1) > f(x_2)$ , то из унимодальности  $f(x)$  следует, что точка  $x_{\min}$  расположена правее  $x_1$ . Если, наоборот,  $f(x_1) < f(x_2)$ , то  $x_{\min}$  расположена левее  $x_2$ . Наконец, если  $f(x_1) = f(x_2)$ , то точка минимума расположена внутри отрезка  $[x_1, x_2]$ .

Поскольку на функцию  $f(x)$  не наложено никаких других ограничений, кроме унимодальности, мы вынуждены сформулировать задачу определения точек приложения измерительного прибора как задачу на минимакс и определить  $x_1$  и  $x_2$  из условия

$$L = \min_{x_1, x_2} \max(b - x_1, x_2 - a).$$

Полагая для определенности  $a = 0$  и  $b = 1$ , получаем, что  $x_1 = 1/2 - \epsilon$ ,  $x_2 = 1/2 + \epsilon$ , где  $\epsilon$  — произвольно малое положительное число. Его можно истолковать как такой минимальный интервал, при котором прибор (алгоритм) еще различает любые два значения функции  $f(x)$  во всей области ее определения.

Итак, точки приложения прибора должны быть расположены вблизи точки  $x = 1/2$  — один раз правее, и один раз левее нее. Искомая точка минимума расположена в интервале  $[0, 1/2 + \epsilon]$ , если  $f(x_1) < f(x_2)$ , и в интервале  $[1/2 - \epsilon, 1]$ , если  $f(x_1) > f(x_2)$ . Если вдруг окажется, что  $f(x_1) = f(x_2)$ , то точка минимума расположена в интервале  $[1/2 - \epsilon, 1/2 + \epsilon]$ , но, как было оговорено, этот случай, ввиду его исключительности, почти не влияет на общее развитие событий.

После осуществления такого эксперимента задача редуцирована к аналогичной задаче, в которой первоначальная область неопределенности уменьшена почти вдвое, так что  $L_1 = L_0/2 + \epsilon$ . Если осуществлено  $2n$  таких экспериментов, то размер области неопределенности удастся сократить до  $L_n = L_0/2^n + 2\epsilon(1 - 2^{-n})$ . Сделать эту область меньше  $2\epsilon$  не удастся ввиду ограниченной чувствительности прибора.

Если отказаться от требования попарного использования измерений, то процедура поиска может быть несколько улучшена за счет более рационального использования ранее накопленной информации.

Довольно тонкие и пространные рассуждения по отысканию оптимальной стратегии приводят к различным схемам проведения экспериментов в случае, когда заранее известно допустимое число элементарных измерительных актов, и в противоположном случае, когда это число неизвестно.

Если знать заранее число возможных экспериментов  $n$ , то надо определить два числа Фибоначчи  $F_{n-1}$ ,  $F_{n-2}$  и разделить первоначальный отрезок неопределенности в отношении  $F_{n-1}/F_{n-2}$ \*) (рис. 1.1). Отрезок, пропорциональный  $F_{n-1}$ , надо затем разбить еще на два в отношении  $F_{n-2}$ :  $F_{n-3}$  ( $F_{n-2} + F_{n-3} = F_{n-1}$ ). Если окажется, что

\*) Числами Фибоначчи называются члены ряда 1, 1, 2, 3, 5, 8, 13, 21, 34, ... Общий закон образования ряда Фибоначчи таков:  $F_0 = F_1 = 1$ ,  $F_n = F_{n-1} + F_{n-2}$ ,  $n = 2, 3, \dots$

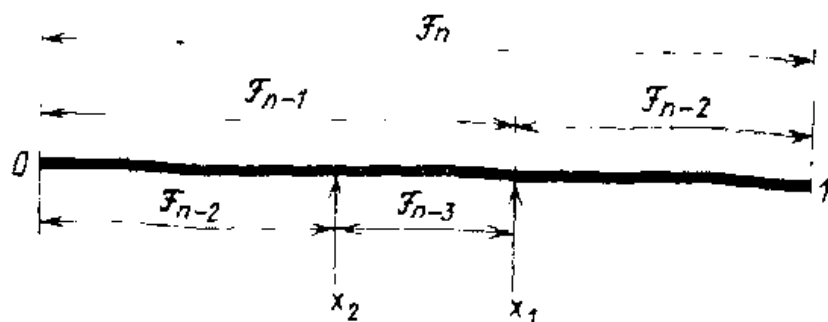


Рис. 1.1

точка, разбивающая его на части, пропорциональные  $F_{n-2}$  и  $F_{n-3}$ . Таким образом, и здесь задача редуцирована к первоначальной задаче, но с меньшей по размерам областью неопределенности.

Повторяя указанную процедуру, мы в состоянии сузить отрезок неопределенности до размера  $L_0 F_{n-2}/F_n$ , а с учетомвилки нечувствительности — до размера

$$L_0 \frac{F_{n-2}}{F_n} + \epsilon \left( 1 + \frac{F_{n-2}}{F_{n-1}} \right) = L_0 \frac{F_{n-2}}{F_n} + \epsilon F_{n-2} \left( \frac{1}{F_{n-2}} + \frac{1}{F_{n-1}} \right).$$

После  $n$  измерений размеры области неопределенности уменьшаются до

$$L_n = L_0 \frac{1}{F_n} + \epsilon \sum_{k=1}^n \frac{1}{F_{n-k}}$$

В данной методике предполагается, что выбор (назначение) точек приложения измерительного прибора зависит от  $n$ . Если  $n$  неизвестно, то можно исходить из предположения, что оно достаточно велико. В этом случае вместо того, чтобы делить первоначальный отрезок в отношении  $\alpha_n = F_{n-2}/F_{n-1}$ , будем делить его в отношении  $\alpha = \lim_{n \rightarrow \infty} \alpha_n$ .

К счастью, этот предел существует, равен  $(1 + \sqrt{5})/2 = 1,618...$  и называется основанием ряда Фибоначчи, а соответствующее ему сечение отрезка — золотым сечением. После  $n$  экспериментов размер отрезка неопределенности сокращается до

$$L_n = L_0 / \alpha^{n-1} + \epsilon (1 - 1/\alpha^n) / (1 - 1/\alpha).$$

Несложные расчеты показывают, что при больших  $n$  и малых  $\epsilon$  длина окончательного интервала неопределенности, полученная методом золотого сечения, всего на 17 % больше длины, найденной при использовании истинных значений чисел Фибоначчи.

В многомерном случае методы прямого поиска становятся намного более трудоемкими [2, 6, 32]. К методам прямого поиска относится, по существу, и такой метод решения экстремальных комбинаторных задач, как метод ветвей и границ [3].

## 1.2. МЕТОДЫ ПЕРВОГО ПОРЯДКА (ИЛИ ГРАДИЕНТНЫЕ МЕТОДЫ)

Пусть дана функция многих переменных  $f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$ . Уравнения  $f(x_1, x_2, \dots, x_n) = C$  определяют линии равного уровня поверхности  $z = f(x_1, x_2, \dots, x_n)$ , а вектор-градиент  $\nabla_x f$  — направление и скорость наибольшего возрастания значения этой функции. Равный по значению и противоположный по направлению вектор называется антиградиентом. Под  $\nabla_x$  здесь и далее понимается дифференциальный оператор Гамильтона, т. е. вектор-столбец, элементы которого определены чисто формально и равны  $\frac{\partial}{\partial x_i}$ . Таким образом,

$$\nabla_x^T = \left\| \frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_n} \right\|$$

( $T$  — индекс транспонирования).

Простейшим из методов первого порядка является метод, при котором очередной шаг из точки  $\mathbf{x}^k$  в точку  $\mathbf{x}^{k+1}$  осуществляется в соответствии с формулой

$$\mathbf{x}^{k+1} = \mathbf{x}^k - h \nabla_x f(\mathbf{x}) \pmod{\nabla f(\mathbf{x})}, \quad (1.1)$$

где длина шага  $h$  назначается постоянной. Процесс продолжается до тех пор, пока  $f(\mathbf{x}^{k+1})$  не превысит  $f(\mathbf{x}^k)$  при каком-нибудь  $k$ . Для более точного отыскания минимума процесс обычно продолжают и далее, но с меньшим шагом (например,  $h/2$ ).

Недостатком этого метода является низкая скорость сходимости последовательности точек  $\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^k$  к точке минимума  $\mathbf{x}^*$ . Достоинством — простота вычислений.

Иногда критерием для уменьшения шага является условие  $|f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)| < \epsilon$ , где  $\epsilon$  — малое число. В этом случае в окрестности некоторых особенностей рельефа, еще до достижения точки минимума, шаг может измельчиться настолько, что процесс практически остановится, намного не доходя до искомой точки.

В целях ускорения сходимости итерационного процесса шаг чаще всего с самого начала делают непостоянным. Так, для упрощения алгоритма и уменьшения объема вычислений очередной шаг выполняют поочередно в направлении каждой координатной оси, закрепив значения остальных координат. Длину шага определяют с помощью прямых ме-

\* Скорость сходимости обычно оценивают путем сравнения с геометрической прогрессией. Если

$$\|\mathbf{x}^k - \mathbf{x}^*\| / \|\mathbf{x}^0 - \mathbf{x}^*\| = (q + \delta)^k,$$

где  $q = \text{const} < 1$ , а  $\delta = O(q^k)$ , то говорят, что последовательность  $\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^k$  имеет линейную сходимость или сходимость со скоростью геометрической прогрессии. Если показатель степени в правой части этого уравнения больше  $k$ , то говорят о сверхлинейной сходимости, если меньше — о сходимости, худшей по сравнению с линейной.

тодов, осуществляя прямой одномерный поиск в направлении свободной координаты  $x_1$ . После отыскания точки минимума  $(x_1^{k+1}, x_2^k, \dots, x_n^k)$  «освобождают» координату  $x_2$  и производят прямой поиск по этой координате. Таким же образом поступают и далее. После минимизации по  $x_n$  функции  $f(x_1^{k+1}, x_2^{k+1}, \dots, x_n^{k+1}, x_n)$  возвращаются к первой координате.

В методе «кратчайшего» спуска одномерный поиск осуществляется в направлении антиградиента. В качестве  $x^{k+1}$  принимается  $x^k + \alpha^k t_{\min}^k$ , где  $\alpha^k = \|\alpha_i^k\|$ ;

$$t_{\min}^k = \arg \min_t f(x_1^k + \alpha_1^k t, x_2^k + \alpha_2^k t, \dots, x_n^k + \alpha_n^k t);$$

$$\alpha_i^k = \frac{\partial f}{\partial x_i} / \sqrt{\sum_{i=1}^n \left( \frac{\partial f}{\partial x_i} \right)^2} \Big|_{x_i = x_i^k}$$

В этих методах выигрыш благодаря уменьшению числа шагов обычно перекрывает потери на одномерный поиск при определении длины каждого шага.

Для выпуклых недифференцируемых функций можно ввести понятие, аналогичное градиенту, а именно, понятие субградиента. Напомним, что градиентом  $\nabla f$  называют вектор-столбец частных производных функции  $f(x)$ . Для градиента всегда выполняется условие

$$f(x+y) = f(x) + \nabla f^T y + o(y),$$

где  $x \in R^n$ ,  $y \in R^n$ , а  $R^n$  — обозначение  $n$ -мерного евклидова пространства.

Субградиентом функции  $f(x)$  в точке  $x$  называется вектор  $\partial f$ , для которого справедливо соотношение

$$f(x+y) \leq f(x) + \partial f^T y. \quad (1.2)$$

Субградиент, вообще говоря, определяется неоднозначно (рис. 1.2), в точке  $x$  можно построить целое множество субградиентов.

Для выпуклых дифференцируемых функций множество  $\partial f$  содержит только один элемент  $\nabla f$ .

Часто с помощью  $\partial f$  обозначают и все множество векторов, для которых справедливо неравенство (1.2), и любого представ

ителя этого множества. Однако такая неоднозначность в обозначениях не приводит к недоразумениям, так как обычно по смыслу ясно, о чем идет речь.

С помощью субградиента можно:

а) вычислить производную по направлению, определив ее следующим образом:  $df/dl = \max_{v \in \partial f} [vl]$ ;

б) найти направление наискорейшего спуска  $g_0 = -v/|v_0|$ , где  $v_0 = \min_{v \in \partial f} |v|$ ;

в) записать условие минимума  $0 \in \partial f(x^*)$ .

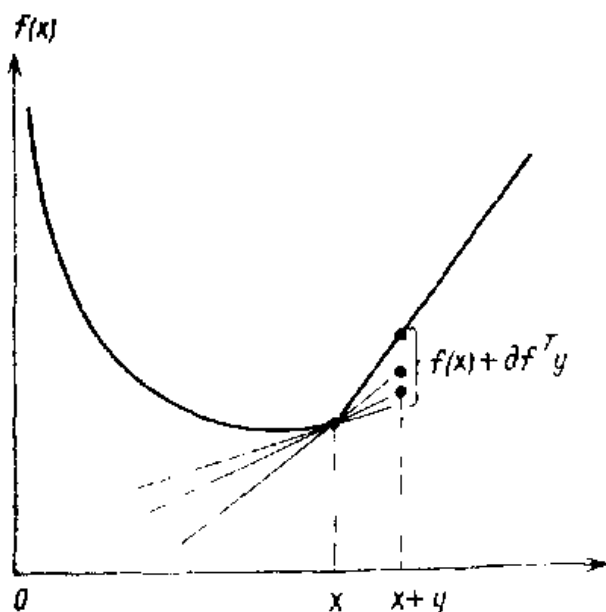


Рис. 1.2

Иными словами, субградиент обладает примерно такими же свойствами по отношению к выпуклым функциям, что и градиент по отношению к дифференцируемым функциям. Поэтому, казалось бы, для отыскания минимума недифференцируемой функции можно было бы применить градиентные методы, например метод наискорейшего спуска. К сожалению, это не так, и прямой аналог метода наискорейшего спуска в применении к недифференцируемым функциям работать не будет. Вообще «субградиентный метод не может считаться эффективным средством решения выпуклых недифференцируемых задач и возникает потребность в более мощных методах оптимизации» [25].

Покоординатный спуск, «кратчайший» спуск, как, впрочем, и другие разновидности градиентного метода, очень чувствителен к появлению «оврагов», т. е. таких особенностей функции, когда линии равного уровня функции  $f(x)$  вытянуты в каком-нибудь одном направлении (рис. 1.3).

В окрестностях оврагов наблюдается особенно интенсивное измельчение длины шага. Для борьбы с этим явлением обычно поступают так: осуществив с малым шагом спуск на дно оврага из каких-либо двух то-

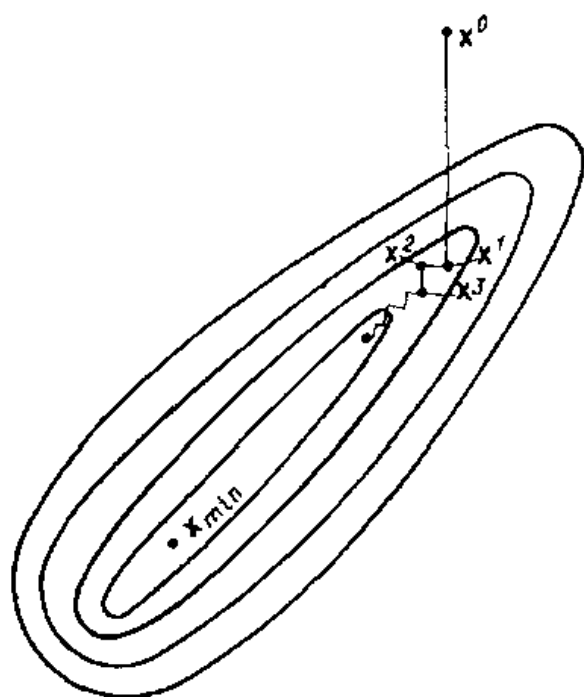


Рис. 1.3

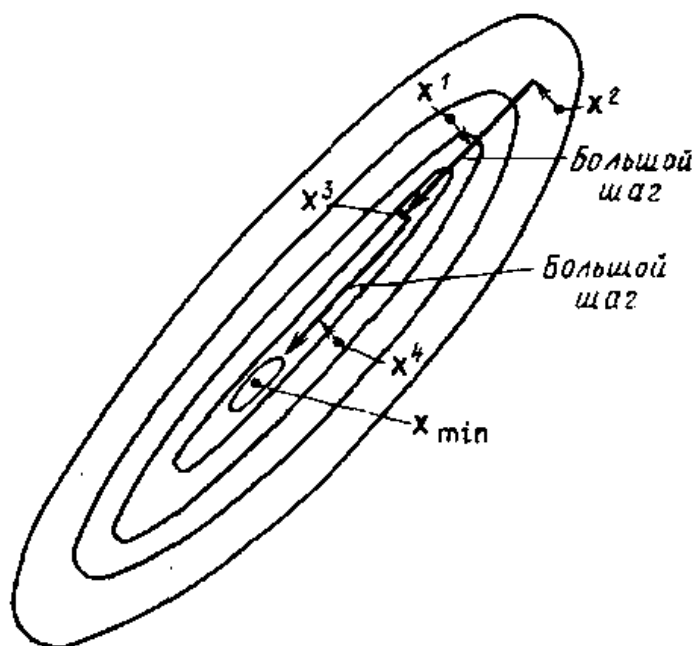


Рис. 1.4

чек, нащупывают линию «водостока» и следующий (большой) шаг делают в направлении этой линии, после чего, добавив еще одну точку, повторяют описанную процедуру (рис. 1.4). Этот метод так и называется: «метод оврагов».

Другим направлением ускорения итерационного процесса является переход к методам второго порядка (релаксационным методам) \*).

\* ) Иногда в литературе релаксационным методом называют метод покоординатного спуска.

### 1.3. МЕТОДЫ ВТОРОГО ПОРЯДКА (ИЛИ РЕЛАКСАЦИОННЫЕ МЕТОДЫ)

Начнем с рассмотрения функции одного аргумента. Из начальных разделов анализа известно, что если минимизируемая функция имеет непрерывную первую производную, то необходимым условием того, что минимум находится в точке  $x_{\min}$ , является условие

$$f'(x)|_{x=x_{\min}} = 0. \quad (1.3)$$

Если в этой точке, кроме того,

$$f''(x)|_{x=x_{\min}} > 0, \quad (1.4)$$

то эти два условия являются и достаточными.

Обозначим  $\varphi(x) = f'(x)$ . Условие  $f'(x) = 0$  переписывается к виду  $\varphi(x) = 0$ . Для отыскания корней достаточно гладкой функции используется обычно метод Ньютона. Начав с какой-нибудь точки  $x^k$ , находят в этой точке значение функции  $\varphi(x^k)$  и ее производной и проводят через

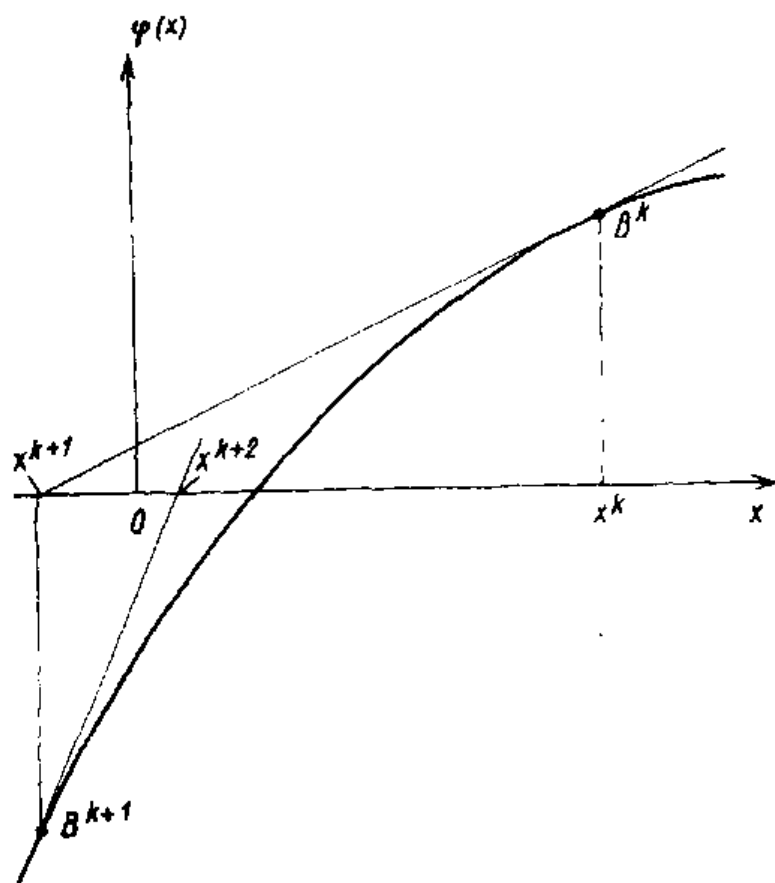


Рис. 1.5

точку  $B(x^k, \varphi(x^k))$  касательную к функции  $\varphi(x)$  до ее пересечения с осью абсцисс (рис. 1.5).

Точка пересечения  $x^{k+1}$  принимается в качестве начальной точки на следующем шаге итерационного процесса. Уравнением касательной является

$$\psi(x) = \varphi(x^k) + \varphi'(x^k)(x - x^k). \quad (1.5)$$

Полагая  $\psi(x) = 0$ , получаем, что

$$x^{k+1} = x^k - \varphi(x^k)/\varphi'(x^k). \quad (1.6)$$

В многомерном случае вместо производной  $f'(x)$  мы должны приравнять нулю составляющие вектор-градиента  $\nabla f(x)$ . Раскладывая каждую компоненту вектор-градиента в ряд по степеням  $\Delta x_i$  и приравни-

вая нулю их линейную часть, получаем следующую систему уравнений:

$$\begin{aligned} q_1(x_1, x_2, \dots, x_s^k) + \sum_{s=1}^n \frac{\partial q_1}{\partial x_s} (x_s - x_s^k) &= 0, \\ q_2(x_1, x_2, \dots, x_s^k) + \sum_{s=1}^n \frac{\partial q_2}{\partial x_s} (x_s - x_s^k) &= 0, \\ &\vdots \\ q_n(x_1, x_2, \dots, x_s^k) + \sum_{s=1}^n \frac{\partial q_n}{\partial x_s} (x_s - x_s^k) &= 0. \end{aligned} \quad (1.7)$$

Вспомнив, что  $\|\varphi_1, \varphi_2, \dots, \varphi_n\| = \nabla_x^T f$ , эту систему можно переписать в более компактном виде

$$\nabla_x f + (\nabla_x \nabla_x^T f|_{x=x^k})(x - x^k) = 0, \quad (1.8)$$

## Откуда

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \Gamma_{\mathbf{x}=\mathbf{x}^k}^{-1} \nabla_{\mathbf{x}} f, \quad (1.9)$$

где  $\Gamma = \nabla_x \nabla_x' f$  — матрица вторых производных функции  $f(x)$ , называемая иногда матрицей Гессе или гессианом.

В одномерном случае итерационный процесс, описываемый формулой

$$x^{k+1} = x^k - (f''(x))^{-1} f'(x), \quad (1.10)$$

допускает еще одну интерпретацию, представляющую для нас особый интерес. Аппроксимируем функцию  $f(x)$  в окрестности точки  $x$  параболой  $\psi(x) = A + (x - C)^2 B/2$  и предъявим к ней следующие требования:

$$\psi(x^k) = f(x^k), \quad \psi'(x^k) = f'(x^k), \quad \psi''(x^k) = f''(x^k).$$

Это приводит к следующим выражениям:

$$A = f(x^k) - \frac{f'(x^k)}{2} \frac{f'(x^k)}{f''(x^k)}, \quad B = f''(x^k), \quad C = x^k - \frac{f'(x^k)}{f''(x^k)}.$$

Парабола  $\psi(x)$  имеет единственную точку минимума  $C$ , которую мы и принимаем за следующее приближение к  $x_{\min} = \operatorname{argmin} f(x)$ , так что

$$x^{k+1} = x^k - f'(x^k)/f''(x^k).$$

Аналогично в многомерном случае формула (1.9) определяет точку минимума поверхности

$$\psi(\mathbf{x}) = A + (\mathbf{x} - \mathbf{C})^T \mathbf{\Gamma}_{\mathbf{x}, \mathbf{x}}^{-1, k} (\mathbf{x} - \mathbf{C}), \quad (1.11)$$

касающейся поверхности  $z = f(x)$  в точке  $x^k$  и имеющей одинаковые с нею первые и вторые производные.

Парабола (1.11) называется параболой Ньютона, а сам метод, опирающийся на формулы (1.6) и (1.9), — методом Ньютона.

Последовательность точек, полученных методом Ньютона, сходится к точке минимума лишь в том случае, когда начальная точка  $x^k$  расположена вблизи нее, а сама функция в окрестности  $x_{\min}$ , включающей точку  $x^k$ , удовлетворяет условию сильной выпуклости.

О степени выпуклости функций судят по поведению ее производных. Если  $f'(x)$  по любому направлению возрастает, то функция  $f(x)$  называется выпуклой, если  $f'(x)$  возрастает монотонно, то функция  $f(x)$  называется строго выпуклой. Если  $f'(x)$  возрастает не только монотонно, но и равномерно, так что  $M|y| \geq f'(x+y) - f'(x) \geq m|y|$ , где  $M > m \geq \delta > 0$ , то функция  $f(x)$  называется сильно выпуклой. Несильно выпуклые функции в дальнейшем будем называть слабо выпуклыми.

Для любой точки  $x^k$  график сильно выпуклой функции  $f(x)$  целиком заключен между двумя параболой, касающимися этой функции в точке  $x^k$  (рис. 1.6). Математически это означает, что при любых  $x$  и  $x + y$  из области определения функции

$$my^T y \leq y^T \Gamma_x y \leq My^T y,$$

где  $\Gamma_x = \nabla_x \nabla_x^T f(x)$ , а  $M$  и  $m$  — некоторые положительные числа, для которых  $M \geq m > 0$ .

В общем случае процесс расходится. Отыскание первого приближения, т. е. нахождение точки, расположенной в достаточной близости от экстремума, — задача не менее сложная, чем последующее отыска-

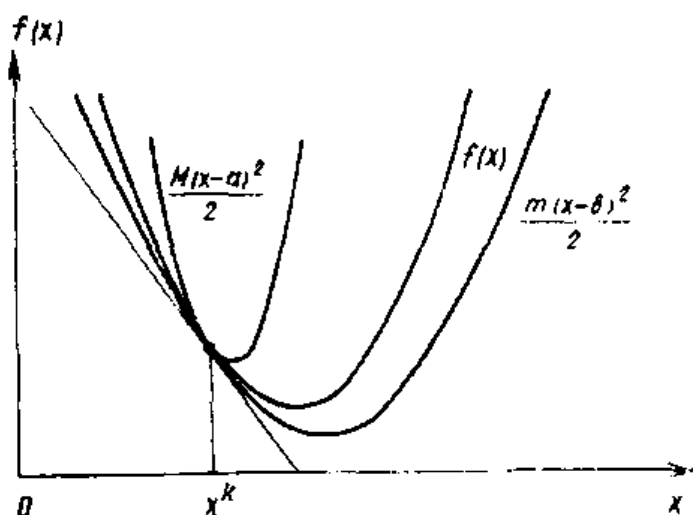


Рис. 1.6

ние самого экстремума. Для того чтобы гарантировать сходимость, в метод Ньютона внесены изменения, а именно, очередной шаг делается в направлении точки  $z = \arg \min \psi(x, x^k) = x^k - \Gamma_{x=x^k}^{-1} \nabla_x f$  не на полную длину, а уменьшенную в  $\alpha_k$  раз, где  $\alpha_k \geq 1$ . Величина  $\alpha_k$  выбирается так, чтобы значение функции  $f(\alpha) = f(x^k + (z - x^k)/\alpha_k)$  обратилось в минимум подобно тому, как это делается в методе «кратчайшего»

спуска. Такая модификация метода Ньютона носит название метода Ньютона — Рафсона.

Имеет место следующая теорема. Пусть  $f(x)$  — дважды дифференцируемая функция — сильно выпукла, а матрица ее вторых производ-

ных удовлетворяет условию Липшица  $\| \Gamma(x) - \Gamma(y) \| < R(x - y)$ , где  $R < \infty$ . Тогда последовательность

$$x^{k+1} = x^k - [V_x^T V_x f(x^k)]^{-1} \nabla_x f(x^k) / \alpha_k$$

сходится к искомой точке минимума независимо от выбора начального приближения.

Достоинством метода Ньютона—Рафсона по сравнению с градиентными методами является гораздо более высокая скорость сходимости, достигаемая благодаря лучшей аппроксимации функции  $f(x)$  в окрестности точки  $x^k$ . Недостатком является, то, что этот метод предъявляет намного более высокие требования к аналитическим свойствам  $f(x)$ . При этом имеет место значительное усложнение вычислительной процедуры. Помимо одномерного поиска, присущего и градиентным методам, здесь требуется произвести вычисление и обращение матрицы вторых производных. Для сокращения объема вычислений, производимых на каждой итерации, применяют различные эвристические приемы, например, вычисляют матрицу вторых производных не на всех шагах, а лишь на некоторых, сохраняя неизменной эту матрицу на протяжении нескольких шагов итерационного процесса. В предельном случае матрицу производных вычисляют всего один раз — в начальной точке  $x^0$ , с которой начинается этот процесс. Формула, по которой ведутся вычисления, имеет вид

$$x^{k+1} = x^k - \frac{1}{\alpha_k} H_k^{-1} \nabla_x f(x^k), \quad (1.12)$$

в которой матрица  $H_k$  уже не совпадает с матрицей вторых производных в точке  $x^k$ . Из-за снижения качества аппроксимации число итераций в такой модификации метода Ньютона—Рафсона увеличивается и, кроме того, могут нарушиться условия, при которых гарантируется сходимость итерационного процесса.

Часто поступают наоборот. Назначив более или менее произвольно  $H_k$  в (1.12) на первых шагах итерационного процесса и используя данные о градиенте, собранные в ходе последующих итераций, добиваются того, что  $H_k \rightarrow \nabla_x^T \nabla_x f(x^k)$ . В результате эти методы переходят в пределе в метод Ньютона — Рафсона. К их числу относится метод Давидова — Флетчера — Пауэлла, который мы будем неоднократно упоминать в дальнейшем при оценке сравнительной трудоемкости метода ВВКМ.

При отыскании корня функции  $f'(x)$  иногда вместо вычисления производных используют их приближенные значения, определяемые с помощью секущих. В одномерном случае этот метод достаточно нагляден (рис. 1.7). Задавшись двумя значениями аргумента  $x^0$  и  $x^1$  такими, что  $f'_0 = f'(x^0) < 0$ , а  $f'_1 = f'(x^1) > 0$ , и проведя через точки  $(x^0, f'_0)$  и  $(x^1, f'_1)$  секущую  $\psi'(x) = f'(x^0) + B(x - x^0)$ , где  $B = (f'_1 - f'_0) /$

$f(x_1 - x_0)$ , получим следующее выражение для приближенного значения искомого корня:

$$x^2 = x^0 + f'_0 (x^1 - x^0) / (f'_1 - f'_0).$$

После этого мы должны отбросить одну из точек  $x^0$  или  $x^1$  (а именно ту, в которой значение  $f'(x)$  противоположно по знаку  $f'(x^2)$ ) и повторить указанную процедуру с точкой  $x^2$  и оставшейся точкой.

Пусть  $f'(x)$  выпукла вниз. Тогда  $\psi'(x) = f'(x^0) + B(x - x^0) \geq f'(x)$  на отрезке  $[x^0, x^1]$ . Интегрируя обе части этого неравенства с начальными условиями  $\psi'(x^0) = f'(x^0)$ , получаем, что на некотором отрезке, включающем  $[x^0, x^1]$ ,

$$\int_{x^0}^x \psi'(x) dx \geq \int_{x^0}^x f'(x) dx.$$

Из последнего соотношения следует

$$f(x^0) + f'(x^0)(x - x^0) + (x - x^0)^2 B / 2 \geq f(x)$$

Таким образом, на отрезке  $[x^0, x^1]$  минимизируемая функция  $f(x)$  мажорируется специальным образом подобранной параболой.

Минимум этой параболы достигается в точке  $x^2$ , лежащей внутри участка мажорирования. В данном случае формулу

$$B = [f'(x^1) - f'(x^0)] / (x^1 - x^0)$$

можно переписать в виде

$$B = \max_{x^0 \leq x \leq x^1} \{(f'(x) - f'(x^0)) / (x - x^0)\}. \quad (1.13)$$

Последняя формула выведена для функций  $f(x)$  с выпуклой вниз производной  $f'(x)$ . Однако ею можно пользоваться и в общем случае, т. е. по отношению к любым монотонно возрастающим  $f'$ . Определив по этой формуле  $B$ , построив прямую

$$\psi'(x) = f'(x^0) + B(x - x^0) \quad (1.14)$$

и параболу

$$\psi(x) = f(x^0) + f'(x^0)(x - x^0) + (x - x^0)^2 B / 2, \quad (1.15)$$

мы гарантируем, что точка пересечения (1.14) с осью абсцисс (и значит минимум параболы (1.15)) лежит внутри отрезка  $[x^0, x^1]$ , причем в этой точке  $f(x) \leq f(x^0)$ . Отметим, что ни в одно из выражений (1.13) - (1.15) значения вторых производных не входят.

Рассмотрев наиболее распространенные методы поиска экстремума, можно заметить, что по мере повышения порядка производных, используемых при расчетах, резко сокращается число итераций, выполняемых в ходе расчетов, но также резко возрастают требования к точности начального приближения и к аналитическим свойствам минимизируемой функции. В частности, для успешного применения метода Ньютона - Рафсона и его модификаций от нее требуется существование первых и вторых производных и так называемая сильная выпуклость.

Для слабо выпуклых функций поиск минимума можно осуществлять на основе субградиентных методов. Однако число итераций в этих методах обычно очень велико и в большинстве случаев сходимость процесса оказывается намного ниже линейной.

## Глава 2.

### МЕТОД ВАРИАЦИОННО-ВЗВЕШЕННЫХ КВАДРАТИЧЕСКИХ МАЖОРАНТ

Как мы видели, условия, при которых гарантируется сходимость метода Ньютона - Рафсона, достаточно жестки: требуется существование и непрерывность первых и вторых производных минимизируемой функции  $f(x)$  и сильная выпуклость этой функции. Вместе с тем, посмотрев на этот процесс внимательнее, можно заметить, что сходимость итерационного процесса, направленного на достижение минимума, обеспечивается, если специальным образом построенный параболоид  $\psi(x, x^k)$ , проходящий через точку  $(x, f(x^k))$ , мажорирует эту функцию, т. е. в некоторой области определения этой функции соблюдается условие  $\psi(x, x^k) \geq f(x)$ , в которое значения вторых производных  $f''(x)$  не входят. Это открывает возможность дальнейшего обобщения метода Ньютона и его применения для минимизации слабо выпуклых функций. Одним из таких обобщений является метод вариационно-взвешенных квадратических мажорант (ВВКМ).

#### 2.1. ОСНОВЫ МЕТОДА ВВКМ

Пусть дана гладкая функция  $f(x)$ , подлежащая минимизации, и параболоид

$$\psi(x, x^k) = A(x^k) - [x - C(x^k)]^T B(x^k) [x - C(x^k)] = \\ A^k - (x - C^k)^T B^k (x - C^k),$$

проходящий через точку  $(x^k, f(x^k))$  и мажорирующий функцию  $f(x)$  так, что

$$\psi(x, x^k)|_{x=x^k} = f(x^k), \quad \psi(x, x^k) \geq f(x).$$

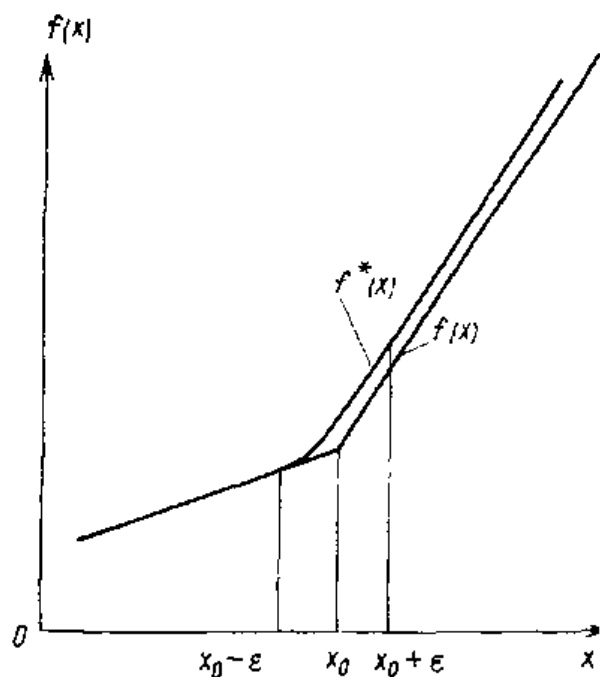


Рис. 2.1

Первым следствием мажорирования является совпадение у  $\psi(x, x^k)$  и  $f(x)$  первых производных в точке  $x = x^k$ , что легко доказывается от противного. Кроме того, найдя минимум  $\psi(x, x^k)$ , по  $x$ , укажем точку

$$x^{k+1} = \arg \min_x \psi(x, x^k),$$

такую, что

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

Поскольку  $\psi(x, x^k)$  мажорирует  $f(x)$ , то  $\psi(x^{k+1}, x^k) \geq f(x^{k+1})$  и, таким образом,  $f(x^{k+1}) \leq f(x^k)$ .

Как будет показано в дальнейшем, случай равенства является исключительным. Он свидетельствует об окончании итерационного процесса и до-

стижении стационарной точки. Если  $f(x)$  выпукла вниз (хотя бы и не строго), то полученная точка принадлежит к множеству точек, доставляющих абсолютный минимум функции  $f(x)$ .

В одномерном случае для существования такой мажоранты от функции  $f(x)$  требуется лишь гладкость и нестрогая выпуклость. На  $f''(x)$  не налагается никаких условий, кроме существования и ограниченности сверху:  $0 \leq f''(x) < D < \infty$ .

Перечисленные условия можно ослабить и дальше. Во-первых, если  $f(x)$  непрерывна и выпукла, но не всюду гладка, а имеет конечное число точек разрыва производной  $f'(x)$ , то всегда можно осуществить малое изменение функции  $f(x)$ , после которого она будет удовлетворять требуемым условиям гладкости, выпуклости и ограниченности сверху по второй производной.

В самом деле, пусть  $x_0$  — точка, в которой нарушается непрерывность первой производной, так что

$$\lim_{t \rightarrow 0} (f'(x_0 + t) - f'(x_0 - t)) = k \neq 0$$

(в силу выпуклости  $f(x)$  имеем  $k > 0$ ). Окружим эту точку отрезком  $[x_0 - \epsilon, x_0 + \epsilon]$ , где  $\epsilon$  — произвольно малое положительное число, и построим параболу (рис. 2.1)

$$\psi(x) = A + (x - C)^2 B/2,$$

удовлетворяющую следующим условиям:

$$\psi(x_0 - \epsilon) = f(x_0 - \epsilon),$$

$$\psi'(x_0 - \epsilon) = f'(x_0 - \epsilon),$$

$$\psi'(x_0 + \epsilon) = f'(x_0 + \epsilon).$$

Отсюда немедленно получаем:

$$B = [f'(x_0 + \varepsilon) - f'(x_0 - \varepsilon)] / (2\varepsilon), \quad (2.1)$$

$$C = x_0 - \varepsilon - f'(x_0 - \varepsilon) / B, \quad (2.2)$$

$$A = f(x_0 - \varepsilon) - [f'(x_0 - \varepsilon)]^2 / (2B). \quad (2.3)$$

Рассмотрим функцию  $f^*(x)$ , мажорирующую  $f(x)$  и построенную следующим образом:

$$f^*(x) = \begin{cases} f(x), & \text{если } x \leq x_0 - \varepsilon; \\ \psi(x), & \text{если } x_0 - \varepsilon \leq x \leq x_0 + \varepsilon; \\ f(x) + [\psi(x_0 + \varepsilon) - f(x_0 + \varepsilon)], & \text{если } x_0 + \varepsilon \leq x. \end{cases}$$

Разность

$$f^*(x) - f(x) = \begin{cases} 0, & \text{если } x \leq x_0 - \varepsilon; \\ \psi(x) - f(x), & \text{если } x_0 - \varepsilon \leq x \leq x_0 + \varepsilon; \\ \psi(x_0 + \varepsilon) - f(x_0 + \varepsilon), & \text{если } x_0 + \varepsilon \leq x. \end{cases}$$

имеет тот же порядок, что и  $\varepsilon$ . В самом деле, пусть  $x_0$  находится на участке, где  $f'(x_0 - \varepsilon) \geq 0$  и  $f'(x_0 + \varepsilon) > 0$ . В этом случае

$$\begin{aligned} f^*(x) - f(x) &\leq \max_{x_0 - \varepsilon \leq x \leq x_0 + \varepsilon} (\psi(x) - f(x)) \leq \psi(x_0 + \varepsilon) - f(x_0 - \varepsilon) = \\ &= f(x_0 - \varepsilon) - \frac{[f'(x_0 - \varepsilon)]^2}{2B} + \frac{B}{2} \left[ x_0 + \varepsilon - x_0 + \varepsilon + \frac{f'(x_0 - \varepsilon)}{B} \right]^2 - \\ &\quad - f(x_0 - \varepsilon) = \frac{B}{2} \left[ 2\varepsilon + \frac{f'(x_0 - \varepsilon)}{B} \right]^2 - \frac{[f'(x_0 - \varepsilon)]^2}{2B} = \\ &= \frac{B}{2} (2\varepsilon)^2 + \frac{B}{2} \cdot 4 \cdot \frac{\varepsilon f'(x_0 - \varepsilon)}{B} = \varepsilon [f'(x_0 + \varepsilon) - f'(x_0 - \varepsilon)] + \\ &\quad + 2\varepsilon f(x_0 - \varepsilon) = \varepsilon [f'(x_0 + \varepsilon) + f'(x_0 - \varepsilon)] + \\ &\quad + 2\varepsilon f(x_0 - \varepsilon). \end{aligned}$$

Аналогично доказывается малость  $f^*(x) - f(x)$  при другом расположении точки разрыва производной, т. е. при условиях, что  $f'(x_0 - \varepsilon) < 0$  и  $f'(x_0 + \varepsilon) \leq 0$  или что  $f'(x_0 - \varepsilon) \leq 0$  и  $f'(x_0 + \varepsilon) \geq 0$ .

В данном случае величина  $B$  зависит от  $\varepsilon$ .

Можно поступить наоборот: задавшись достаточно большим  $B = B$ , с помощью (2.1) указать  $\varepsilon$ -окрестность каждой точки разрыва производной, внутри которой следует осуществлять описанную выше корректировку. При попадании  $x^k$  в эту окрестность мажорирующей параболой является

$$\psi(x, x^k) = A + (x - C)^2 \tilde{B} / 2,$$

где  $A$  и  $C$  определяются формулами (2.2) и (2.3).

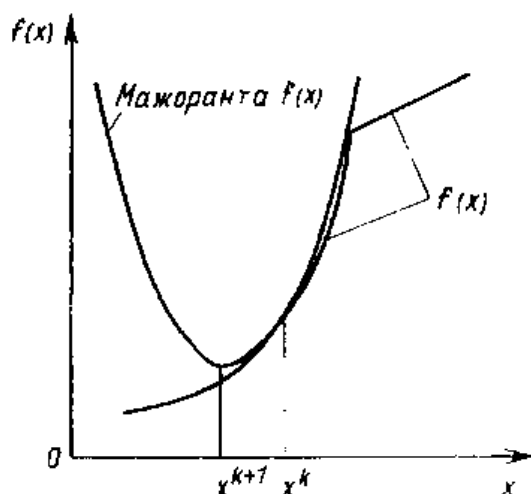


Рис. 2.2

Кроме того существование мажоранты, касающейся в данной точке поверхности  $z = f(x)$ , вообще говоря, не предполагает ее выпуклости (рис. 2.2). Однако учитывая, что в дальнейшем мы будем иметь дело исключительно с суммами вида  $\Phi = \sum_i f_i(x)$ , для обеспечения уни-  
 модальности суммы предположим, что каждое слагаемое выпукло вниз. Известно, что сумма выпуклых вниз функций сама выпукла вниз. Это гарантирует односвязанность минимума.

Эти замечания о возможном ослаблении требований к аналитическим свойствам функции  $f(x)$  остаются в силе и тогда, когда точка  $x$  является точкой многомерного пространства.

Для некоторых простейших функций  $f_i(x)$  построение мажорирующей параболы, проходящей через точку  $[x^k, f_i(x^k)]$  и касающейся ее в этой точке, представляет элементарную задачу. Однако если взять сумму таких функций даже не с очень большим числом слагаемых, то для нее построить такую мажоранту уже весьма непросто. Но если учесть, что сумма мажорант отдельных слагаемых является мажорантой суммы, то получим регулярный способ отыскания минимума достаточно широкого класса сложных функций с плохими аналитическими свойствами.

Например, нетрудно указать способ построения квадратических мажорант для функций, показанных на рис. 2.3. Труднее указать мажоранту для их суммы (рис. 2.4). Но опираясь на ранее сделанное замечание о том, что сумма мажорант является мажорантой суммы, мы можем без труда мажорировать и эту сложную функцию.

Достижение минимума гарантируется следующими рассуждениями. Мы условились, что мажорирующая парабола (в многомерном слу-

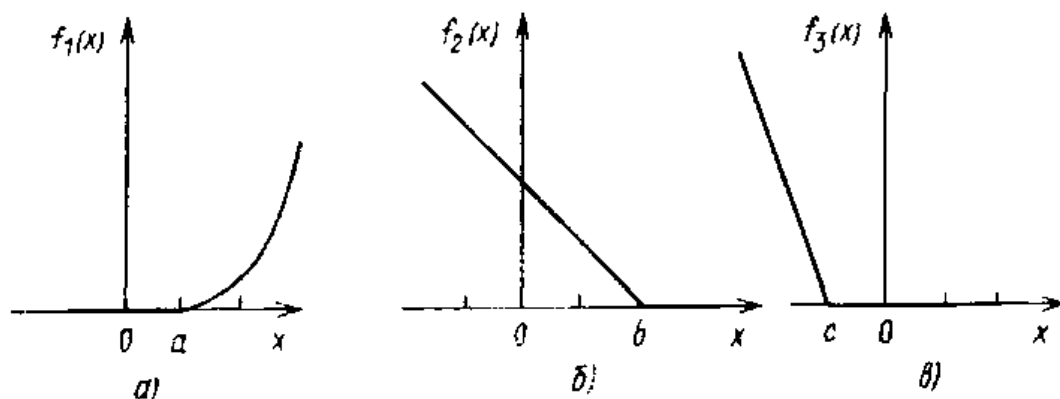


Рис. 2.3

чае параболоид) проводится так, чтобы соблюдались условия

$$\psi(x, x^k) = f(x) \big|_{x=x^k}, \quad (2.4)$$

$$\nabla_x \psi(x, x^k) = \nabla_x f(x) \big|_{x=x^k}, \quad (2.5)$$

$$\psi(x, x^k) \geq f(x). \quad (2.6)$$

Пусть в некоторой точке процесс итераций закончился и имеет место равенство  $x^{k+1} = x^k = x^*$ . Учитывая, что

$$x^{k+1} = x^k - \Gamma^{-1} \nabla_x \psi,$$

где  $\Gamma = (\nabla_x \nabla_x^T \psi(x, x^k)) \neq 0$ , мы в этом случае должны принять, что  $\nabla_x \psi(x, x^k) \big|_{x=x^*} = 0$ , а учитывая (2.5), — что  $\nabla_x f(x) \big|_{x=x^*} = 0$ . Это по отношению к гладким выпуклым вниз функциям является необходимым и достаточным условием минимума.

Как уже говорилось, метод ВВКМ можно считать одной из модификаций метода Ньютона. Однако при этом нужно иметь в виду, что в методе Ньютона достаточно гладкая функция *аппроксимируется* параболоидом второй степени, а в методе ВВКМ минимизируемая в общем случае функция *мажорируется* таким параболоидом \*).

## 2.2. ПРИЕМЫ МАЖОРИРОВАНИЯ

Перейдем к указанию мажорирующих парабол (параболоидов) для ряда функций, которые в дальнейшем придется рассматривать в качестве слагаемых в суммах вида  $\sum_i f_i(x)$  при решении прикладных задач.

**Пример 2.1.** Задана функция

$$f(x) = \begin{cases} 0, & \text{если } x \leq 0, \\ x^2/2, & \text{если } x \geq 0, \end{cases}$$

график которой представлен на рис. 2.5.

Рассмотрим отдельно два случая.

1)  $x^k < 0$ : мажорирующая парабола  $\psi(x, x^k) = (x - x^k)^2/2$ ;

2)  $x^k \geq 0$ : мажорирующая парабола  $\psi(x, x^k) = x^2/2$ .

В целом

$$\psi(x, x^k) = \begin{cases} (x - x^k)^2/2, & \text{если } x^k < 0, \\ x^2/2, & \text{если } x^k \geq 0. \end{cases} \quad (2.7)$$

\*) В общем случае мажорантой может быть и не параболоид. Например, в гл. 4 при решении задач шкалирования мажорирование осуществляется в два этапа. на первом этапе в качестве мажоранты выступает выпуклый вниз многогранник и лишь на втором — параболоид

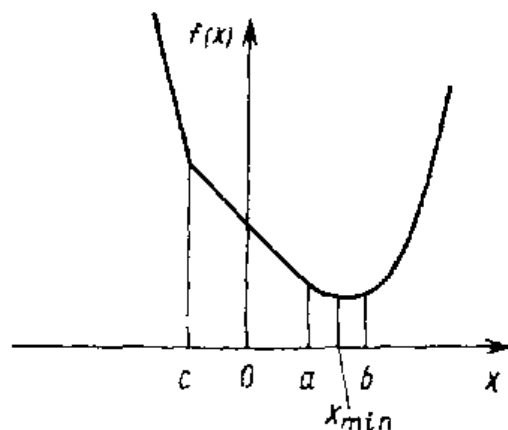


Рис. 2.4

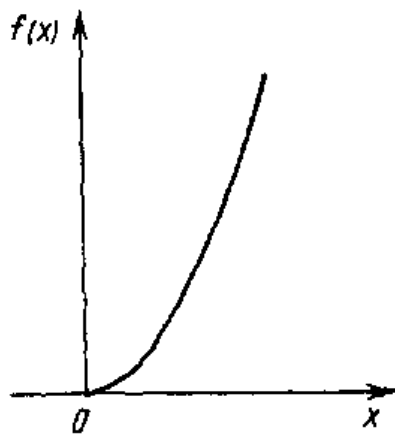


Рис. 2.5

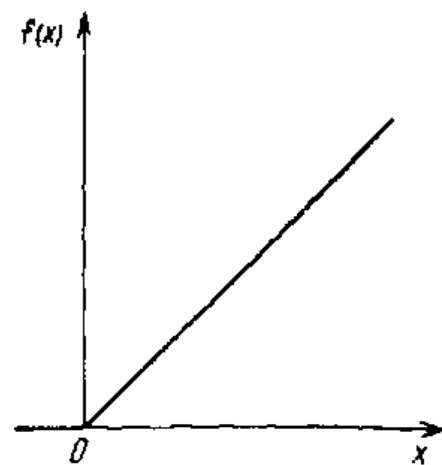


Рис. 2.6

**Пример 2.2.** Задана функция

$$f(x) = \begin{cases} 0, & \text{если } x < 0, \\ x, & \text{если } x \geq 0, \end{cases}$$

график которой показан на рис. 2.6.

Здесь рассмотрим три случая.

1. При  $x^k < 0$  мажорирующую параболу проведем так, чтобы она наряду с обязательными условиями

$$\psi(x^k, x^k) = f(x^k), \quad \psi'(x, x^k)|_{x=x^k} = f'(x^k)$$

удовлетворяла бы дополнительным условиям

$$\psi(\hat{x}, x^k) = f(\hat{x}), \quad \psi'(x, x^k)|_{x=\hat{x}} = f'(\hat{x}),$$

где  $\hat{x} > 0$  (рис. 2.7), гарантирующим соблюдение третьего обязательного условия

$$\psi(x, x^k) \geq f(x).$$

Для определения коэффициентов мажорирующей параболы

$$\psi(x) = A + (x - C)^2 B/2$$

и точки  $\hat{x}$  имеем четыре условия:

$$\psi(x) = f(x)|_{x=x^k},$$

$$\psi'(x) = f'(x)|_{x=x^k},$$

$$\psi(x) = f(x)|_{x=\hat{x}},$$

$$\psi'(x) = f'(x)|_{x=\hat{x}},$$

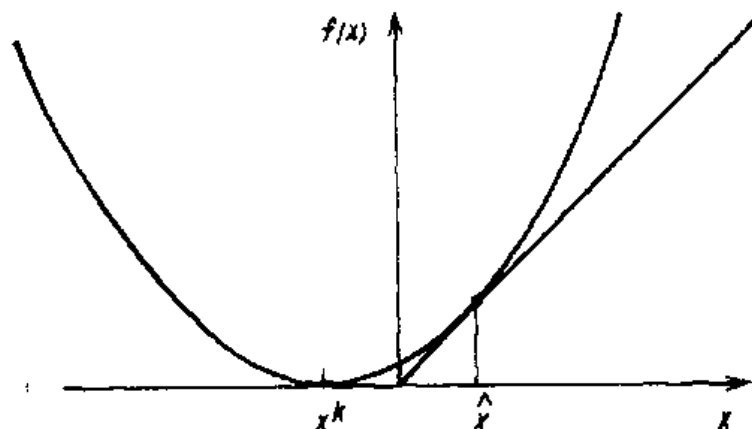


Рис. 2.7

что с учетом конкретного вида функции  $f(x)$  дает

$$A + (x^k - C)^2 B/2 = 0, \quad A + (\hat{x} - C)^2 B/2 = \hat{x},$$

$$B(x^k - C) = 0, \quad B(\hat{x} - C) = 1.$$

Отсюда после несложных преобразований получаем

$$\hat{x} = -x^k, \quad B = (2|x^k|)^{-1}, \quad C = x^k, \quad A = 0.$$

В дальнейшем значение  $A$  интересоваться не будет, так как оно не сказывается на положении минимума мажорирующей параболы и, следовательно, на значении очередного приближения  $x^{k+1}$ .

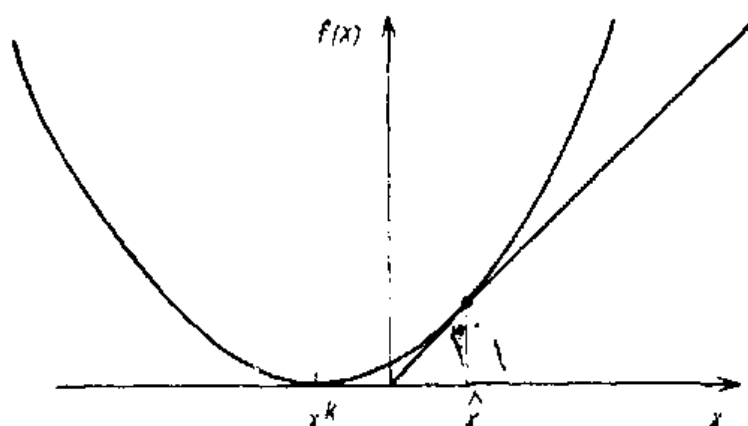


Рис. 2.8

2. При  $x^k > 0$  мажоранту будем проводить, исходя из следующих условий (рис. 2.8):

$$\psi(x) = f(x)|_{x=x^k}, \quad \psi'(x) = f'(x)|_{x=x^k}, \quad \min \psi(x) = 0.$$

Имеем  $A + (x^k - C)^2 B/2 = x^k$ ,  $B(x^k - C) = 1$ ,  $A = 0$ , откуда  $C = -x^k$ ,  $B = (2|x^k|)^{-1}$ .

3. При  $x^k = 0$  на основании ранее сделанных замечаний в случае попадания  $x^k$  в малую окрестность точки  $x_0$  разрыва производной полагаем  $A = 0$ ,  $B = \tilde{B}$  (заранее выбранное большое число),  $C = -\epsilon$ . Критерием того, что мы попали в эту малую окрестность служит  $B = (2|x^k|)^{-1} > (2\epsilon)^{-1}$  при попытках построить мажорирующую параболу в случаях 1 и 2.

Объединив результаты, можно воспользоваться следующими формулами:

$$C = -|x^k|, \quad B = [2(|x^k| + \epsilon)]^{-1},$$

где  $\epsilon$  — произвольно малое положительное число.

Этот же результат можно получить и иначе, используя формулу (1.13). Видоизменив функцию  $f(x)$  так, как рекомендуется в § 2.1, получим, что

$$f^{*'}(x) = \begin{cases} 0, & \text{если } x \leq -\varepsilon; \\ (x + \varepsilon)/(2\varepsilon), & \text{если } -\varepsilon \leq x \leq \varepsilon; \\ 1, & \text{если } x \geq \varepsilon. \end{cases}$$

Полагая, что область возможных положений точки минимума есть  $R = \{x: -\infty < x < +\infty\}$  по формуле

$$B = \max_{x \in R} \{|f^{*'}(x) - f^*(x^0)|/(x - x^0)\}$$

находим, что

$$B = \begin{cases} |2(|x^0| + \varepsilon)|^{-1}, & \text{если } x^0 \geq \varepsilon; \\ (2\varepsilon)^{-1}, & \text{если } -\varepsilon \leq x^0 \leq \varepsilon; \\ |2(-|x^0| + \varepsilon)|^{-1}, & \text{если } x^0 \leq -\varepsilon. \end{cases}$$

Отсюда следует, что

$$C = x^0 - f'(x^0)/B \approx \begin{cases} x^0, & \text{если } x^0 \leq -\varepsilon; \\ 0, & \text{если } -\varepsilon \leq x^0 \leq +\varepsilon; \\ -x^0, & \text{если } x^0 \geq \varepsilon. \end{cases}$$

Коэффициент  $A$  в ходе итерационного процесса не используется\*).

**Пример 2.3.** Задана функция

$$f(x_1, x_2, \dots, x_n) = \begin{cases} \sum_{j=1}^n a_j x_j + b, & \text{если } \sum_{j=1}^n a_j x_j + b \geq 0; \\ 0 & \text{в противоположном случае.} \end{cases}$$

Обозначив  $y = \sum_{j=1}^n a_j x_j + b$ , приходим к задаче построения мажоранты для только что рассмотренной функции и, таким образом,

$$\Psi(x_1, x_2, \dots, x_n) = A + \left[ \sum_{j=1}^n (a_j x_j + b) - C \right]^2 B/2,$$

где

$$A = 0, \quad B = \left( 2 \left| \sum_{j=1}^n a_j x_j^0 + b \right| + 2\varepsilon \right)^{-1}, \quad C = - \left| \sum_{j=1}^n a_j x_j^0 + b \right|.$$

\*) Отметим, что в силу теоремы Харди — Литлвуда — Поля [18, с. 451] любая выпуклая вниз функция  $f(x)$  на конечном интервале может быть равномерно приближена суммой  $f(x) = \sum_{i=1}^n f_i(x)$ , где  $f_i(x)$  либо имеет вид, рассмотренный в примере 2.2, либо линейна.

Вообще во всех случаях, когда удастся построить мажоранту для функций  $f(y)$ ,  $f(y, z)$  и т. д. от небольшого числа аргументов, мы автоматически получаем квадратичные мажоранты и для следующих функций от многих переменных  $x_1, x_2, \dots, x_n$ :

$$f(\sum a_j x_j + b), f(\sum a_{1j} x_j + b_1; \sum a_{2j} x_j + b_2) \text{ и т. д.}$$

**Пример 2.4.** Задана функция  $f(x) = |x|$  (рис. 2.9). Представим ее в виде

$$f(x) = f_1(x) + f_1(-x), \quad (2.8)$$

где

$$f_1(x) = \begin{cases} 0, & \text{если } x < 0; \\ x, & \text{если } x \geq 0. \end{cases}$$

Первое слагаемое в (2.8) мажорируется параболой

$$\psi_1(x, x^k) = (-x + |x^k|)^2 / [4(|x^k| + \epsilon)],$$

а второе — параболой

$$\psi_2(x, x^k) = (x + |x^k|)^2 / [4(|x^k| + \epsilon)].$$

Сама же  $f(x)$  мажорируется параболой

$$\psi(x, x^k) = \psi_1(x, x^k) + \psi_1(-x, -x^k) = (x^k)^2 / 2 + x^2 / [2(|x^k| + \epsilon)],$$

т. е. параболой, у которой

$$B(x^k) = [2(|x^k| + \epsilon)]^{-1}, \quad C(x^k) = 0.$$

Коэффициент  $A$ , как мы уже отмечали, нас не интересует. При  $x^k = 0$  полагаем  $B = \tilde{B} = (2\epsilon)^{-1}$ .

**Пример 2.5.** Функции, рассмотренные в примерах 2.1 и 2.2, являются частными случаями функции более общего вида

$$f(x) = \begin{cases} x^{1+\alpha}, & \text{если } x \geq 0, \\ 0, & \text{если } x < 0. \end{cases} \quad (0 \leq \alpha \leq 1).$$

При  $\alpha = 1$  получается функция, описанная в примере 2.1, при  $\alpha = 0$  — в примере 2.2.

Как и прежде, рассмотрим три случая.

1. При  $x^k < 0$  уравнения для определения коэффициентов мажорирующей параболы  $\psi(x) = A + (x^k - C)^2 B / 2$ , вытекающие из условий (2.4) — (2.6) и аналогичные условиям (2.7), выглядят так:

$$\begin{aligned} A + (x^k - C)^2 B / 2 &= 0, \\ B(x^k - C) &= 0, \\ A + (\hat{x} - C)^2 B / 2 &= \hat{x}^{1+\alpha}, \\ B(\hat{x} - C) &= (1 + \alpha) \hat{x}^\alpha. \end{aligned} \quad (2.9)$$

Неизвестными являются  $A, B, C$  и  $\hat{x}$ , причем  $A$  и  $\hat{x}$  интересуют нас лишь как промежуточные величины при определении  $B$  и  $C$ .

Разрешая первые два уравнения (2.9) относительно  $A$  и  $C$ , получаем  $C = x^k$ ,  $A = 0$ . После подстановки этих значений в третье и четвертое уравнения (2.9) и деления их друг на друга, найдем, что

$$(\hat{x} - x^k)/2 = \hat{x}/(1 + \alpha);$$

отсюда

$$\hat{x} = \frac{\alpha + 1}{\alpha - 1} x^k, \quad B = \frac{(1 + \alpha) \hat{x}^\alpha}{(\hat{x} - x^k)}. \quad (2.10)$$

Разумеется, это не единственный способ построения мажорирующей функции.

2. При  $x^k > 0$  условия (2.4) — (2.6) дают

$$A + (x^k - C)^2 B/2 = (x^k)^{1+\alpha},$$

$$B(x^k - C) = (x^k)^\alpha, \quad A = 0,$$

откуда непосредственно получаем

$$C = -x^k, \quad B = |x^k|^{\alpha-1}/2 = (2|x^k|^{1-\alpha})^{-1}. \quad (2.11)$$

Как и следовало ожидать, формулы (2.10) превращаются в формулы примера 2.1 при  $\alpha = 1$  и в формулы примера 2.2 при  $\alpha = 0$ .

3. При  $x^k = 0$ , как и всегда в таких случаях, в качестве мажоранты примем параболу  $\psi(x) = B(x - C)^2$ , где  $B = \hat{B}$ , а  $C = -\epsilon$ .

Парабола с коэффициентами, определяемыми по (2.10), (2.11), не единственный вид мажорирующей параболы. Например, на интервале  $(-\infty, -1)$ , другой, более простой по конструкции коэффициентов, но несколько хуже аппроксимирующей функцию  $f(x)$  параболой является парабола

$$\psi(x, x^k) = A + (x - x^k)^2. \quad (2.12)$$

Нетрудно убедиться, что если  $x^k$  лежит в этом интервале, парабола (2.12) удовлетворяет условиям (2.4) — (2.6), которые мы предъявили к мажорирующей параболе.

**Пример 2.6.** Задана функция

$$f(x) = \begin{cases} -\ln(2 - e^x), & \text{если } x \leq 0; \\ x, & \text{если } x > 0, \end{cases}$$

график которой показан на рис. 2.10.

Рассмотрим два случая.

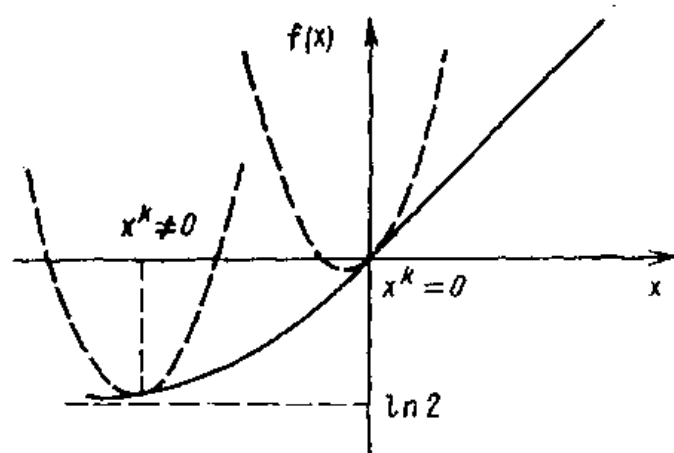
1. При  $x^k > 0$  один из способов построения мажорирующей параболы предложен в примере 2.2, п. 2. Он приводит к следующим значениям коэффициентов:

$$A(x^k) = 0, \quad B(x^k) = (2x^k)^{-1}, \quad C(x^k) = -x^k.$$

2. При  $x^k \leq 0$  вначале покажем, что в диапазоне  $-\infty \leq x \leq \infty$  функция  $f'(x)$  непрерывна, а в диапазоне  $-\infty \leq x \leq 0$  непрерывна, монотонно возрастает и достигает максимума в точке  $x = 0$ . Поэтому парабола  $\psi(x)$ , проходящая через любую точку  $x^k < 0$ , касающаяся в этой точке и имеющая кривизну  $B = f''(0)$ , будет круче уходить вверх, чем функция  $f(x)$  (рис. 2.11). В силу этого производная функция  $\psi(x, x^k)$  в точке  $\hat{x}$ , соответствующей пересечению  $\psi(x, x^k)$  с осью абсцисс, будет больше 1, что гарантирует отсутствие точек пересечения  $\psi(x, x^k)$  с  $f(x)$  в положительной области  $x > 0$ .

В самом деле, в диапазоне  $-\infty < x \leq 0$

$$\begin{aligned} f(x) &= -\ln(2 - e^x), \\ f'(x) &= 1/(2e^{-x} - 1), \\ f''(x) &= 2e^x/(2 - e^x)^2, \\ f'_{\max} &= f''(0) = 2. \end{aligned}$$



Пусть  $\psi(x, x^k) = A + (x - C)^2 B/2$ . Полагая  $\psi(x^k) = f(x^k)$ ,  $\psi_k = B(x^k - C) = f'(x^k)$  и  $\psi'' = B = 2$ , мы гарантируем мажорирование функции  $f(x)$  этой

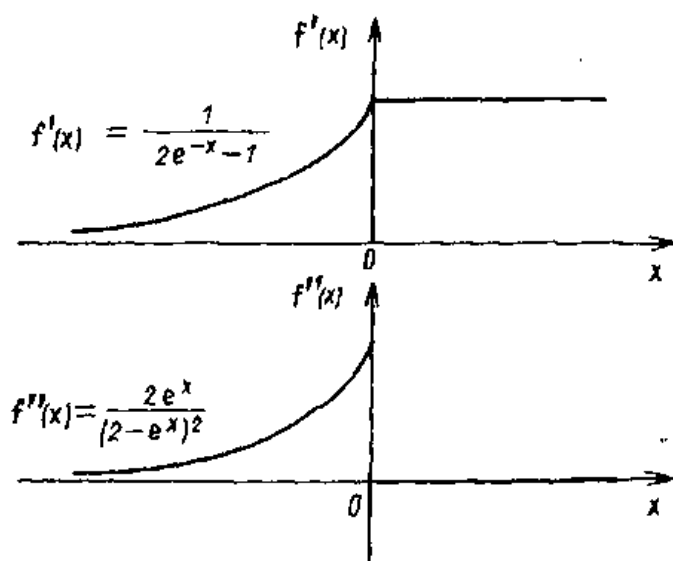
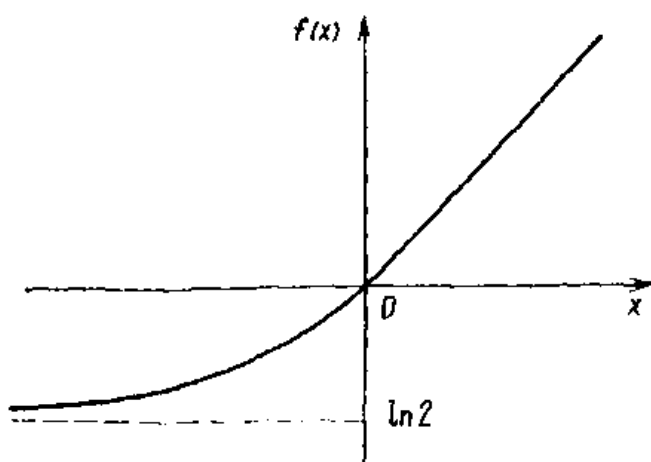


Рис. 2.10

Рис. 2.11

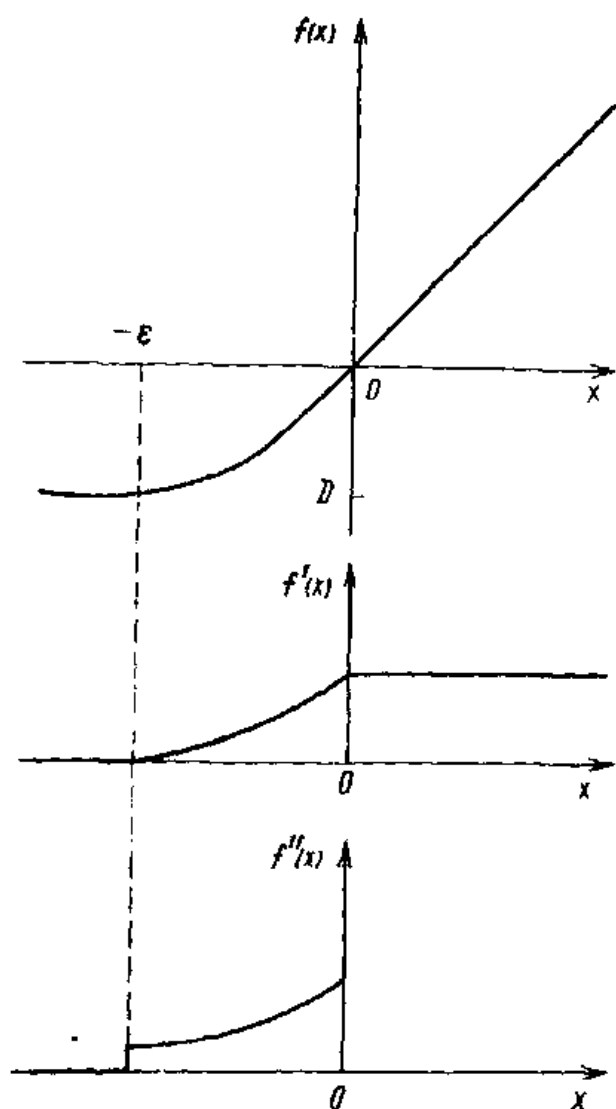


Рис. 2.12

**Пример 2.7.** Задана функция

$$f(x) = \begin{cases} x & , \text{ если } x \geq 0; \\ \log \frac{1 - \exp(-\varepsilon) \operatorname{ch}(x + \varepsilon)}{\operatorname{sh} \varepsilon \exp(-\varepsilon)} & , \text{ если } -\varepsilon \leq x < 0; \\ D & , \text{ если } x < -\varepsilon, \end{cases}$$

где

$$D = -\log \frac{1 - \exp(-\varepsilon)}{\exp(-\varepsilon) \operatorname{sh} \varepsilon} = -\log 2 \frac{1 - \exp(-\varepsilon)}{1 - \exp(-2\varepsilon)}.$$

Графики этой функции и ее производных изображены на рис. 2.12, а ее происхождение будет объяснено в гл. 4.

При построении мажоранты рассмотрим три случая.

1. Для случая  $x > 0$  пригодна мажоранта, построенная в п. 2 примера 2.2, т. е. парабола  $\psi(x) = A + (x - C)^2 B/2$ , где  $B = (2x^k)^{-1}$ , а  $C = -x^k$ .

2. Для  $-\varepsilon \leq x \leq 0$ , так же как в п. 2 примера 2.6, покажем, что первая и вторая производные функции  $f(x)$  монотонно возрастают.

параболой. Условие  $B = 2$  дает возможность определить неизвестные коэффициенты  $A$  и  $C$ . В частности,

$$C = x^k - \frac{\exp x^k}{2(2 - \exp x^k)}.$$

Аналогичным образом можно установить, что при  $\beta > 0$  функция

$$\begin{aligned} f(x) &= \\ &= \begin{cases} -\beta \log [2 - \exp(x/\beta)], & \text{если } x \leq 0; \\ x, & \text{если } x > 0, \end{cases} \end{aligned}$$

в диапазоне  $(-\infty, 0)$  мажорируется параболой  $\psi(x) = A + (x - C)^2 B/2$  со следующими коэффициентами:

$$B = \frac{2}{\beta},$$

$$C = x^k - \frac{\beta}{2} \frac{\exp(x^k/\beta)}{2 - \exp(x^k/\beta)}.$$

Заметим, что мажорирующая парабола будет несколько лучше аппроксимировать функцию  $f(x)$ , если воспользоваться формулой (1.13).

Поэтому если в качестве значения  $B$  в любой точке  $x^k$  рассматриваемого диапазона взять величину  $f''_{\max} = f''(0)$ , то парабола  $\psi(x) = A + (x - C)^2 B/2$ , проходящая через эту точку и имеющая с ней общую касательную, будет более круто уходить вверх, чем функция  $f(x)$ . В результате точка  $\hat{x}$  ее пересечения с осью абсцисс будет расположена левее начала координат, а касательная к  $\psi(x)$  в этой точке — отклонена от горизонтали сильнее, чем касательная к  $f(x)$  в начале координат. Это гарантирует соблюдение условия  $\psi^k(x) \geq f(x)$  во всем диапазоне изменения переменной  $x$ . В диапазоне  $-\varepsilon \leq x \leq 0$  имеем

$$\begin{aligned} f(x) &= -\log \{1 - \exp(-\varepsilon) \operatorname{ch}(x + \varepsilon)\} + \log [\exp(-\varepsilon) \operatorname{sh} \varepsilon], \\ f'(x) &= \frac{\operatorname{sh}(x + \varepsilon)}{\exp \varepsilon - \operatorname{ch}(x + \varepsilon)}, \\ f''(x) &= \frac{\exp \varepsilon \operatorname{ch}(x + \varepsilon) - 1}{[\exp \varepsilon - \operatorname{ch}(x + \varepsilon)]^2}. \end{aligned} \quad (2.13)$$

С ростом  $x$  в диапазоне  $-\varepsilon \leq x \leq 0$  функция  $\operatorname{ch}(x + \varepsilon)$  монотонно возрастает от 1 до  $\operatorname{ch} \varepsilon$ . Следовательно, числитель дроби в (2.13) монотонно возрастает в этом диапазоне, а ее знаменатель монотонно убывает. Поэтому

$$f''_{\max} = f''(0) = \frac{\exp \varepsilon \operatorname{ch} \varepsilon - 1}{[\exp \varepsilon - \operatorname{ch} \varepsilon]^2} = \frac{2}{1 - \exp(-2\varepsilon)},$$

откуда

$$B = \frac{2}{1 - \exp(-2\varepsilon)}, \quad C = x^k - \frac{1}{B} \frac{\operatorname{sh}(x^k + \varepsilon)}{\exp \varepsilon - \operatorname{ch}(x^k + \varepsilon)}.$$

Совершенно аналогичным образом устанавливаем, что если

$$f(x) = \begin{cases} x & , \text{ если } x > 0; \\ -\beta \log \frac{1 - \exp(-\varepsilon/\beta) \operatorname{ch}[(x + \varepsilon)/\beta]}{\exp(-\varepsilon/\beta) \operatorname{ch}(\varepsilon/\beta)} & , \text{ если } -\varepsilon \leq x \leq 0; \\ D & , \text{ если } x \leq -\varepsilon, \end{cases}$$

где

$$D = -\beta \log \frac{1 - \exp(-\varepsilon/\beta)}{\exp(-\varepsilon/\beta) \operatorname{sh}(\varepsilon/\beta)} = -\beta \log 2 \frac{1 - \exp(-\varepsilon/\beta)}{1 - \exp(-2\varepsilon/\beta)},$$

то в качестве коэффициентов мажорирующей параболы в этом диапазоне могут быть приняты

$$\begin{aligned} B &= \frac{2}{\beta} \frac{1}{1 - \exp(-2\varepsilon/\beta)}, \\ C &= x^k - \frac{1}{B} \frac{\operatorname{sh}[(x^k + \varepsilon)/\beta]}{\exp(\varepsilon/\beta) - \operatorname{ch}[(x^k + \varepsilon)/\beta]}. \end{aligned}$$

3. При  $x^k < -\epsilon$  в качестве мажоранты может быть принята функция  $\psi^k(x) = A + (x - x^k)^2 B/2$ , где

$$B = 2/[\beta(1 - e^{-2\epsilon/\beta})].$$

Несколько лучшего качества мажорирования можно добиться, если исходить из следующих соображений. Нетрудно показать, что, во-первых,  $|D| < \epsilon/2$ , а во-вторых, что  $f'(x) < (x + \epsilon - |D|)/[2(\epsilon - |D|)]$  во всем рассматриваемом диапазоне. Это означает (рис. 2.13), что парабола  $\psi^k(x) = A + (x - x^k)^2 B/2$ , касающаяся в точках  $x^k$  и  $\hat{x}$ , одна из которых расположена левее точки  $x = -\epsilon$ , а другая правее

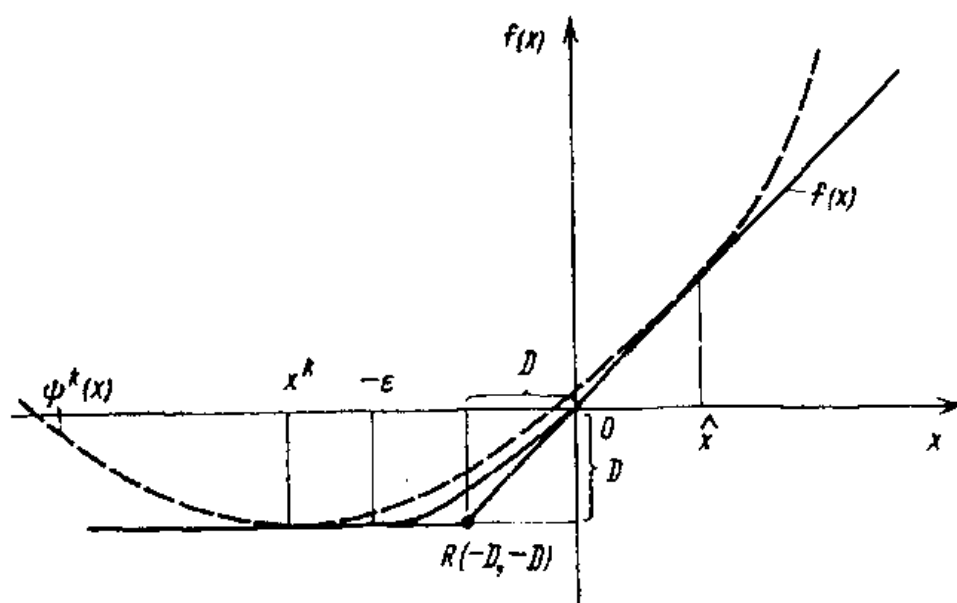


Рис. 2.13

точки  $x = 0$ , нигде не пересечет  $f(x)$ . Перенеся начало координат в точку  $R(-D, -D)$ , мы попадаем в условия примера 2.2 (п.1). Следовательно, в рассматриваемом случае

$$B = (2|x^k + D|)^{-1}, \quad C = x^k,$$

а  $A^k$ , как уже говорилось ранее, нас не интересует.

**Пример 2.8.** Задана функция

$$f(x_1, x_2) = |r|,$$

где  $|r| = \sqrt{(x_1 - a_1)^2 + (x_2 - a_2)^2}$ .

В соответствии с результатами примеров 2.3 и 2.4 эта функция мажорируется параболоидом

$$\begin{aligned} \psi^k(x_1, x_2) &= A + |r|^2 B/2 = \\ &= A + [(x_1 - a_1)^2 + (x_2 - a_2)^2] / (2|\sqrt{(x_1^k - a_1)^2 + (x_2^k - a_2)^2}|). \end{aligned}$$

Аналогично этому функция  $f(x_1, x_2) = |r|^m$  ( $1 \leq m \leq 2$ ) мажорируется параболоидом  $\psi(x_1, x_2) = A + |r|^2 B/2$ , проходящим через заданную точку  $(x_1^0, x_2^0)$  и касающимся поверхности  $f = |r|^m$  в этой точке, если величину  $B$  выбрать на основании условия

$$\frac{d}{d|r|} \psi = \frac{d}{d|r|} f \Big|_{|r|=|r^0|} \quad \text{или} \quad B|r^0| = m|r^0|^{m-1},$$

что дает  $B = m|r^0|^{m-2}$ , где  $r^0 = \sqrt{(x_1^0 - a_1)^2 + (x_2^0 - a_2)^2}$ .

**Пример 2.9.** Дана функция

$$f(x) = \begin{cases} x^2 \beta/2 & , \text{ если } |x| \leq 1; \\ \beta/2 + \beta|x-1|, & \text{ если } |x| > 1, \end{cases}$$

график которой показан на рис. 2.14, а происхождение будет объяснено в гл. 4.

Как нетрудно понять из предыдущих примеров, в данном случае при  $|x^k| \leq 1$  коэффициентами мажорирующей параболы являются  $B = \beta$ ,  $C = 0$ , а при  $|x^k| \geq 1$  имеем  $B = \beta/|x^k|$ ,  $C = 0$ .

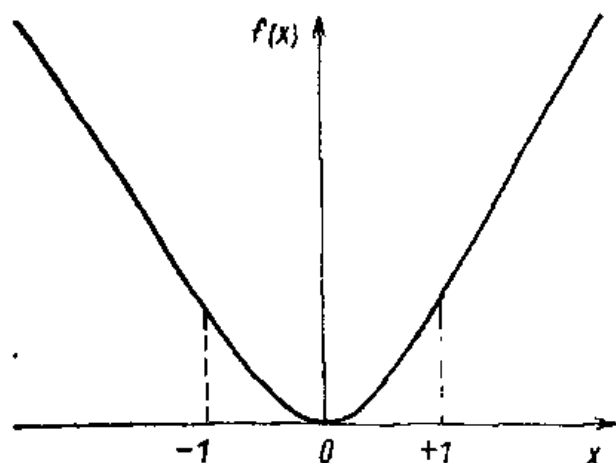


Рис. 2.14

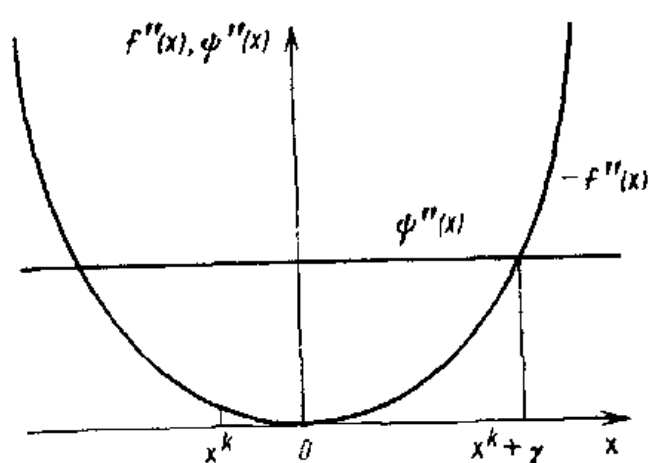


Рис. 2.15

**Пример 2.10.** Задана функция  $f(x) = |x|^m$  ( $m > 2$ ). Функции такого рода возникают при отыскании чебышевских аппроксимаций и рассматриваются в § 2.4.

Нетрудно убедиться в том, что никакой квадратичной параболой

$$\psi(x) = A + (x - C)^2 B/2 \tag{2.14}$$

мажорировать функцию  $|x|^m$  в диапазоне  $[-\infty, \infty]$  не удастся, так как при достаточно больших  $|x|$  функция  $f(x)$  обязательно превысит  $\psi(x)$ . Однако легко убедиться и в том, что в любом заранее заданном диапазоне  $[x^k, x^k + \gamma]$  ( $\gamma \geq 0$ ) функция  $f(x)$  мажорируется параболой

(2.14), если коэффициенты  $A, B, C$  выбрать из следующих соотношений:

$$\begin{aligned}\psi(x^k) &= f(x^k), \quad \psi'(x^k) = f'(x^k), \\ B^k &= \max \begin{cases} f''(x^k), \\ f''(x^k + \gamma), \end{cases}\end{aligned}\quad (2.15)$$

откуда, как обычно, получим, что

$$C = x^k - f'(x^k)/B, \quad A = f(x^k) - [f'(x^k)]^2/2B.$$

В самом деле, в точке  $x^k$  имеем  $f = \psi$  и  $f' = \psi'$ , а на отрезке  $[x^k, x^k + \gamma]$ , как видно из рис. 2.15,  $\psi'' \geq f''$ . Дважды интегрируя  $\psi''$  и  $f''$  с начальными условиями (2.15) в пределах от  $x^k$  до  $x^k + \gamma$ , мы убеждаемся в этом утверждении.

**Пример 2.11.** Задана функция  $f(y) = \alpha \exp(\beta y)$ . Функции такого рода являются слагаемыми целевой функции в основной задаче геометрического программирования [25]. Элементарной заменой переменных они приводятся к виду  $f(x) = \exp x$ .

Функции вида  $f(x) = \exp x$ , так же как и функции  $|x|^m$ , не мажорируются квадратичной параболой ни при каком  $x^0$  в силу того, что с ростом  $x$  экспонента растет быстрее любого, в том числе и квадратичного, полинома. Однако, так же как и для функции  $|x|^m$ , для нее можно указать параболу, мажорирующую  $f(x)$  на любом заранее заданном отрезке. Воспользовавшись формулой (1.13), можно показать, что на отрезке  $[x^0, x^n]$  функция  $\exp x$  мажорируется параболой

$$\psi(x) = A + (x - C)^2 B/2,$$

где

$$B = \begin{cases} \exp x^0, & \text{если } x^n \leq x^0 \\ \frac{\exp x^n - \exp x^0}{x^n - x^0}, & \text{если } x^n \geq x^0 \end{cases} = \max \begin{cases} \exp x^0, \\ \frac{\exp x^n - \exp x^0}{x^n - x^0}, \end{cases}$$

и, как обычно,

$$C = x^0 - \exp x^0/B \quad \text{и} \quad A = \exp x^0 - \frac{B}{2} (x^0 - C)^2.$$

В приведенных обозначениях  $x^n$  верхний индекс «н» означает, что в качестве одной из граничных точек взята точка минимума ньютоновской параболы, построенной в точке  $x^0$ , хотя ради справедливости следует заметить, что точку  $x^n$  можно было бы выбрать и по жребию.

**Пример 2.12.** Задана функция  $f(x_1, x_2) = r^m$  ( $m > 2$ ), где  $r = \|x - a\| = \sqrt{(x_1 - a_1)^2 + (x_2 - a_2)^2}$ . Функции такого рода возникают при отыскании точки, находящейся на минимаксном расстоянии от заданной совокупности объектов.

Пусть дана точка  $x^k$  и круг радиуса  $r^n$  с центром в точке  $a$ . Рассмотрение примера 2.10 показывает, что в вертикальной плоскости  $(f, s)$ ,

проходящей через точки  $a$  и  $x^k$ , функция,  $f = |s|^m$  на отрезке  $[-r^n, r^n]$  мажорируется квадратичной параболой

$$\psi(s) = A + (s - C)^2 B/2,$$

проходящей через точку  $(r^k, f(r^k))$ , если только

$$B = \max \begin{cases} f''_{ss}(r^k) \\ f''_{ss}(r^n) \end{cases} = \{m(m-1)\} \max \begin{cases} (r^k)^{m-2}; \\ (r^n)^{m-2}, \end{cases}$$

где  $r^k = \|x^k - a\|$ ,  $r^n = \|x^n - a\|$ .

При этом точка  $C$  располагается внутри отрезка  $[0, r^k]$  на абсолютном расстоянии  $f'(r^k)/B$  от точки  $r^k$  и, следовательно, на относительном расстоянии

$$\alpha = \frac{1}{|r^k|} \frac{f'(r^k)}{B} = \min \begin{cases} (m-1)^{-1}, \\ (r^k/r^n)^{m-2}/(m-1) \end{cases}$$

от нее. В плоскости  $(x_1, x_2)$  координатами точки минимума параболы  $\psi(s)$  являются

$$d_1 = \alpha a_1 + (1-\alpha) x_1^k \text{ и } d_2 = \alpha a_2 + (1-\alpha) x_2^k.$$

Параболоид  $\psi(x_1, x_2) = A + [(x_1 - d_1)^2 + (x_2 - d_2)^2]B/2$  мажорирует функцию  $f(x_1, x_2)$  в круге  $|r| \leq r^n$ . Чтобы убедиться в этом, достаточно рассмотреть одну из концентрических окружностей  $|r| = k \leq r^n$ . На этой окружности  $f(x_1, x_2)$  остается неизменной, а  $\psi(x_1, x_2)$  достигает минимума в точке  $z$  пересечения этой окружности с отрезком, соединяющим точки  $a$  и  $x^k$ . В результате имеем  $f(x) = f(z) \leq \psi(z) \leq \psi(x)$ , что и требовалось доказать.

**Пример 2.13.** Задана функция  $f(x_1, x_2) = \exp r$ , где  $r = \sqrt{(x_1 - a_1)^2 + (x_2 - a_2)^2}$ . Функции такого рода так же, как и  $f = |r|^m$  ( $m > 2$ ), играют важную роль при решении минимаксных задач размещения. Дословно повторяя рассуждения предыдущего примера, убедимся в том, что параболоид

$$\psi(x_1, x_2) = A + [(x_1 - d_1)^2 + (x_2 - d_2)^2] B/2,$$

где

$$B = \max \begin{cases} \exp r^k, \\ \frac{\exp r^n - \exp r^k}{r^n - r^k}, \end{cases}$$

$$d_1 = \alpha a_1 + (1-\alpha) x_1^k, \quad x_2 = \alpha a_2 + (1-\alpha) x_2^k,$$

$$\alpha = \frac{f'_r(r^k)}{r^k B} = \frac{\exp r^k}{r^k B}, \quad r^k = \|x^k - a\|,$$

мажорирует функцию  $f(x_1, x_2) = \exp r$  в круге  $\|x - a\| \leq r^n$ .

Пусть дана избыточная и противоречивая система алгебраических уравнений

$$\sum_j a_{ij} x_j + b_i = 0, i = 1, 2, \dots, N.$$

Решением этой системы в смысле Чебышева является такой вектор  $\mathbf{X} = (x_1, x_2, \dots, x_n)$ , который обращает в минимум максимальную остаточную невязку

$$\Phi(\mathbf{x}) = \max \left( \left| \sum_j a_{1j} x_j + b_1 \right|, \left| \sum_j a_{2j} x_j + b_2 \right|, \dots, \left| \sum_j a_{Nj} x_j + b_N \right| \right).$$

Со ссылками на работы Пойя Е. Я. Ремез [26] указывает, что решение задачи сводится к многократной минимизации выражения

$$\Phi_m(\mathbf{x}) = \sum_{i=1}^N \left| \sum_{j=1}^n a_{ij} x_j + b_i \right|^m \quad (2.16)$$

при возрастающих  $m$  (не обязательно целочисленных) с последующим нахождением предельной точки  $\mathbf{x} = \lim_{m \rightarrow \infty} \hat{\mathbf{x}}^m$ , где  $\hat{\mathbf{x}}^m = \arg \min \Phi_m(\mathbf{x})$ .

Для минимизации выражения (2.16) Е. Я. Ремез доказал пригодность так называемого  $\alpha$ -алгоритма, сводящегося к итерационной процедуре, на каждом шаге которой:

а) решается задача на нахождение минимума суммы квадратов линейных функций

$$\psi_x^k = \sum_i P_i^k \left( \sum_j a_{ij} x_j + b_i \right)^2, \quad (2.17)$$

где

$$P_i^k = \left| \sum_j a_{ij} x_j^k + b_i \right|^{m-2},$$

а  $\mathbf{x}^k = (x_1^k, x_2^k, \dots, x_n^k)$  — некоторое «текущее» приближение;

б) отыскивается новое приближение по формуле

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k (\mathbf{x}_{\min}^k - \mathbf{x}^k),$$

где под  $\mathbf{x}_{\min}^k$  понимается минимум функции (2.17), а  $\alpha^k$  назначается по результатам одномерного поиска по  $\alpha$  выражения (2.16), в которое вместо  $\mathbf{x}$  подставлено выражение

$$\mathbf{x} = \mathbf{x}^k + \alpha (\mathbf{x}_{\min}^k - \mathbf{x}^k).$$

Основным результатом Е. Я. Ремеза является доказательство того, что  $\alpha^k$  конечно ( $\alpha \geq \delta^0 > 0$ ) и приближенно равно  $1/(m-1)$ . Величина  $(\mathbf{x}_{\min}^k - \mathbf{x}^k)/(m-1)$  соответствует значению ньютоновской поправки, вычисленной вблизи точки минимума функции (2.16).

Попытаемся с помощью метода ВВКМ обеспечить отыскание минимума сумм вида  $\sum_i \varphi_i(y_i)$ , где  $y_i = \sum_j a_{ij}x_j + b_i$ , все слагаемые  $\varphi_i(y_i)$  выпуклы вниз и имеют вид  $\varphi_i(y_i) = |y_i|^{m_i}$ , причем в отдельных случаях может быть  $m_i > 2$ .

В соответствии с общей идеологией метода ВВКМ мажорируем каждое слагаемое  $\varphi_i(y_i)$  параболой

$$\psi_i^k(y_i) = A_i^k + (y_i - C_i^k)^2 B_i^k / 2, \quad (2.18)$$

где  $A_i^k$ ,  $B_i^k$  и  $C_i^k$  зависят от текущего приближения  $x^k$ . К сожалению, как уже говорилось, слагаемые вида  $|y_i|^{m_i}$  ( $m_i > 2$ ) такому мажорированию во всем диапазоне  $-\infty < y_i < \infty$  не поддаются. Несмотря на это проведем параболу (2.18), выполняя условия

$$\begin{aligned} \varphi_i(y_i) &= \psi_i(y_i) \big|_{y_i = y_i^k}, \\ \varphi_i'(y_i) &= \psi_i'(y_i) \big|_{y_i = y_i^k}, \\ \varphi_i''(y_i) &= \psi_i''(y_i) \big|_{y_i = y_i^k}. \end{aligned} \quad (2.19)$$

Отметим, что при любых  $B_i^k$  и  $C_i^k$  надлежащим выбором  $A_i^k$  в (2.18) можно добиться соблюдения первого из условий (2.19). Далее при любом  $B_i^k$  надлежащим подбором  $C_i^k$  можно добиться соблюдения и второго условия (2.19). Разлагая функции  $\psi_i^k$  и  $\varphi_i$  в ряд в некоторой окрестности точки  $x^k$ , легко установить, что знак разности  $\psi_i - \varphi_i$  в этой окрестности определяется знаком разности  $\psi_i'' - \varphi_i''$ . Поэтому, выбирая  $B_i^k$  из условия  $B_i^k > \varphi_i''$ , добиваемся мажорирования функции  $\varphi_i$  функцией  $\psi_i$  в некоторой окрестности точки  $x^k$ . Минимум суммы мажорант указывает направление уменьшения функции  $\varphi(x)$ . Если точка  $x^{k+1} = \arg \min_x \sum_i \psi_i$  такова, что для всех  $i$  справедливо  $\varphi_i(x^{k+1}) \leq \psi_i(x^{k+1})$ , то промежуточная цель итерационного процесса достигнута. Если это не так, то, значит, был сделан слишком большой шаг и мы вышли из области мажорирования хотя бы по одному из слагаемых. Если шаг несколько уменьшить, то мы введем точку  $\hat{x}^{k+1} = x^k + \alpha(x^{k+1} - x^k)$  в нужную область.

Помимо прямого уменьшения длины шага можно пойти и по другому, более рациональному пути, а именно, расширять область мажорирования только того слагаемого, у которого  $\varphi_i(x^{k+1}) \geq \psi_i(x^{k+1})$ . Расширение области мажорирования достигается увеличением  $B_i^k$ .

Пусть, например, надо минимизировать  $\Phi = \sum_i \varphi_i = \sum_i |y_i|^{m_i}$ , где все  $m_i$  одинаковы. Положив  $\varphi_i''(y_i^k) = B_i^n$  и построив параболу Ньютона  $\psi_i^{nk}(x)$  по формулам

$$C_i^n = y_i^k - \frac{\varphi_i'(y_i^k)}{B_i^n}, \quad A_i^n = \varphi_i(y_i^k) - \frac{[\varphi_i'(y_i^k)]^2}{2 B_i^n},$$

$$\psi_i^{nk} = A_i^n + \frac{B_i^n}{2} (y_i - C_i^n)^2,$$

$$\psi^{nk}(x) = \sum_i \psi_i^{nk}(y_i),$$

где

$$y_i = \sum_j a_{ij} x_j + b_i = a_i x + b_i, \quad y_i^k = a_i x^k + b_i,$$

$$a_i = \|a_{i1}, a_{i2}, \dots, a_{in}\|, \quad x^T = \|x_1, x_2, \dots, x_n\|,$$

мы в состоянии найти точку

$$z^k = \operatorname{argmin}_x \sum_i \psi_i^k(y_i)$$

и значение ньютоновской поправки  $z^k - x^k$ . Для этой цели по составляющим  $y_i - C_i^n = y_i - y_i^k + \varphi_i'/B_i^n$  построим вектор  $y - C^n = ax - ax^k + (B^n)^{-1} \nabla_y \varphi$ . В последнем выражении использованы следующие сокращения:

$$y = \left\| \begin{array}{c} y_1 \\ y_2 \\ \dots \\ y_N \end{array} \right\|, \quad C^n = \left\| \begin{array}{c} C_1^n \\ C_2^n \\ \dots \\ C_N^n \end{array} \right\|, \quad a = \left\| \begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & \dots & a_{Nn} \end{array} \right\| = \left\| \begin{array}{c} a_1 \\ a_2 \\ \dots \\ a_N \end{array} \right\|,$$

$$B^n = \left\| \begin{array}{cccc} B_1^n & 0 & \dots & 0 \\ 0 & B_2^n & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & B_N^n \end{array} \right\|, \quad \Delta_y \varphi = \left\| \begin{array}{c} \varphi'_1(y_1) \\ \varphi'_2(y_2) \\ \dots \\ \varphi'_N(y_N) \end{array} \right\|_{y=y^k}$$

Как следует из основных формул метода наименьших квадратов [20], точкой минимума суммы парабол  $\sum \psi_i^n$  является

$$z^k = (a^T B^n a)^{-1} a^T B^n (a x^k - (B^n)^{-1} \nabla_y \varphi) = \\ = x^k - (a^T B^n a)^{-1} a^T \nabla_y \varphi|_{y=y^k}.$$

Ньютоновская поправка

$$\Delta x^{nk} = z^k - x^k = - (a^T B^n a)^{-1} a^T \nabla_y \varphi|_{y=y^k}.$$

Всякая замена любого из элементов диагональной матрицы  $B^n$  большим числом уменьшает абсолютное значение составляющих вектора  $z^k = x^k$ , и, следовательно, каждая из величин  $y_i^{k+1} = a^T x^{k+1} + b_i$ , где

$$x^{k+1} = \underset{x}{\operatorname{argmin}} \sum \psi_i(y_i),$$

$$\psi_i = A_i + (y_i - C_i) B_i / 2, \quad B_i \geq B_i^n,$$

оказывается внутри соответствующего отрезка мажорирования

$$[y_i^k, \tilde{y}_i^k], \quad \text{где } \tilde{y}_i^k = a_i z^k + b_i.$$

Рассмотрение примера 2.10 показывает, что отрезки  $[y_i^k, \tilde{y}_i^k]$  являются отрезками мажорирования функций  $\varphi_i(y)$  функциями  $\psi_i(y)$ , если  $B_i$ , в частности, назначить так:

$$B_i = \max \begin{cases} \varphi_i''(y^k) \\ \varphi_i''(\tilde{y}^k) \end{cases} = \max \begin{cases} B_i^n, \\ \varphi_i''(\tilde{y}^k). \end{cases}$$

Отсюда вытекает следующий алгоритм минимизации сумм вида  $\varphi(x) = \sum |y_i|^{m_i}$ , где  $y_i = \sum_j a_{ij} x_j + b_i$  ( $m_i > 2$ ).

1. Задаемся произвольной точкой  $x^k$  и подсчитываем значения  $y_i^k$  по формуле

$$y_i^k = \sum_j a_{ij} x_j^k + b_i.$$

2. Подсчитываем значения первых и вторых производных функций  $\varphi_i(y_i)$  в точке  $y_i^k$  и величину ньютоновской поправки:

$$\varphi_i'(y_i^k) = m_i |y_i^k|^{m_i-1} \operatorname{sign} y_i^k,$$

$$\varphi_i''(y_i^k) = m_i(m_i-1) |y_i^k|^{m_i-2},$$

$$\Delta x^{nk} = -(a^T B^n a)^{-1} a^T \nabla_y \varphi,$$

где  $a = \|a_{ij}\|$ ,  $B_i^n = \varphi_i''(y_i^k)$  — элементы диагональной матрицы  $B^n$ .

3. Находим значения  $\tilde{y}_i^k = \sum_j a_{ij} (x_j^k + \Delta x_j^{nk}) + b_i$ .

4. Корректируем значения  $B_i$ :

$$B_i = \max \begin{cases} B_i^n, \\ \varphi_i''(y_i^k). \end{cases}$$

5. Находим окончательное значение поправки  $\Delta x^k$  и следующее приближение к точке минимума  $f$ :

$$\Delta x^k = -(a^T B a)^{-1} a^T \nabla_y \varphi,$$

где  $\mathbf{B}$  — диагональная матрица с элементами  $B_i$ ,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta \mathbf{x}^k.$$

6. Возвращаемся к п.1.

Аналогично путем корректировки некоторых  $B_i$  строится алгоритм минимизации сумм вида  $\varphi = \sum_i e^{u_i}$ . Проиллюстрируем сказанное числовыми примерами.

**Пример 2.14.** Задана функция

$$f(x_1, x_2) = |x_1 + x_2 - 3|^7 + |2x_1 + x_2 - 6|^9 + |x_1 + 4x_2 - 20|^3.$$

Ход итерационного процесса поиска минимума из начальной точки  $\mathbf{x}^0$  ( $x_1^0, x_2^0$ ) = (1,1) приведен в табл. 2.1.

Т а б л и ц а 2.1

| Номер шага<br>$k$ | $x_1^k$ | $x_2^k$ | $f(x_1^k, x_2^k)$ | Номер шага<br>$k$ | $x_1^k$ | $x_2^k$ | $f(x_1^k, x_2^k)$ |
|-------------------|---------|---------|-------------------|-------------------|---------|---------|-------------------|
| 0                 | 1,000   | 1,000   | 23059,000         | 5                 | 0,156   | 4,408   | 42,932            |
| 1                 | -0,043  | 3,590   | 3947,051          | 6                 | 0,193   | 4,385   | 42,412            |
| 2                 | -0,299  | 4,520   | 734,983           | 7                 | 0,194   | 4,384   | 42,411            |
| 3                 | -0,175  | 4,619   | 157,947           | 8                 | 0,194   | 4,384   | 42,411            |
| 4                 | 0,026   | 4,490   | 56,276            |                   |         |         |                   |

Еще раз напомним, что в данном случае решена несколько более общая задача чем та, для решения которой разработан алгоритм Ремеза, так как отдельные слагаемые функции имеют различающиеся между собой показатели степени.

**Пример 2.15.** Задана функция

$$f(x_1, x_2, x_3) = e^{2x_1+1} + e^{-3x_1+2x_2+2} + e^{-x_2+3x_3+3} + e^{-x_3+4}.$$

Процесс поиска минимума заданной функции отражен в табл. 2.2.

Т а б л и ц а 2.2

| Номер шага | $x_1^k$ | $x_2^k$ | $x_3^k$ | $f(x_1^k, x_2^k, x_3^k)$ |
|------------|---------|---------|---------|--------------------------|
| 0          | 5,212   | 6,016   | 3,813   | 96112,938                |
| 1          | 4,528   | 5,332   | 3,130   | 24477,574                |
| 2          | 3,844   | 4,648   | 2,446   | 6238,816                 |
| 3          | 3,161   | 3,962   | 1,767   | 1600,143                 |
| 4          | 2,479   | 3,265   | 1,116   | 429,528                  |
| 5          | 1,809   | 2,507   | 0,623   | 146,146                  |
| 6          | 1,196   | 1,699   | 0,400   | 84,647                   |
| 7          | 0,784   | 1,149   | 0,263   | 76,027                   |
| 8          | 0,692   | 1,026   | 0,232   | 75,795                   |
| 9          | 0,691   | 1,025   | 0,232   | 75,795                   |

## 2.4. ЛИНЕЙНАЯ СХОДИМОСТЬ МЕТОДА ВВКМ

Пусть  $f(x^k)$  — значение минимизируемой функции в точке  $x^k$ , полученной на  $k$ -м шаге итерационного процесса,  $f(x^*)$  — ее значение в стационарной точке  $x^*$ . Напомним, что сходимость называется линейной, если

$$f(x^{k+1}) - f(x^*) \leq \alpha [f(x^k) - f(x^*)], \quad (2.20)$$

где  $\alpha < 1$ . Из (2.20) следует, что

$$f(x^k) - f(x^*) \leq \alpha^k [f(x^0) - f(x^*)],$$

где  $x^0$  — начальное приближение, т. е. линейная сходимость эквивалентна сходимости со скоростью геометрической прогрессии.

**Теорема [40].** Пусть  $f(x)$  выпукла во всей области ее существования и строго выпукла в некоторой области  $U$ , окружающей точку  $x^*$ , а минимальное собственное значение матрицы ее вторых производных (гессиана)

$$H(x) = \nabla_x^T \nabla_x f(x) = \left\| \frac{\partial^2}{\partial x_i \partial x_j} f(x) \right\|$$

больше некоторого положительного  $\delta$ . Пусть далее  $\psi(x, x^k)$  — параболоид, мажорирующий функцию  $f(x)$ , такой, что

$$\psi(x, x^k)|_{x=x^k} = f(x)|_{x=x^k},$$

$$\nabla_x \psi(x, x^k)|_{x=x^k} = \nabla_x f(x)|_{x=x^k},$$

$$(x - x^k)^T C^k (x - x^k) \geq (x - x^k)^T [\nabla_x^T \nabla_x f(x^k)] (x - x^k),$$

где

$$C^k = \nabla_x^T \nabla_x \psi(x, x^k)|_{x=x^k} < C^0$$

— некоторая положительно-определенная матрица.

В этих условиях имеет место линейная сходимость  $f(x^k)$  к  $f(x^*)$ , где

$$x^k = \underset{x}{\operatorname{argmin}} \psi(x, x^{k-1}), \quad k = 1, 2, \dots,$$

а  $x^0 \in U$ .

## ЗАКЛЮЧЕНИЕ

Известные (и классические, и разработанные в последнее время) методы отыскания безусловного экстремума оказываются плохо пригодными для минимизации слабо выпуклых функций с плохими аналитическими свойствами. В то же время существует класс очень важных с практической точки зрения функций  $f(x)$  с именно такими свойствами, у которых сравнительно просто отыскивается абсолютный минимум при помощи метода ВВКМ. В наиболее распространенном случае этот

метод предполагает, что функция  $f(x)$  представима в виде суммы  $f(x) = \sum_i f_i(y_i)$ , где  $y_i = \sum_j a_{ij}x_j + b_i$ , и основан на мажорировании каждого слагаемого этой функции квадратичной параболой, коэффициенты которой зависят от произвольным образом выбранного начального приближения. Минимум суммы таких парабол находится без труда при помощи стандартной процедуры наименьших квадратов. Этот минимум принимается за начальное приближение на следующем шаге, устроенном аналогично. Доказано, что этот итерационный процесс приводит к отысканию искомого экстремума функции  $f(x)$ .

### Глава 3.

#### РЕШЕНИЕ НЕКОТОРЫХ ЗАДАЧ РАЗМЕЩЕНИЯ МЕТОДОМ ВАРИАЦИОННО-ВЗВЕШЕННЫХ КВАДРАТИЧЕСКИХ МАЖОРАНТ

С необходимостью наилучшего размещения объектов приходится сталкиваться при решении самых разнообразных инженерных задач. Чаще всего это задачи оптимального соединения объектов транспортной или энергетической сетью и каналами связи. Достаточно полный перечень и классификацию задач размещения можно найти в [36]. В общей постановке эти задачи довольно сложны и служат объектом изучения специалистов по системному анализу и исследованию операций. Однако многие задачи размещения в своей основе имеют несложные модели, допускающие ясную геометрическую интерпретацию.

Общим для этих моделей является понятие расстояния. Под расстоянием между двумя точками  $X^{(1)}(x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)})$  и  $X^{(2)}(x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)})$  в  $n$ -мерном пространстве понимается величина

$$l_p(X^{(1)}, X^{(2)}) = (|x_1^{(1)} - x_1^{(2)}|^p + |x_2^{(1)} - x_2^{(2)}|^p + \dots + |x_n^{(1)} - x_n^{(2)}|^p)^{1/p} = \left( \sum_{i=1}^n |x_i^{(1)} - x_i^{(2)}|^p \right)^{1/p}, \quad (3.1)$$

где  $p \geq 1$ .

Введенная функция расстояния  $l_p$  обладает следующими свойствами [3].

1. Расстояние неотрицательно:  $l_p(X^{(1)}, X^{(2)}) \geq 0$ , причем

$$l_p(X^{(1)}, X^{(1)}) = 0.$$

2. Расстояние симметрично:  $l_p(X^{(1)}, X^{(2)}) = l_p(X^{(2)}, X^{(1)})$ .

3. Расстояние между двумя точками зависит только от их взаимного расположения и инвариантно по отношению к параллельным переносам.

4. Функция расстояния удовлетворяет неравенству треугольника  $l_p(X^{(1)}, X^{(3)}) \leq l_p(X^{(1)}, X^{(2)}) + l_p(X^{(2)}, X^{(3)})$ .

5. Функция расстояния однородна, т. е. если точка  $X^{(1)}$  имеет координаты  $(x_i^{(1)} | i = 1, 2, \dots, n)$ , а точка  $X^{(2)}$  — координаты  $(a x_i^{(1)} | i = 1, 2, \dots, n)$ , где  $a$  — некоторая неотрицательная постоянная, то  $l_p(0, X^{(2)}) = a l_p(0, X^{(1)})$ , здесь точка 0 обозначает начало координат  $(0, 0, \dots, 0)$ .

В практических задачах чаще всего используются евклидово ( $p = 2$ ) и прямоугольное ( $p = 1$ ) расстояния. Для  $1 < p < 2$  расстояние  $l_p$  лежит между ними. Для  $p > 2$  расстояние  $l_p$  меньше евклидова расстояния и продолжает уменьшаться по мере увеличения  $p$ . В пределе при  $p = \infty$  получаем чебышевское расстояние

$$l_{\infty}(X^{(1)}, X^{(2)}) = \max \{ |x_1^{(1)} - x_1^{(2)}|, |x_2^{(1)} - x_2^{(2)}|, \dots, |x_n^{(1)} - x_n^{(2)}| \}.$$

В настоящей главе рассматриваются некоторые геометрические задачи, связанные с минимизацией сумм расстояний.

### 3.1. РАЗМЕЩЕНИЕ ТОЧЕЧНЫХ ОБЪЕКТОВ. МИНИМИЗАЦИЯ СУММАРНЫХ РАССТОЯНИЙ

Пусть дана плоскость  $A_1 x_1 + A_2 x_2 + A_3 x_3 + A_4 = 0$  и точка  $X$  с координатами  $x_1, x_2, x_3$  в трехмерном пространстве. Из аналитической геометрии известно, что расстояние от этой точки до заданной плоскости определяется выражением

$$l = |A_1 x_1 + A_2 x_2 + A_3 x_3 + A_4| (V A_1^2 + A_2^2 + A_3^2)^{-1}. \quad (3.2)$$

Если задано  $n$  плоскостей, то точкой, находящейся на минимальном суммарном расстоянии от их совокупности, будем считать точку  $X(x_1, x_2, x_3)$  такую, что

$$X = \arg \min_{x_1, x_2, x_3} \sum_{i=1}^n l_i(x_1, x_2, x_3),$$

где

$$l_i(x_1, x_2, x_3) = |A_{1i} x_1 + A_{2i} x_2 + A_{3i} x_3 + A_{4i}| (V A_{1i}^2 + A_{2i}^2 + A_{3i}^2)^{-1}.$$

В соответствии с примерами 2.3 и 2.4 можно утверждать, что каждое слагаемое  $l_i(x_1, x_2, x_3)$  при произвольных начальных условиях  $x_1^0, x_2^0$  и  $x_3^0$  мажорируется параболоидом

$$\psi_i(x_1, x_2, x_3) = \frac{(A_{1i} x_1 + A_{2i} x_2 + A_{3i} x_3 + A_{4i})^2}{|A_{1i} x_1^0 + A_{2i} x_2^0 + A_{3i} x_3^0 + A_{4i}| V A_{1i}^2 + A_{2i}^2 + A_{3i}^2},$$

а сумма  $L = \sum_{i=1}^n l_i$  — соответствующей суммой параболоидов

$$\psi = \sum_i \psi_i(x_1, x_2, x_3).$$

В силу этого, начав с произвольной точки  $x_1^0, x_2^0$  и  $x_3^0$ , можно организовать процесс последовательных приближений, осуществляемых в соответствии с формулой

$$X^{k+1} = \arg \min_X \sum_{i=1}^n P_i^k (A_{1i} x_1 + A_{2i} x_2 + A_{3i} x_3 + A_{4i})^2, \quad (3.3)$$

где

$$P_i^k = (\sqrt{A_{1i}^2 + A_{2i}^2 + A_{3i}^2} |A_{1i} x_1^k + A_{2i} x_2^k + A_{3i} x_3^k + A_{4i}|)^{-1}.$$

В результате этих последовательных приближений будет получена точка, являющаяся точным решением рассматриваемой задачи.

Следующей разновидностью задачи размещения является отыскание точки на плоскости, наименее удаленной от заданной совокупности точек (задача Штейнера, она же задача Вебера). С решения именно этой задачи Е. Вейсфельдом началось практическое применение ВВКМ [41].

Пусть на плоскости задана совокупность точек  $A_1, A_2, \dots, A_n$  и требуется найти точку  $X(x_1, x_2)$ , наименее удаленную от этой совокупности. Эта задача является одной из центральных в теории размещения [36], имеет многочисленные практические приложения и допускает целый ряд обобщений. Некоторые из этих обобщений будут рассмотрены ниже.

Расстояние между точками  $X$  и  $A_i$  определяется выражением

$$l_i = |\sqrt{(x_1 - a_{1i})^2 + (x_2 - a_{2i})^2}|.$$

Если задано  $n$  точек, то наименее удаленной от заданной совокупности точек  $(a_{1i}, a_{2i})$  ( $i = 1, 2, \dots, n$ ) будем считать точку  $X(x_1, x_2)$  такую, что

$$X = \arg \min_{x_1, x_2} \sum_{i=1}^n l_i(x_1, x_2). \quad (3.4)$$

Как показано в гл. 2, при произвольных  $x_1^0$  и  $x_2^0$  каждое слагаемое  $l_i(x_1, x_2)$  в (3.4) мажорируется параболоидом

$$\psi_i(x_1, x_2) = \frac{l_i^2(x_1, x_2)}{2 |l_i(x_1^0, x_2^0)|} = \frac{(x_1 - a_{1i})^2 + (x_2 - a_{2i})^2}{2 |\sqrt{(x_1^0 - a_{1i})^2 + (x_2^0 - a_{2i})^2}|}.$$

С учетом этого для поиска координат оптимальной точки можно записать следующую рекуррентную формулу:

$$X^{k+1} = \arg \min_{x_1, x_2} \left( \sum_{i=1}^n P_i^k (x_1 - a_{1i})^2 + \sum_{i=1}^n P_i^k (x_2 - a_{2i})^2 \right), \quad (3.5)$$

где

$$P_i^k = (2 |\sqrt{(x_1^k - a_{1i})^2 + (x_2^k - a_{2i})^2}|)^{-1}.$$

При этом отдельным точкам могут быть приписаны весовые коэффициенты. Введение этих коэффициентов позволяет учесть различную значимость объектов.

В качестве непосредственного обобщения задачи Штейнера можно рассмотреть ее многомерный вариант, т. е. отыскание точки на минимальном расстоянии от заданной совокупности точек в многомерном пространстве. Учет многомерности пространства незначительно усложняет итерационную процедуру (3.5), которая для евклидовой метрики может быть переписана следующим образом:

$$X^{k+1} = \arg \min_{x_1, x_2, \dots, x_n} \left\{ \sum_{j=1}^m \sum_{i=1}^n P_i^k (x_j - a_{ji})^2 \right\}, \quad (3.6)$$

где

$$P_i^k = w_i \left( 2 \left| \sqrt{\sum_{j=1}^m (x_j^k - a_{ji})^2} \right| \right)^{-1};$$

$m$  — размерность пространства;  $n$  — число заданных точек;  $w_i$  — коэффициенты важности (вес) отдельных точек.

Распространенным вариантом задачи Штейнера является размещение не одного, а нескольких точечных объектов. Этой задаче посвящена обширная литература. Задача формулируется следующим образом: найти

$$X^* = (X_1^*, X_2^*, \dots, X_m^*) = \arg \min_X f(X), \quad (3.7)$$

где

$$f(X) = \sum_{1 \leq j \leq k \leq m} v_{jk} l(X_j, X_k) + \sum_{j=1}^m \sum_{i=1}^n w_{ji} l(X_j, A_i);$$

$X_j$  — вектор координат вновь размещаемого  $j$ -го объекта,  $1 \leq j \leq m$ ;  $A_i$  — вектор координат заданного  $i$ -го объекта,  $1 \leq i \leq n$ ;  $l(X_j, X_k)$  — расстояние (евклидово, прямоугольное или др.) между вновь размещаемыми  $j$ -м и  $k$ -м объектами,  $1 \leq j \leq k \leq m$ ;  $l(X_j, A_i)$  — расстояние между вновь размещаемым объектом  $j$  и заданным  $i$ -м объектом;  $v_{jk}$  — неотрицательный коэффициент важности связи между вновь размещаемыми  $j$ -м и  $k$ -м объектами;  $w_{ji}$  — неотрицательный коэффициент важности связи вновь размещаемого объекта  $j$  и заданного  $i$ -го объекта.

Выпуклость функционала  $f(X)$  обеспечивается выпуклостью функций расстояния  $l$  и неотрицательностью весовых коэффициентов  $v_{jk}$  и  $w_{ji}$ . Из выпуклости функционала  $f(X)$  следует отсутствие у него локальных экстремумов. Это является решающим обстоятельством для задачи с заданной структурой связей. Структура связей задается в неявной форме через коэффициенты важности  $v_{jk}$  и  $w_{ji}$ .

Еще одним направлением обобщения задачи Штейнера, которого мы коснемся, является отыскание точки, наименее удаленной от сово-

купности объектов конечных размеров (отрезков, выпуклых фигур и т. п.). Пусть, например, дано  $n$  отрезков  $A_i B_i$  на плоскости, заданных своими концами  $A_i (a_{1i}, a_{2i})$  и  $B_i (b_{1i}, b_{2i})$ , где  $i = 1, 2, \dots, n$ .

Под расстоянием от отрезка  $AB$  до точки  $X (x_1, x_2)$  будем понимать (рис. 3.1)

$$l(X) = \begin{cases} \delta, & \text{если } |r_A^2 - r_B^2| \leq a^2, \\ r_A, & \text{если } |r_A^2 - r_B^2| > a^2, r_A^2 < r_B^2, \\ r_B, & \text{если } |r_A^2 - r_B^2| > a^2, r_A^2 > r_B^2, \end{cases} \quad (3.8)$$

где

$$r_A = \sqrt{(x_1 - a_1)^2 + (x_2 - a_2)^2}, \quad r_B = \sqrt{(x_1 - b_1)^2 + (x_2 - b_2)^2}, \\ a = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}, \quad \delta = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2}.$$

В свою очередь координаты точки  $C$  определяются из соотношений

$$c_1 = \alpha a_1 + (1 - \alpha) b_1, \quad c_2 = \alpha a_2 + (1 - \alpha) b_2,$$

где

$$\alpha = \frac{(a_1 - b_1)(a_1 - x_1) + (a_2 - b_2)(a_2 - x_2)}{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

— отношение, в котором перпендикуляр, опущенный из искомой точки, делит заданный отрезок.

Так как мы имеем дело с  $n$  отрезками, то необходимо минимизировать суммарное расстояние и отыскать точку  $X (x_1, x_2)$  такую, что

$$X = \arg \min_{x_1, x_2} \sum_{i=1}^n l_i(x_1, x_2). \quad (3.9)$$

Функция расстояния для каждого из заданных отрезков образует симметричную поверхность, похожую на чашу современного стадиона (рис. 3.2). В основании этой поверхности лежит заданный отрезок.

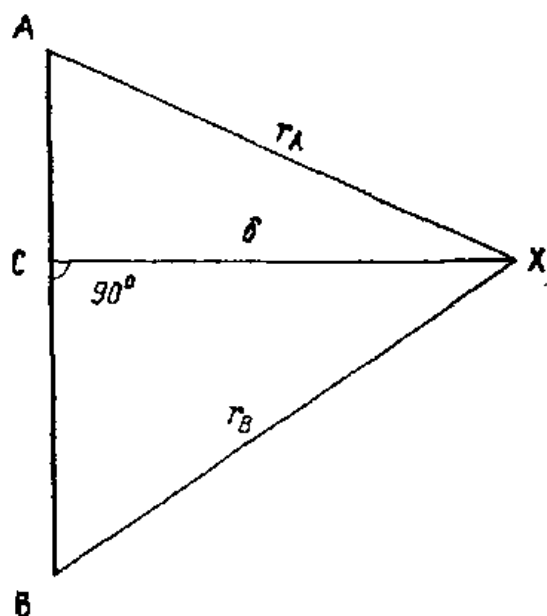


Рис. 3.1

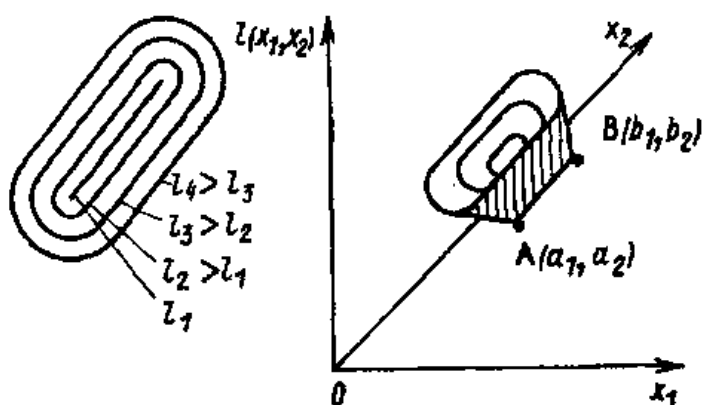


Рис. 3.2

В соответствии с общей идеологией метода ВВКМ для каждой из таких фигур, число которых равно числу заданных отрезков, строим мажорирующие поверхности. Как показано в гл. 2 (пример 2.8), в качестве такой мажоранты можно выбрать фигуру, описываемую уравнением

$$\psi(x_1, x_2) = \text{const} + [(x_1 - u_1^0)^2 + (x_2 - u_2^0)^2] / l(x_1^0, x_2^0),$$

где  $x_1^0, x_2^0$  — координаты исходной точки, а

$$(u_1^0, u_2^0) = \begin{cases} (c_1^0, c_2^0), & \text{если } |(r_A^0)^2 - (r_B^0)^2| \leq a^2, \\ (a_1^0, a_2^0), & \text{если } |(r_A^0)^2 - (r_B^0)^2| > a^2 \text{ и } (r_A^0)^2 < (r_B^0)^2, \\ (b_1^0, b_2^0), & \text{если } |(r_A^0)^2 - (r_B^0)^2| > a^2 \text{ и } (r_B^0)^2 < (r_A^0)^2. \end{cases} \quad (3.10)$$

Верхний индекс у переменных  $u^0, a^0, b^0$  и  $c^0$  в приведенной выше формуле указывает, что эти параметры определяются для точки начального приближения  $(x_1^0, x_2^0)$ . Построив мажорирующие параболоиды, мы без труда организуем итерационный процесс поиска минимума суммы исходных функций расстояния

$$X^{k+1} = \arg \min_X \left\{ \sum_{i=1}^n P_i^k [(x_1 - u_{1i}^k)^2 + (x_2 - u_{2i}^k)^2] \right\}, \quad (3.11)$$

где

$$P_i^k = [l_i(x_1^k, x_2^k) + \varepsilon]^{-1},$$

причем  $(u_{1i}^k, u_{2i}^k)$  и  $l_i(x_1^k, x_2^k)$  определяются по (3.8) и (3.10), исходя из расположения точки текущего приближения по отношению к  $i$ -му отрезку;  $\varepsilon$  — малое наперед заданное положительное число.

В задаче Штейнера на плоскости каждую точку  $A_i(a_{1i}, a_{2i})$  можно считать результатом пересечения плоскости  $(x_1, x_2)$  прямой, параллельной оси  $x_3$ . Тогда оптимальная точка  $X(x_1, x_2)$  минимизирует сумму расстояний до  $n$  таких параллельных между собой прямых. В качестве еще одного из обобщений задачи Штейнера рассмотрим случай, когда прямые полагаются непараллельными между собой.

Каждую точку пространства будем изображать с помощью вектора

$$X = x_1 e_1 + x_2 e_2 + x_3 e_3, \quad (3.12)$$

где  $e_1, e_2$  и  $e_3$  — единичные орты соответствующих осей.

Любую прямую, проходящую через точку  $A_i(a_{1i}, a_{2i}, a_{3i})$  в направлении  $\alpha_{1i} e_1 + \alpha_{2i} e_2 + \alpha_{3i} e_3$ , можно представить в виде

$$r_i(t) = (a_{1i} + \alpha_{1i} t) e_1 + (a_{2i} + \alpha_{2i} t) e_2 + (a_{3i} + \alpha_{3i} t) e_3, \quad (3.13)$$

где

$$\alpha_{1i}^2 + \alpha_{2i}^2 + \alpha_{3i}^2 = 1.$$

Перенесем начало координат в точку  $A_i = a_1 e_1 + a_2 e_2 + a_3 e_3$ . Тогда уравнением точки (3.12) будет

$$X - A_i = (x_1 - a_{1i}) e_1 + (x_2 - a_{2i}) e_2 + (x_3 - a_{3i}) e_3,$$

а уравнением прямой (3.13) —

$$r_i(t) - A_i = (\alpha_{1i} e_1 + \alpha_{2i} e_2 + \alpha_{3i} e_3) t = \alpha_i t.$$

Модуль расстояния между точкой  $X$  и прямой (3.13) определяется формулой

$$|l_i| = \text{mod} [(X - A_i) \times \alpha_i] = \text{mod} \begin{vmatrix} e_1 & e_2 & e_3 \\ \alpha_{1i} & \alpha_{2i} & \alpha_{3i} \\ x_1 - a_{1i} & x_2 - a_{2i} & x_3 - a_{3i} \end{vmatrix} = \\ = \sqrt{[\alpha_{2i}(x_3 - a_{3i}) - \alpha_{3i}(x_2 - a_{2i})]^2 + [\alpha_{3i}(x_1 - a_{1i}) - \alpha_{1i}(x_3 - a_{3i})]^2 + \\ + [\alpha_{1i}(x_2 - a_{2i}) - \alpha_{2i}(x_1 - a_{1i})]^2} \quad (3.14)$$

Сумма расстояний от точки  $X(x_1, x_2, x_3)$  до  $n$  прямых в трехмерном пространстве  $\sum_{i=1}^n l_i$  мажорируется параболоидом

$$\psi = \sum_{i=1}^n \psi_i(X) = \sum_{i=1}^n P_i^k \{ [\alpha_{2i}(x_3 - a_{3i}) - \alpha_{3i}(x_2 - a_{2i})]^2 + \\ + [\alpha_{3i}(x_1 - a_{1i}) - \alpha_{1i}(x_3 - a_{3i})]^2 + [\alpha_{1i}(x_2 - a_{2i}) - \alpha_{2i}(x_1 - a_{1i})]^2 \},$$

где

$$P_i^k = \left( \sqrt{[\alpha_{2i}(x_3^k - a_{3i}) - \alpha_{3i}(x_2^k - a_{2i})]^2 + [\alpha_{3i}(x_1^k - a_{1i}) - \alpha_{1i}(x_3^k - a_{3i})]^2 + [\alpha_{1i}(x_2^k - a_{2i}) - \alpha_{2i}(x_1^k - a_{1i})]^2} \right)^{-1}.$$

Отыскивая минимум этого параболоида, мы автоматически получаем координаты точки, в которой целевая функция  $f(X) = \sum_{i=1}^n l_i$  имеет меньшее значение, чем в начальной точке. Многократное повторение этого процесса приводит в точку минимума заданной функции.

В заключение рассмотрим комбинированную задачу — нахождение точки на минимальном суммарном расстоянии от  $n$  точек,  $m$  прямых и  $r$  плоскостей. Эта задача объединяет полученные ранее результаты.

Точкой, наименее удаленной от перечисленной выше совокупности объектов, будем считать

$$X = \arg \min_{x_1, x_2, x_3} \left\{ \sum_{i=1}^n w_{Ti} l_{Ti}(x_1, x_2, x_3) + \right. \\ \left. + \sum_{p=1}^m w_{\Delta p k} l_{\Delta p k}(x_1, x_2, x_3) + \sum_{j=1}^r w_{\Pi j} l_{\Pi j}(x_1, x_2, x_3) \right\}, \quad (3.15)$$

где  $L_i$  — расстояние от искомой точки до  $i$ -й заданной точки;  $l_{\text{прк}}$  — расстояние от искомой точки до  $v$ -й заданной прямой, определяемое по (3.14);  $l_{\text{пл}j}$  — расстояние от искомой точки до  $j$ -й заданной плоскости, определяемое по формуле (3.2);  $\omega_{Ti}$ ,  $\omega_{\text{прк}}$ ,  $\omega_{\text{пл}j}$  — коэффициенты важности.

Поскольку минимизацией отдельных слагаемых функционала (3.15) мы уже занимались, то без каких-либо пояснений напомним общую рекуррентную формулу для нахождения координат оптимальной точки в виде

$$X^{k+1} = \arg \min_x \left\{ \sum_{i=1}^n P_{1i}^k L_i + \sum_{v=1}^m P_{2v}^k M_v + \sum_{j=1}^r P_{3j}^k N_j \right\}, \quad (3.16)$$

где

$$L_i = (x_1 - a_{1i})^2 + (x_2 - a_{2i})^2 + (x_3 - a_{3i})^2,$$

$$P_{1i}^k = \omega_{Ti} \left( 2 \sqrt{(x_1^k - a_{1i})^2 + (x_2^k - a_{2i})^2 + (x_3^k - a_{3i})^2} \right)^{-1},$$

$$M_v = [\alpha_{2v}(x_3 - b_{3v}) - \alpha_{3v}(x_2 - b_{2v})]^2 + [\alpha_{3v}(x_1 - b_{1v}) - \\ - \alpha_{1v}(x_3 - b_{3v})]^2 + [\alpha_{1v}(x_2 - b_{2v}) - \alpha_{2v}(x_1 - b_{1v})]^2,$$

$$P_{2v}^k = \omega_{\text{прк}v} \left( 2 \sqrt{[\alpha_{2v}(x_3^k - b_{3v}) - \alpha_{3v}(x_2^k - b_{2v})]^2 + [\alpha_{3v}(x_1^k - b_{1v}) - \\ - \alpha_{1v}(x_3^k - b_{3v})]^2 + [\alpha_{1v}(x_2^k - b_{2v}) - \alpha_{2v}(x_1^k - b_{1v})]^2} \right)^{-1},$$

$$N_j = (c_{1j}x_1 + c_{2j}x_2 + c_{3j}x_3 + c_{4j})^2,$$

$$P_{3j}^k = \omega_{\text{пл}j} \left( \sqrt{c_{1j}^2 + c_{2j}^2 + c_{3j}^2} |c_{1j}x_1^k + c_{2j}x_2^k + c_{3j}x_3^k + c_{4j}| \right)^{-1}.$$

### 3.2. РАЗМЕЩЕНИЕ ТОЧЕЧНЫХ ОБЪЕКТОВ. МИНИМАКСНЫЕ ЗАДАЧИ

До сих пор мы рассматривали задачи, в которых требовалось найти точку или совокупность точек, находящихся на минимальном суммарном расстоянии от заданной совокупности объектов. Не менее часто на практике возникает необходимость решать задачи, в которых требуется минимизировать максимальное расстояние от размещаемой точки до любого из объектов. Например, требуемая мощность передающего устройства определяется расстоянием от передатчика до наиболее удаленного приемника. Размещение склада запасного имущества и приборов в заводском помещении может определяться соображениями минимизации времени на их доставку к любому рабочему месту. Места дислокации спасательных команд и дежурных постов также должны выбираться из условий минимизации максимального времени реагирования. Список таких примеров достаточно велик.

Из всего множества минимаксных задач размещения опишем подробно лишь одну, а именно минимаксную задачу Штейнера. Двумерный вариант этой задачи в литературе известен как задача о покрытии

$n$  заданных точек на плоскости кругом минимального радиуса [36]. В этой задаче требуется найти точку  $\mathbf{X} (x_1, x_2)$ , расположенную на минимаксном расстоянии от заданной совокупности точек  $\mathbf{A}_i (a_{i1}, a_{i2})$ ,  $i = 1, 2, \dots, n$ . Исключительно ради простоты будем считать, что все точки  $\mathbf{A}_i$  расположены в одной плоскости.

Общая идея решения опирается на тот почти очевидный факт, что минимаксная точка является решением задачи минимизации функции

$$f(x_1, x_2) = \sum_{i=1}^n |r_i|^m, \quad (3.17)$$

где  $r_i = \sqrt{(x_1 - a_{i1})^2 + (x_2 - a_{i2})^2}$  при достаточно больших  $m$ . Начав с малых  $m$  (например,  $m = 2$ ), найдя точку  $\hat{\mathbf{X}} = \arg \min_{\mathbf{x}} \sum_i |r_i|^2$ , используя ее в качестве стартовой при большем значении  $m$  и выполнив аналогичные действия и далее, мы в конце концов найдем требуемую точку минимакса. Поэтому основным содержанием алгоритма является решение задачи (3.17) при  $m > 2$ .

Как следует из примера 2.12, функция  $f(x_1, x_2) = r^m$ , в круге  $|\mathbf{x} - \mathbf{a}| \leq r^n$  мажорируется параболоидом

$$\psi(x_1, x_2) = A + [(x_1 - d_1)^2 + (x_2 - d_2)^2] B/2, \quad (3.18)$$

где

$$B = \max \begin{cases} f''(r^k) \\ f''(r^{nk}) \end{cases} = [m(m-1)] \max \begin{cases} (r^k)^{m-2}, \\ (r^{nk})^{m-2}, \end{cases}$$

где  $r^k = \|\mathbf{x}^k - \mathbf{a}\|$ ,  $r^{nk} = \|\mathbf{x}^{nk} - \mathbf{a}\|$ ,  $\mathbf{x}^k$  — текущая итеративная, а  $\mathbf{x}^{nk}$  — произвольная, лучше всего ньютоновская, точка.

Ньютоновскую точку на  $k$ -м шаге итерационного процесса в данной задаче можно найти из следующих соображений. В сечении  $[\mathbf{a}_i, \mathbf{x}^k]$  ньютоновская парабола имеет вид

$$\psi_i(s) = A_i + (s - C_i)^2 B_i/2,$$

где  $B_i = f''_{rr}(r_i^k) = m(m-1)(r_i^k)^{m-2}$ , а  $C_i$  делит отрезок  $[\mathbf{a}_i, \mathbf{x}^k]$  в отношении  $\alpha_i$ , где

$$\alpha_i = f'(r_i^k)/(r_i^k B_i) = 1/(m-1).$$

В силу этого ньютоновский параболоид функции  $f_i(x_1, x_2)$  имеет вид

$$\psi_i(x_1, x_2) = A_i + \frac{B_i}{2} (x_1 - d_{1i})^2 + \frac{B_i}{2} (x_2 - d_{2i})^2 = A_i + \frac{B_i}{2} \|\mathbf{x} - \mathbf{d}\|^2,$$

$$\mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = d_i \mathbf{a}_i + (1 - \alpha_i) \mathbf{x}^k = \mathbf{x} - \alpha_i (\mathbf{x}^k - \mathbf{a}_i).$$

Учитывая вид последней формулы, имеем

$$\min \psi(x_1, x_2) = \min \sum_i \psi_i = \sum_i A_i + \frac{1}{2} \min_{x_1} \sum_i B_i (x_1 - d_{1i})^2 + \\ + \frac{1}{2} \min_{x_2} \sum_i B_i (x_2 - d_{2i})^2. \quad (3.19)$$

В силу (3.19)  $x_1^{nk}$  и  $x_2^{nk}$  являются средневзвешенными значениями величин  $d_{1i}$  и  $d_{2i}$ , а сама ньютоновская точка имеет вид

$$x^{nk} = \frac{\sum_i B_i d_i}{\sum_i B_i} = \frac{\sum_i B_i [x^k - \alpha_i (x^k - a_i)]}{\sum_i B_i} = x^k - \frac{\sum_i B_i \alpha_i (x^k - a_i)}{\sum_i B_i} = \\ = x^k - \frac{1}{m-1} \frac{\sum_i B_i (x^k - a_i)}{\sum_i B_i} = x^k - \frac{1}{m-1} \frac{\sum_i (r_i)^{m-2} (x^k - a_i)}{\sum_i (r_i)^{m-2}}.$$

Зная ньютоновскую точку, в соответствии с примером 2.12 можно построить для каждого слагаемого мажорирующий параболоид

$$\psi_i = A_i + \|x - d_i\|^2 B_i / 2,$$

где

$$B_i = m(m-1) \max \begin{cases} r_i^{nk} \\ r_i^k \end{cases}$$

$$d_i = a_i \alpha_i + x(1 - \alpha_i), \quad \alpha_i = \frac{1}{m-1} \min \left\{ 1, \left( r_i^k / r_i^{nk} \right)^{m-2} \right\}.$$

Минимум суммы таких параболоидов определяется по формуле

$$x^{k+1} = x^k - \frac{1}{m-1} \frac{\sum_i (r_i^*)^{m-2} (x^k - a_i) \alpha_i}{\sum_i (r_i^*)^{m-2}} = \\ = x^k - \frac{1}{m-1} \frac{\sum_i (r_i)^{m-2} (x^k - a_i)}{\sum_i (r_i^*)^{m-2}},$$

$$r_i^* = \max \begin{cases} r_i^k \\ r_i^{nk} \end{cases} = \max \begin{cases} \|x^k - a_i\| \\ \|x^{nk} - a_i\| \end{cases}.$$

Нахождение точки  $x^{k+1}$  в пределах области мажорирования по каждому из слагаемых обеспечивается, как и в задаче отыскания чебышевских приближений (§ 2.3), тем, что увеличение любого  $B_i$  по

сравнению с  $f''(r^k)$  одновременно сокращает все разности  $|r_i^{n,k} - r_i^k|$ , а поскольку  $r_i^{n,k}$  и  $r_i^k$  по построению лежат в области мажорирования функции  $f_i(r_i) = r_i^m$  параболой  $\psi_i(r_i)$ , то и точка  $x^{k+1}$  лежит в области мажорирования функции  $f_i(x_1, x_2)$  параболоидом  $\psi_i(x_1, x_2)$  по каждому из слагаемых.

Подкупающая простота данного алгоритма дает ему определенные преимущества по сравнению с алгоритмом, обеспечивающим несколько более быструю сходимость, в котором скорректированные коэффициенты имеют вид

$$B_i = \max \left\{ \frac{m(m-1)(r_i^k)^{m-2}}{m \frac{(r_i^{n,k})^{m-1} - (r_i^k)^{m-1}}{r_i^{n,k} - r_i^k}} \right\} \quad (3.20)$$

и который вытекает из (1.13). И все же при построении рабочего алгоритма следует назначать веса по (3.20). В этом случае меньше вероятность преждевременного измельчения приращений в ходе итерационного процесса.

Минимаксная точка может быть найдена также в результате минимизации функции  $f(x_1, x_2) = \sum_i e^{\alpha r_i}$ , где  $\alpha$  — достаточно большое положительное число.

Последняя задача представляет и определенный самостоятельный интерес при размещении центрального объекта системы связи с учетом затуханий сигналов в линиях передач. Известно, что принимаемая мощность  $P_i$  связана с передаваемой мощностью соотношением  $A_i = P_i e^{-\alpha r_i}$ , где  $\alpha$  — коэффициент затухания. Если потребовать обеспечения определенного уровня мощности  $A$  на  $i$ -м приемном конце, то передаваемая в этом направлении мощность должна быть равна  $P_i = A e^{\alpha r_i}$ , а суммарная мощность источника сообщений  $P = \sum P_i = A \sum e^{\alpha r_i}$ .

Задачи такого рода без труда решаются методом ВВКМ, поскольку слагаемые  $\exp(\alpha r_i)$ , как показано в примере 2.13, мажорируются в пределах любого заданного круга, в том числе и такого, который определяется ньютоновской точкой  $x^{n,k}$ .

Метод ВВКМ позволяет также эффективно решать задачи смешанного типа, в которых показатель качества строится как комбинация функций различных видов. Например, элементарным обобщением рассмотренных задач является задача, в которой стоимость системы определяется затратами на прокладку кабельной сети, пропорциональными суммарной протяженности каналов, и затратами на создание источников тока, мощность которых определяется чувствительностью приемных устройств, их удалением от передающего центра и коэффициента-

ми затухания. При использовании других способов решения задач размещения [35, 39] возникают определенные трудности, как только в число слагаемых целевой функции начинают входить негладкие функции.

### 3.3. РАЗМЕЩЕНИЕ ЛИНЕЙНОГО ОБЪЕКТА

Рассмотрим задачу размещения линейного объекта на минимальном прямоугольном расстоянии от заданной совокупности точек. Задачи такого рода возникают, например, в случае, когда по густо застроенным кварталам современного города необходимо провести линию метрополитена, причем расположить ее нужно так, чтобы промышленные предприятия и учреждения находились от этой линии на минимальном расстоянии, определяемом с учетом допустимых перемещений в пределах городской черты.

Пусть на плоскости задана совокупность точечных объектов (точек)  $\{A_i(a_i, b_i), i = 1, 2, \dots, n\}$  с положительными весовыми коэффициентами  $w_i$ . Необходимо провести линию  $y = sx + t$  таким образом, чтобы сумма определяемых геометрией города расстояний от совокупности объектов-точек до этой линии были минимальна. При этом под «расстоянием в геометрии города» [1] от точки  $A_i(a_i, b_i)$  до прямой  $y = sx + t$  при условии совпадения направлений допустимых перемещений с осями системы координат будем понимать (рис. 3.3)

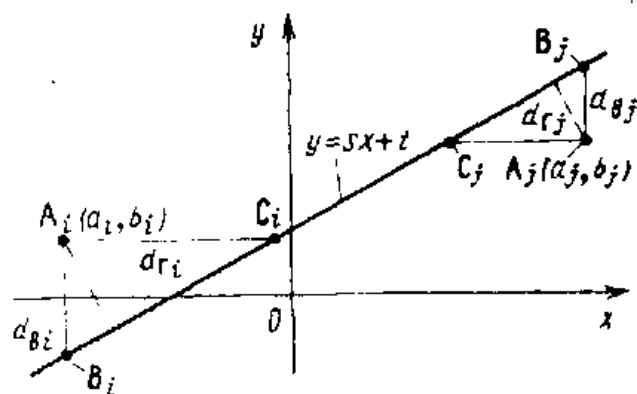


Рис. 3.3

$$l_i(s, t) = \min \{d_{bi}, d_{ri}\},$$

где  $d_{bi} = |b_i - sa_i - t|$  — расстояние от  $i$ -й точки до прямой с коэффициентами  $s$  и  $t$  по вертикальной оси;  $d_{ri} = |a_i - b_i/s + t/s|$  — соответствующее расстояние по горизонтальной оси.

С учетом сделанных замечаний задача минимизации суммы прямоугольных расстояний может быть сформулирована следующим образом:

$$(s, t) = \arg \min_{s, t} \sum_{i=1}^n w_i l_i(s, t), \quad (3.21)$$

где

$$l_i(s, t) = \min \{|b_i - sa_i - t|, |a_i - b_i/s + t/s|\}.$$

В такой постановке задача достаточно сложна для решения численными методами [39]. Однако нетрудно показать, что решение задачи (3.21) можно заменить последовательным решением двух более простых задач.

Снова обратимся к рис. 3.3. Пусть для точки  $A_i(a_i, b_i)$  имеем  $d_{ri} > d_{bi}$ . Вследствие подобия треугольников  $A_iB_iC_i$  и  $A_jB_jC_j$  это утверж-

дение должно быть справедливо и для любой другой точки  $A_i (a_i, b_i)$ . Таким образом, для  $\theta \in (-45^\circ, 45^\circ)$  это неравенство соблюдается на любом множестве точек. При  $\theta = -45^\circ$  и  $\theta = 45^\circ$  имеет место  $d_{bi} = d_{ri}$ . При всех остальных значениях  $\theta$  неизменно  $d_{bi} > d_{ri}$ . Следовательно, если решить задачу (3.21) сначала с расстоянием  $d_i(s, t) = |b_i - sa_i - t|$ , а затем с расстоянием  $d_i(s, t) = |a_i - b_i/s + t/s|$ , и выбрать в качестве решения вариант с меньшим значением целевой функции, то получим искомое решение. Причем каждая из подзадач легко решается методом ВВКМ.

В соответствии с примерами 2.3 и 2.4 для каждого из слагаемых

$$w_i l_i(s, t) = w_i |b_i - sa_i - t| \text{ и } w_i l_i(s, t) = w_i |a_i - b_i/s + t/s|$$

можно построить мажорирующие параболоиды

$$\psi_i(s, t) = \frac{w_i (l_i - sa_i - t)^2}{2 (|b_i - s^0 a_i - t^0|) + \epsilon}$$

и

$$\psi_i(s, t) = \frac{w_i (a_i - b_i/s + t/s)^2}{2 (|a_i - b_i/s^0 + t^0/s^0|) + \epsilon}$$

Рекуррентные формулы решения этих подзадач записываются в следующем виде:

$$(s^{k+1}, t^{k+1}) = \arg \min_{s, t} \sum_{i=1}^n P_{1i}^k (b_i - sa_i - t)^2, \quad (3.22)$$

$$(s^{k+1}, t^{k+1}) = \arg \min_{s, t} \sum_{i=1}^n P_{2i}^k (a_i - b_i/s - t/s)^2, \quad (3.23)$$

где

$$P_{1i}^k = \frac{w_i}{2 (|b_i - s^k a_i - t^k|) + \epsilon}, \quad P_{2i}^k = \frac{w_i}{2 (|a_i - b_i/s^k + t^k/s^k|) + \epsilon},$$

причем  $\epsilon$  — наперед заданная малая положительная величина.

#### 3.4. ЧИСЛОВЫЕ ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ РАЗМЕЩЕНИЯ МЕТОДОМ ВВКМ

**Пример 3.1.** Задано 10 плоскостей, коэффициенты уравнений которых сведены в табл. 3.1. Требуется найти точку, находящуюся на минимальном суммарном расстоянии от этих плоскостей.

Т а б л и ц а 3.1

| Номер<br>плос-<br>кости<br>$i$ | $A_{1i}$ | $A_{2i}$ | $A_{3i}$ | $A_{4i}$ | Номер<br>плос-<br>кости<br>$i$ | $A_{1i}$ | $A_{2i}$ | $A_{3i}$ | $A_{4i}$ |
|--------------------------------|----------|----------|----------|----------|--------------------------------|----------|----------|----------|----------|
| 1                              | 1,000    | 3,000    | 3,000    | 10,000   | 6                              | 9,000    | -1,000   | 2,000    | 5,000    |
| 2                              | 3,000    | -1,000   | 4,000    | 9,000    | 7                              | 7,000    | 1,000    | 1,000    | 4,000    |
| 3                              | 5,000    | 1,000    | 5,000    | 8,000    | 8                              | 5,000    | -1,000   | 0,000    | 3,000    |
| 4                              | 7,000    | -1,000   | 4,000    | 7,000    | 9                              | 3,000    | 1,000    | -1,000   | 2,000    |
| 5                              | 9,000    | 1,000    | 3,000    | 6,000    | 10                             | 1,000    | -1,000   | -2,000   | 1,000    |

Таблица 3.2

| Номер шага $k$ | $x_1^k$ | $x_2^k$ | $x_3^k$ | Суммарное расстояние | Номер шага $k$ | $x_1^k$ | $x_2^k$ | $x_3^k$ | Суммарное расстояние |
|----------------|---------|---------|---------|----------------------|----------------|---------|---------|---------|----------------------|
| 0              | 10,000  | 10,000  | 10,000  | 126,159              | 6              | -0,675  | -0,736  | -0,763  | 4,002                |
| 1              | -0,782  | -0,348  | -0,676  | 4,339                | 7              | -0,671  | -0,757  | -0,770  | 3,996                |
| 2              | -0,699  | -0,621  | -0,729  | 4,033                | 8              | -0,668  | -0,769  | -0,774  | 3,993                |
| 3              | -0,694  | -0,649  | -0,736  | 4,025                | 9              | -0,667  | -0,775  | -0,776  | 3,991                |
| 4              | -0,688  | -0,679  | -0,745  | 4,017                | 10             | -0,667  | -0,777  | -0,777  | 3,991                |
| 5              | -0,681  | -0,709  | -0,754  | 4,009                | 11             | -0,667  | -0,777  | -0,778  | 3,991                |

В качестве начальной точки выберем точку  $(x_1^0, x_2^0, x_3^0)$  такую, что  $x_1^0 = x_2^0 = x_3^0 = 10$ . Ход и результаты итерационного процесса приведены в табл. 3.2.

**Пример 3.2.** Для того чтобы получить наглядное графическое представление о процессе поиска оптимальной точки, рассмотрим двумерный аналог примера 3.1. Коэффициенты и ход итерационного процесса приведены в табл. 3.3 и 3.4 соответственно.

Таблица 3.3

| Номер плоскости (прямой) $i$ | $A_{1i}$ | $A_{2i}$ | $A_{4i}$ | Номер плоскости (прямой) $i$ | $A_{1i}$ | $A_{2i}$ | $A_{4i}$ |
|------------------------------|----------|----------|----------|------------------------------|----------|----------|----------|
| 1                            | 1,000    | 3,000    | 10,000   | 3                            | 5,000    | 1,000    | 8,000    |
| 2                            | 3,000    | -1,000   | 9,000    | 4                            | 7,000    | -1,000   | 7,000    |

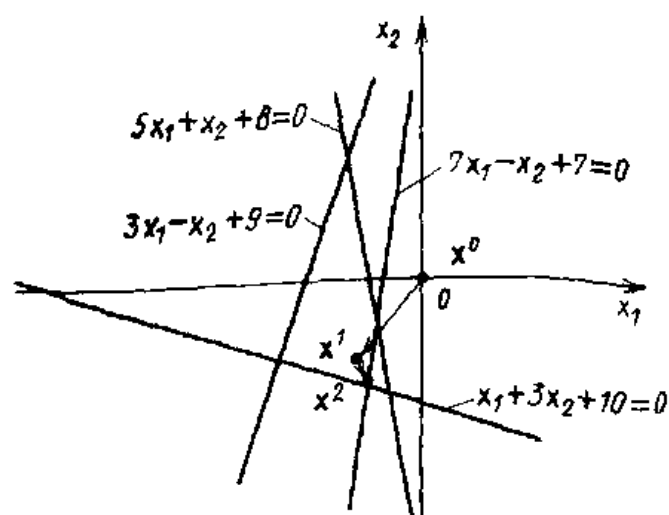
Таблица 3.4

| Номер шага $k$ | $x_1^k$ | $x_2^k$ | Суммарное расстояние | Номер шага $k$ | $x_1^k$ | $x_2^k$ | Суммарное расстояние |
|----------------|---------|---------|----------------------|----------------|---------|---------|----------------------|
| 0              | 0,000   | 0,000   | 8,567                | 6              | -1,406  | -2,838  | 2,801                |
| 1              | -1,689  | -2,022  | 3,473                | 7              | -1,407  | -2,850  | 2,796                |
| 2              | -1,524  | -2,455  | 3,106                | 8              | -1,408  | -2,857  | 2,793                |
| 3              | -1,434  | -2,672  | 2,918                | 9              | -1,409  | -2,860  | 2,791                |
| 4              | -1,405  | -2,770  | 2,841                | 10             | -1,409  | -2,862  | 2,790                |
| 5              | -1,403  | -2,816  | 2,813                | 11             | -1,409  | -2,862  | 2,790                |

На рис. 3.4 показана траектория перемещений точки минимума мажорирующего параболоида в ходе итерационного процесса. Как видно, для отыскания оптимальной точки  $(-1,409, -2,862)$  потребовалось совсем небольшое число шагов.

**Пример 3.3.** Рассмотрим задачу Штейнера — Вебера, исходные данные для которой представлены в табл. 3.5.

За начальную примем точку с координатами  $(0,0)$ . Результаты решения и ход итерационного процесса представлены в табл. 3.6 и на рис. 3.5. Кроме того, в таблице показан ход итерационного процесса решения этой же задачи методом Давидона — Флетчера (использовалась стандартная процедура FMFP из пакета научных подпрограмм на языке ПЛ/1).



Т а б л и ц а 3.5

| Точка            | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$   |
|------------------|-------|-------|-------|-------|---------|
| Координаты точки | (1,2) | (3,1) | (6,1) | (7,0) | (56,50) |

Рис. 3.4

Т а б л и ц а 3.6

| Номер шага $k$ | Метод ВВКМ |         |                      | Метод Давидона-Флетчера |         |                      |
|----------------|------------|---------|----------------------|-------------------------|---------|----------------------|
|                | $x_1^k$    | $x_2^k$ | Суммарное расстояние | $x_1^k$                 | $x_2^k$ | Суммарное расстояние |
| 0              | 0,000      | 0,000   | 93,554               | 0,000                   | 0,000   | 93,554               |
| 1              | 3,808      | 1,883   | 81,064               | 3,393                   | 2,510   | 82,286               |
| 2              | 4,256      | 1,409   | 80,489               | 3,413                   | 2,525   | 82,285               |
| 3              | 4,592      | 1,342   | 80,270               | 3,413                   | 2,525   | 82,285               |
| ...            | ...        | ...     | ...                  | ...                     | ...     | ...                  |
| 18             | 5,996      | 1,001   | 79,522               | 3,413                   | 2,525   | 82,285               |
| 19             | 5,998      | 1,001   | 79,521               | 3,413                   | 2,525   | 82,285               |
| 20             | 5,999      | 1,000   | 79,521               | 3,413                   | 2,525   | 82,285               |

Как видим, при одинаковых требованиях к точности определения минимума целевой функции метод Давидона — Флетчера дает более грубое приближение к оптимальному решению.

Необходимо отметить, что на практике могут встретиться подобные задачи с исключительно «пологой» целевой функцией, для решения которых необходимо

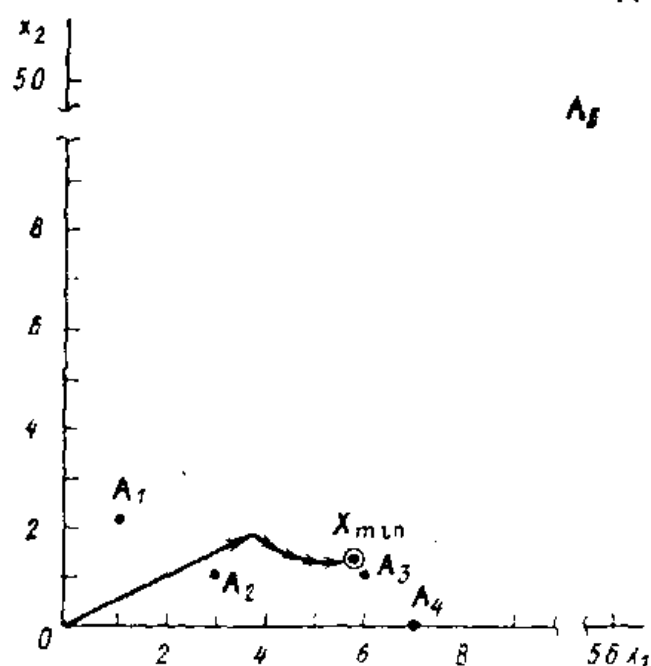


Рис. 3.5

привлекать специальные «овражные» методы [5] или разработать «овражную модификацию» метода ВВКМ. В качестве примера может послужить на первый взгляд простая задача Штейнера, приведенная в [36] и являющаяся своеобразным эталоном сложности задач данного типа. Даны четыре точки с координатами (0,0), (1,0), (0,1) и (100, 100), каждая с положительным весом, соответственно 1,1,1 и 3. Точным решением является точка (100, 100). Однако за разумное время счета при использовании алгоритма ВВКМ такого результата получить не удалось, потребовалось проделать 640 000 итераций, чтобы переместиться из точки  $x^0$  с координатами (1,1) в точку  $x^k$  с координатами  $x_1^k = x_2^k = 99,83249$ . И причина здесь вовсе не в плохой работе алгоритма, а

в исключительной «пологости» целевой функции. Отношение целевой функции в начальной и конечной точках составляет всего лишь  $f(1,1)/f(100, 100) = 1,001376915$ .

**Пример 3.4.** Заданы координаты пяти точек в четырехмерном пространстве (табл. 3.7). Требуется решить задачу Вебера — Штейнера.

Т а б л и ц а 3.7

| Точка | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
|-------|-------|-------|-------|-------|
| $A_1$ | 0,000 | 0,000 | 1,000 | 0,000 |
| $A_2$ | 2,000 | 2,000 | 2,000 | 2,000 |
| $A_3$ | 3,000 | 2,000 | 5,000 | 4,000 |
| $A_4$ | 4,000 | 5,000 | 2,000 | 4,000 |
| $A_5$ | 7,000 | 8,000 | 1,000 | 0,000 |

Ход итерационного процесса и результаты решения этой задачи методом ВВКМ представлены в табл. 3.8. В этой же таблице приведены результаты решения методом Давидона — Флетчера (использовалась стандартная процедура FMFR из пакета научных подпрограмм на языке ПЛ/1).

Т а б л и ц а 3.8

| Номер шага $k$ | Метод ВВКМ |         |         |         |                      | Метод Давидона-Флетчера |         |         |         |                      |
|----------------|------------|---------|---------|---------|----------------------|-------------------------|---------|---------|---------|----------------------|
|                | $x_1^k$    | $x_2^k$ | $x_3^k$ | $x_4^k$ | Суммарное расстояние | $x_1^k$                 | $x_2^k$ | $x_3^k$ | $x_4^k$ | Суммарное расстояние |
| 0              | 7,000      | 7,000   | 7,000   | 7,000   | 47,005               | 7,000                   | 7,000   | 7,000   | 7,000   | 47,005               |
| 1              | 3,453      | 3,662   | 2,402   | 2,355   | 19,992               | 2,798                   | 2,973   | 1,923   | 0,841   | 21,113               |
| 2              | 3,084      | 3,330   | 2,342   | 2,528   | 19,665               | 3,227                   | 3,429   | 2,218   | 1,960   | 19,939               |
| 3              | 2,928      | 3,087   | 2,358   | 2,469   | 19,565               | 3,227                   | 3,429   | 2,218   | 1,960   | 19,939               |
| 4              | 2,826      | 2,949   | 2,352   | 2,408   | 19,518               | 3,200                   | 3,400   | 2,200   | 2,000   | 19,891               |
| 5              | 2,753      | 2,857   | 2,337   | 2,362   | 19,493               | 3,200                   | 3,400   | 2,200   | 2,000   | 19,891               |
| ...            | ...        | ...     | ...     | ...     | ...                  | ...                     | ...     | ...     | ...     | ...                  |
| 13             | 2,566      | 2,636   | 2,269   | 2,247   | 19,463               | 3,200                   | 3,400   | 2,200   | 2,000   | 19,891               |
| 14             | 2,557      | 2,626   | 2,266   | 2,242   | 19,462               | 3,200                   | 3,400   | 2,200   | 2,000   | 19,891               |
| 15             | 2,551      | 2,618   | 2,263   | 2,238   | 19,462               | 3,200                   | 3,400   | 2,200   | 2,000   | 19,891               |

**Пример 3.5.** Рассмотрим задачу соединения  $m$  объектов на плоскости с помощью  $n$  узлов связи при заданной топологии сети. Для этой цели обозначим объекты  $A_i (a_i, b_i)$ , а узлы связи  $X_j (x_j, y_j)$ . Веса дуг между объектами и узлами связи и дуг соединяющих узлы связи, обозначим соответственно  $w_{ji}$  и  $v_{jv}$ . Суммарное взвешенное расстояние между всеми объектами сети имеет вид

$$f(X) = \sum_{1 \leq j \leq v \leq m} v_{jv} [(x_j - x_v)^2 + (y_j - y_v)^2]^{1/2} + \sum_{j=1}^m \sum_{i=1}^n w_{ji} [(x_j - a_i)^2 + (y_j - b_i)^2]^{1/2}$$

или короче

$$f(X) = \sum_{1 \leq j \leq v \leq m} \varphi_{jv} + \sum_{j=1}^m \sum_{i=1}^n \varphi_{ij}^1,$$

где

$$\varphi_{jv} = v_{jv} [(x_j - x_v)^2 + (y_j - y_v)^2]^{1/2},$$

$$\varphi_{ij}^1 = w_{ji} [(x_j - a_i)^2 + (y_j - b_i)^2]^{1/2}.$$

Для каждого из слагаемых  $\varphi_{jv}$  и  $\varphi_{ij}^1$  строим соответствующую мажоранту. Так, для слагаемых вида  $\varphi_{jv}$  мажорантами будут

$$\psi_{jv} = \frac{v_{jv} [(x_j - x_v)^2 + (y_j - y_v)^2]}{2 \left| \sqrt{(x_j^0 - x_v)^2 + (y_j^0 - y_v)^2} \right|},$$

а для слагаемых вида  $\varphi_{ij}^1$  —

$$\psi_{ij}^1 = \frac{w_{ji} [(x_j - a_i)^2 + (y_j - b_i)^2]}{2 \left| \sqrt{(x_j^0 - a_i)^2 + (y_j^0 - b_i)^2} \right|}.$$

С учетом этих соотношений можно записать ставшее уже привычным выражение итерационного процесса поиска координат оптимального размещения  $m$  объектов следующим образом:

$$X^{k+1} = \operatorname{argmin}_{x,y} \left\{ \sum_{1 \leq j \leq v \leq m} R_{jv}^k (x_j - x_v)^2 + \sum_{1 \leq j \leq v \leq m} R_{jv}^k (y_j - y_v)^2 + \right. \\ \left. + \sum_{j=1}^m \sum_{i=1}^n P_{ji}^k (x_j - a_i)^2 + \sum_{j=1}^m \sum_{i=1}^n P_{ji}^k (y_j - b_i)^2 \right\},$$

где

$$X = (X_1, X_2, X_3, \dots, X_m),$$

$$R_{jv}^k = \frac{v_{jv}}{2 \left| \sqrt{(x_j^k - x_v^k)^2 + (y_j^k - y_v^k)^2} \right|},$$

$$P_{ji}^k = \frac{w_{ji}}{2 \left| \sqrt{(x_j^k - a_i)^2 + (y_j^k - b_i)^2} \right|}.$$

Пусть требуется с помощью двух узлов связи  $X_1$  и  $X_2$  соединить объекты  $A_1(0,0)$ ;  $A_2(2,4)$ ;  $A_3(6,2)$ ;  $A_4(6,10)$ ;  $A_5(8,8)$ , причем матрицы весовых коэффициентов

$$\|w_{ji}\| = \begin{bmatrix} 4 & 2 & 3 & 0 & 0 \\ 0 & 2 & 1 & 3 & 2 \end{bmatrix}, \quad \|v_{12}\| = \begin{bmatrix} 2 \end{bmatrix}.$$

В качестве исходных точек примем  $X_1^0(0,0)$  и  $X_2^0(0,0)$ . Ход итерационного процесса и результаты решения отражены в табл. 3.9, топология оптимального расположения узлов сети показана на рис. 3.6.



Расположение отрезков и оптимальная точка показаны на рис. 3.7, а ход итерационного процесса представлен в табл. 3.11.

Представленный пример чрезвычайно прост. Читатель, заинтересованный в решении более сложных задач подобного типа, может обратиться к приложению. В нем содержатся текст и краткое описание соответствующей программы на языке ПЛ/1.

**Пример 3.7.** На трех гранях единичного куба проведены диагонали (рис. 3.8). Необходимо найти точку, минимально удаленную от этих диагоналей, и рассчитать суммарное расстояние от искомой точки до них. Координаты концов заданных диагоналей представлены в табл. 3.12, ход итерационного процесса и результаты решения задачи — в табл. 3.13.

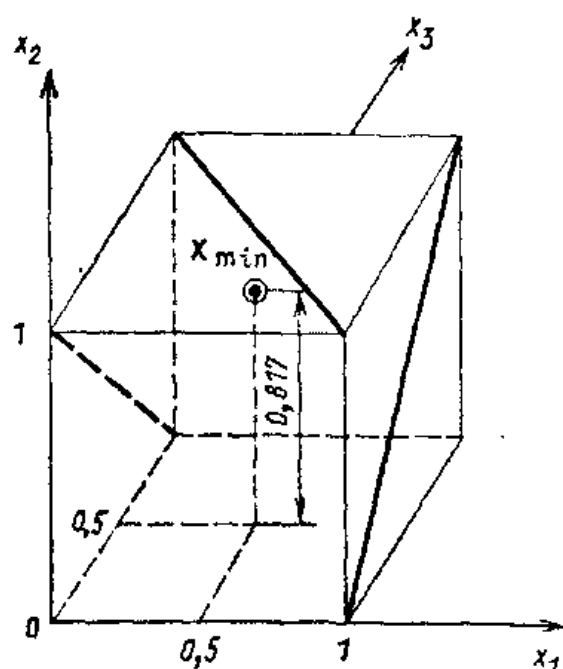


Рис. 3.8

Т а б л и ц а 3.12

| Номер диагонали | Координаты концов диагоналей |         |
|-----------------|------------------------------|---------|
| 1               | (0,0,1)                      | (0,1,0) |
| 2               | (1,1,0)                      | (0,1,1) |
| 3               | (1,0,0)                      | (1,1,1) |

Т а б л и ц а 3.13

| Номер шага $k$ | $x_1^k$ | $x_2^k$ | $x_3^k$ | Суммарное расстояние |
|----------------|---------|---------|---------|----------------------|
| 0              | 0,5     | 0,500   | 0,500   | 1,506                |
| 1              | 0,5     | 0,750   | 0,500   | 1,316                |
| 2              | 0,5     | 0,796   | 0,500   | 1,293                |
| 3              | 0,5     | 0,810   | 0,500   | 1,287                |
| 4              | 0,5     | 0,814   | 0,500   | 1,285                |
| 5              | 0,5     | 0,816   | 0,500   | 1,285                |
| 6              | 0,5     | 0,817   | 0,500   | 1,284                |
| 7              | 0,5     | 0,817   | 0,500   | 1,284                |

**Пример 3.8.** Заданы шесть точек, координаты которых приведены в табл. 3.14, и необходимо оптимальным образом расположить линейный объект (провести линию).

Т а б л и ц а 3.14

| Точка            | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ |
|------------------|-------|-------|-------|-------|-------|-------|
| Координаты точки | (1,3) | (7,2) | (9,1) | (8,8) | (9,3) | (1,5) |

Результаты решения задач по отысканию минимальных вертикальных и горизонтальных расстояний (рис. 3.9) приведены в табл. 3.15. В качестве окончательного решения выбираем вариант, полученный в первой задаче, так как суммарное расстояние в этом случае составляет 10,006, в то время как невязка для второй задачи 16,667.

Таблица 3.15

| Номер шага $k$ | Решение задачи (3.22) |       |                      | Решение задачи (3.23) |         |                      |
|----------------|-----------------------|-------|----------------------|-----------------------|---------|----------------------|
|                | $s^k$                 | $t^k$ | Суммарное расстояние | $s^k$                 | $t^k$   | Суммарное расстояние |
| 0              | 1,000                 | 1,000 | 27,000               | 1,000                 | 1,000   | 27,000               |
| 1              | 0,374                 | 3,178 | 16,721               | 1,295                 | -1,095  | 24,675               |
| 10             | -0,052                | 3,398 | 10,052               | 6,597                 | -44,274 | 16,697               |
| 11             | -0,046                | 3,357 | 10,046               | 6,386                 | -42,768 | 16,687               |
| 18             | 20,20                 | 3,157 | 10,020               | 6,021                 | -40,153 | 16,668               |
| 19             | -0,018                | 3,139 | 10,018               | 6,014                 | -40,102 | 16,667               |
| 20             | -0,016                | 3,123 | 10,016               | 6,009                 | -40,067 | 16,667               |
| 27             | -0,006                | 3,050 | 10,006               | 6,009                 | -40,067 | 16,667               |
| 28             | -0,006                | 3,044 | 10,006               | 6,009                 | -40,067 | 16,667               |
| ...            | ...                   | ...   | ...                  | ...                   | ...     | ...                  |

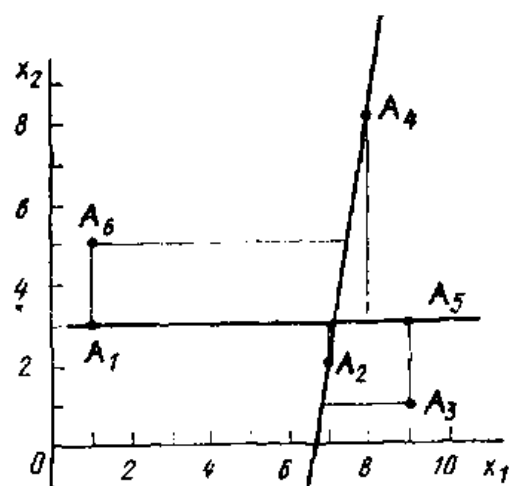


Рис. 3.9

**Пример 3.9.** Заданы пять точек в трехмерном пространстве (табл. 3.16). Требуется минимизировать максимальное расстояние от размещаемой точки до любой из заданных.

Таблица 3.16

| Точка            | $A_1$     | $A_2$        | $A_3$           | $A_4$               | $A_5$     |
|------------------|-----------|--------------|-----------------|---------------------|-----------|
| Координаты точки | (1; 1; 0) | (0; 0,95; 1) | (0,1; 0,1; 0,1) | (0,4; 0,345; 0,225) | (3; 4; 5) |

Формально задачу можно записать так: найти

$$\min_{\mathbf{X}} \max_i r_i, \quad i = 1, 2, \dots, n,$$

где  $\mathbf{X}$  — вектор координат искомой точки;  $r_i$  — расстояние от  $i$ -й точки до точки  $\mathbf{X}$ ;  $n$  — число точек ( $n = 5$ ).

Для решения воспользуемся алгоритмом, описанным в § 3.2, и заменим решение этой задачи решением задачи минимизации функции  $f_m(\mathbf{X}) = \sum_{i=1}^5 |r_i^m|$  при постоянно возрастающих значениях  $m$ . Результаты вычислений сведены в табл. 3.17.

В данном случае координаты минимаксной точки получены уже при  $m = 42$ . Счет был остановлен, когда два максимальных расстояния сравнялись по значению с точностью до 0,001.

В таблице помимо координат точки минимума функции  $f(\mathbf{X})$  и расстояний  $r_i$  от нее до каждой из заданных приведено число итераций, которое потребовалось для минимизации вспомогательной функции при каждом значении степени  $m$ .

Таблица 3.17

| $m$ | $x_1^m$ | $x_2^m$ | $x_3^m$ | $r_1$ | $r_2$ | $r_3$ | $r_4$ | $r_5$ | Число итераций |
|-----|---------|---------|---------|-------|-------|-------|-------|-------|----------------|
| 2   | 0,900   | 1,279   | 1,265   | 1,299 | 0,994 | 1,840 | 1,485 | 5,076 | 3              |
| 6   | 1,494   | 1,981   | 2,229   | 2,548 | 2,232 | 3,212 | 2,859 | 3,693 | 16             |
| 10  | 1,555   | 2,040   | 2,437   | 2,707 | 2,382 | 3,368 | 3,017 | 3,535 | 23             |
| 14  | 1,564   | 2,050   | 2,490   | 2,761 | 2,424 | 3,415 | 3,065 | 3,487 | 29             |
| 18  | 1,563   | 2,052   | 2,516   | 2,784 | 2,440 | 3,433 | 3,084 | 3,468 | 32             |
| 22  | 1,559   | 2,052   | 2,530   | 2,796 | 2,447 | 3,441 | 3,094 | 3,460 | 32             |
| 26  | 1,557   | 2,051   | 2,537   | 2,803 | 2,449 | 3,446 | 3,098 | 3,456 | 29             |
| 30  | 1,555   | 2,051   | 2,542   | 2,806 | 2,451 | 3,443 | 3,100 | 3,453 | 24             |
| 34  | 1,554   | 2,051   | 2,544   | 2,808 | 2,452 | 3,449 | 3,102 | 3,452 | 19             |
| 38  | 1,554   | 2,051   | 2,546   | 2,809 | 2,453 | 3,450 | 3,103 | 3,452 | 14             |
| 42  | 1,553   | 2,051   | 2,546   | 2,810 | 2,453 | 3,450 | 3,103 | 3,451 | 10             |

Следует отметить, что в общем случае минимаксное решение достигается где-то при  $m \approx 100$  [35], поэтому приходится иметь дело с большими числами и при составлении программ для ЭВМ нужно принимать необходимые меры, исключая возникновение ситуации «переполнения», или соответствующим образом масштабировать входные данные. Кроме того, теоретически не исключается возможность чрезмерного измельчения поправок  $\Delta X$  и  $\Delta r_{\max}$  при больших  $m$  еще до достижения минимаксной точки; в нашей практике таких случаев не встречалось.

### 3.5. ОДНА ЗАДАЧА ИЗ ОБЛАСТИ СТАТИКИ

К задачам размещения условно можно отнести и некоторые задачи механики. В качестве примера приведем задачу отыскания положения равновесия механизма, образованного пружинными балками (подвесами). Рассматриваются пружинные балки (подвесы), имеющие при отсутствии нагрузки длину  $l_i$  ( $i = 1, 2, \dots, n$ ). Коэффициент упругости  $i$ -й балки обозначим  $K_i$ . Эти балки закреплены одними своими концами в точках  $A_i (a_{i1}, a_{i2})$  ( $i = 1, 2, \dots, n$ ) и сочленены другими концами между собой в общей точке  $X = (x_1, x_2)$  (рис. 3.10). К точке  $X$  приложена нагрузка  $F (f_1, f_2)$ . Требуется найти положение равновесия.

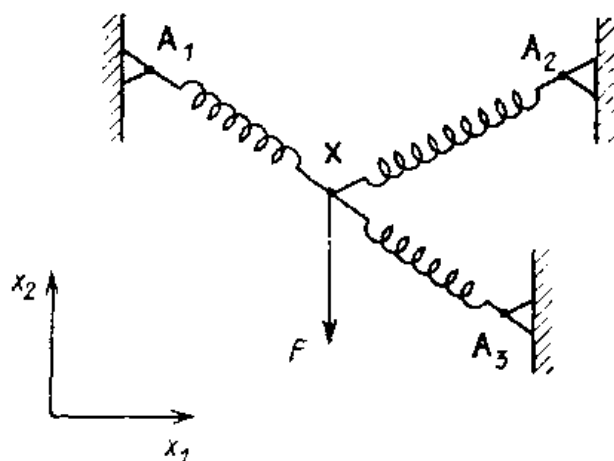


Рис. 3.10

Примем за начало координат положение точки  $X$  при отсутствии нагрузки, а ось  $x_2$  направим в направлении, противоположном направлению нагрузки. В этих условиях потенциальная энергия системы равна

$$\begin{aligned} \Phi(x_1, x_2) = & \sum_i \frac{K_i}{2} [\sqrt{(x_1 - a_{i1})^2 + (x_2 - a_{i2})^2} - \\ & - l_i]^2 - f_2 x_2 = \sum_{i=1}^n \frac{K_i}{2} [(x_1 - a_{i1})^2 + (x_2 - a_{i2})^2 + \\ & + l_i^2 - 2l_i \sqrt{(x_1 - a_{i1})^2 + (x_2 - a_{i2})^2}] - f_2 x_2. \end{aligned} \quad (3.24)$$

Минимум потенциальной энергии достигается в точке равновесия. Заметим, что если даны два вектора

$$X^k - A = \begin{pmatrix} x_1^k - a_1 \\ x_2^k - a_2 \end{pmatrix} \text{ и } X - A = \begin{pmatrix} x_1 - a_1 \\ x_2 - a_2 \end{pmatrix},$$

то косинус угла  $\alpha$  между ними можно определить по формуле

$$\cos \alpha = \frac{(x_1^k - a_1)(x_1 - a_1) + (x_2^k - a_2)(x_2 - a_2)}{\sqrt{(x_1^k - a_1)^2 + (x_2^k - a_2)^2} \sqrt{(x_1 - a_1)^2 + (x_2 - a_2)^2}} \leq 1, \quad (3.25)$$

откуда

$$\sqrt{(x_1 - a_1)^2 + (x_2 - a_2)^2} \geq \frac{(x_1^k - a_1)(x_1 - a_1) + (x_2^k - a_2)(x_2 - a_2)}{\sqrt{(x_1^k - a_1)^2 + (x_2^k - a_2)^2}}.$$

При  $x_1 = x_1^k, x_2 = x_2^k$  левая и правая части этого неравенства равны между собой. Нетрудно проверить, что одновременно при этом условии равны между собой и их градиенты. Поэтому квадратичная функция

$$\begin{aligned} \psi(X, X^k) = & \sum_{i=1}^n \frac{K_i}{2} \left[ (x_1 - a_{i1})^2 + (x_2 - a_{i2})^2 - \right. \\ & - 2l_i \frac{(x_1^k - a_{i1})(x_1 - a_{i1})}{\sqrt{(x_1^k - a_{i1})^2 + (x_2^k - a_{i2})^2}} - \\ & \left. - 2l_i \frac{(x_2^k - a_{i2})(x_2 - a_{i2})}{\sqrt{(x_1^k - a_{i1})^2 + (x_2^k - a_{i2})^2}} + l_i^2 \right] - f_2 x_2 \end{aligned} \quad (3.26)$$

мажорирует функцию (3.24) с соблюдением всех условий, требуемых методом ВВКМ.

Формулы, аналогичные (3.26), имеют место и в более общем случае, когда балки сочленяются не в одной, а в нескольких точках [40].



Если имеется достаточно хорошее начальное приближение  $\theta^0$ , то, разложив  $\xi_i(\theta_1, \theta_2, \dots, \theta_n)$  в ряд Тейлора и введя обозначения

$$x_j = \theta_j - \theta_j^0, \quad b_i = \tilde{b}_i - \xi_0, \quad a_{ij} = \left. \frac{\partial \xi_i}{\partial \theta_j} \right|_{\theta = \theta^0},$$

систему уравнений (4.2) можно переписать в виде

$$\sum_j a_{ij} x_j + \Delta_i = b_i, \quad i = 1, 2, \dots, N. \quad (4.3)$$

Обработать измерения — значит извлечь из них сведения об интересующем нас предмете или явлении, в данном случае — определить значения величин  $\theta_1, \theta_2, \dots, \theta_n$  из системы (4.2) или величин  $x_1, x_2, \dots, x_n$  из системы (4.3).

Система уравнений (4.2) является не вполне обычной системой, решаемой с помощью аппарата высшей алгебры. С одной стороны, число неизвестных в ней превышает число уравнений. Помимо величин  $\theta_1, \theta_2, \dots, \theta_n$  неизвестными являются также значения ошибок  $\Delta_1, \Delta_2, \dots, \Delta_N$ . Система, как говорят, недоопределена. Если более или менее произвольно выбрать значения  $\theta_1, \theta_2, \dots, \theta_n$ , то значения  $\Delta_1, \Delta_2, \dots, \Delta_N$  можно найти из соотношений

$$\Delta_i = \tilde{b}_i - \xi_i(\theta_1, \theta_2, \dots, \theta_n), \quad i = 1, 2, \dots, N.$$

В этом случае величина  $\Delta_i$  называется остаточной невязкой  $i$ -го измерения.

Ввиду недоопределенности систем алгебраических уравнений (4.2) и (4.3) их решение ищут не в обычном алгебраическом, а в статистическом смысле, а именно стараются так подобрать значения неизвестных величин  $\theta_1, \theta_2, \dots, \theta_n$ , чтобы совокупность остаточных невязок обладала бы какими-то экстремальными свойствами. Для этой цели составляется некоторая функция  $\Phi(\Delta_1, \Delta_2, \dots, \Delta_N)$  от невязок и минимизируется (иногда максимизируется) в пространстве неизвестных величин  $\theta_1, \theta_2, \dots, \theta_n$  (или соответственно  $x_1, x_2, \dots, x_n$ , если речь идет о поправках к начальному приближению).

Если известна функция плотности распределения ошибок  $\Delta_i$  измерений, то в качестве функции  $\Phi(\Delta_1, \Delta_2, \dots, \Delta_N)$  теория рекомендует выбирать функцию правдоподобия  $P(\Delta_1, \Delta_2, \dots, \Delta_N)$ , где  $P$  — плотность совместного распределения ошибок. Оценки, полученные на этой основе, называются оценками максимального правдоподобия. При большом количестве измерений эти оценки являются в известном смысле наилучшими, так как обладают асимптотической несмещенностью, состоятельностью и асимптотической эффективностью (т. е. минимальной дисперсией).

Если предположить, что все ошибки измерений независимы друг от друга, то

$$P(\Delta_1, \Delta_2, \dots, \Delta_N) = \prod_{i=1}^N p_i(\Delta_i). \quad (4.4)$$

В этом случае для получения оценок максимального правдоподобия нужно максимизировать (4.4) или ее логарифм

$$\Phi(\Delta_1, \Delta_2, \dots, \Delta_N) = \sum_{i=1}^N \ln p_i(\Delta_i). \quad (4.5)$$

Вид функции  $\Phi$  выбирается исходя из предположений о происхождении ошибок измерения. Часто этим предположениям отвечает кусочно-выпуклая функция. Такое положение возникает, например, при засорении общего массива измерений помехами [20,34] или при наличии в измерительных приборах заметной вилки нечувствительности. В этих случаях метод ВВКМ оказывается удобным средством отыскания точки  $\theta$ , доставляющей максимум функции  $\Phi$ .

При решении многих трудно формализуемых задач помимо физических измерений широко используется экспертная информация. Для удобства информация, полученная от экспертов, также подвергается необходимой обработке. Чаще всего целью обработки мнений экспертов является выработка группового решения. Например, определение группового предпочтения можно рассматривать как выбор точки в некотором пространстве на минимальном расстоянии от совокупности точек, соответствующих предпочтениям отдельных экспертов.

По своей сути экспертиза — это разновидность измерения. В современной трактовке к измерениям относят не только установление численного соотношения неизвестной величины и единицы измерения. В широком смысле под понятие измерения подпадает любое высказывание, сделанное по отношению к объектам эмпирического множества и установленное опытным путем. При таком подходе ошибки измерений связаны с построением неправильных высказываний. Обработка измерений в данном случае сводится к обработке этих высказываний путем минимизации специальным образом построенной функции штрафа.

Часто в качестве функции штрафа выбирают логарифм апостериорной вероятности гипотезы, взятый со знаком «минус». Если апостериорная вероятность гипотезы мала, то штраф велик, и наоборот. И здесь функция, подлежащая минимизации, далеко не всегда обладает гладкостью, дифференцируемостью и другими свойствами, допускающими применение стандартных алгоритмов\*).

---

\*) При написании § 4.1 — 4.3 авторы широко пользовались советами кандидата технических наук В. Л. Кушко.

Рассмотрим класс загрязненных распределений. Этот класс образуется так. С вероятностью  $1 - \epsilon$  измерения подчинены известному закону  $\psi(b, \theta)$ , но с вероятностью  $\epsilon$  они могут быть подчинены любому другому неизвестному обработчику закону распределения  $r(b, \theta)$  (загрязняющее распределение). Таким образом, результирующая функция распределения

$$f(b, \theta) = (1 - \epsilon) \psi(b, \theta) + \epsilon r(b, \theta). \quad (4.6)$$

В [34] Хьюбер (в другой транскрипции Хубер) показал, что наилучшим вариантом загрязняющей функции является функция

$$r(z) = \begin{cases} \psi(z_1) \exp[k(z - z_1)] - \psi(z), & \text{если } z \leq z_1, \\ 0, & \text{если } z_1 \leq z \leq z_2, \\ \psi(z_2) \exp[-k(z - z_2)] - \psi(z), & \text{если } z \geq z_2, \end{cases} \quad (4.7)$$

а соответствующая  $\epsilon$ -загрязненная функция имеет вид

$$f_0(z) = \begin{cases} (1 - \epsilon) \psi(z_1) \exp[k(z - z_1)], & \text{если } z \leq z_1, \\ (1 - \epsilon) \psi(z), & \text{если } z_1 \leq z \leq z_2, \\ (1 - \epsilon) \psi(z_2) \exp[-k(z - z_2)], & \text{если } z \geq z_2, \end{cases} \quad (4.8)$$

где  $k$  связано определенным образом с уровнем загрязнения,  $z = b - \sum a_j \theta_j$ , а  $z_1$  и  $z_2$  определяются из уравнения  $|\psi'/\psi| = k$ . Для симметричных  $\psi(z)$  всегда  $z_1 = -z_2$ . Уровень загрязнения определяется из опыта как процент отбракованных измерений.

Если загрязняется нормальное распределение  $p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{z^2}{2\sigma^2}\right)$ , то  $\epsilon$ -загрязненная функция выглядит, как показано на рис. 4.1, а. Ее логарифм приведен на рис. 4.1, б (с точностью до постоянного слагаемого).

При загрязнении, приближающемся к 100 %-ному, функция, изображенная на рис. 4.1, б, вырождается в функцию модуля величины  $z$ , (рис. 4.2), где  $z_i = b_i - \sum_j a_{ij} x_j$ ,  $x_j = \theta_j - \theta_j^0$ .

Усечение сверху того или иного закона распределения соответствует предположению о равновероятности, а следовательно, и о равноправности любого результата измерения внутри некоторого числового интервала  $[-A, A]$ , именуемого «вилкой нечувствительности» (рис. 4.3).

Если в качестве загрязняемых распределений рассмотреть множество усеченных нормальных распределений, то при малой степени загрязнения график функции  $\ln p_i(z)$  имеет вид, показанный на рис. 4.4, а, при большой степени загрязнения — на рис. 4.4, б.

Функции, представленные на рис. 4.2—4.4, без труда мажорируются квадратичными параболой, правила вычисления коэффициентов

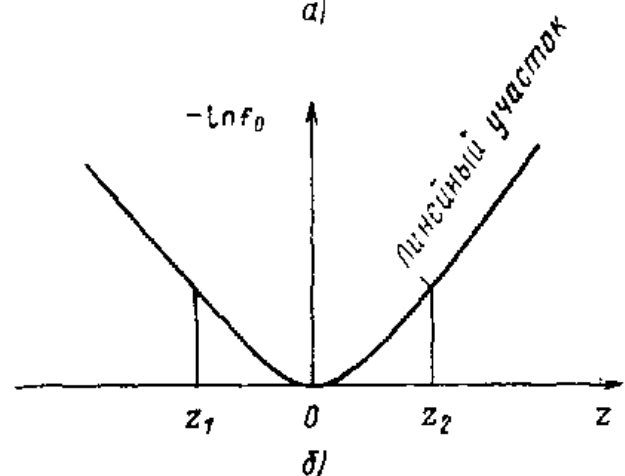
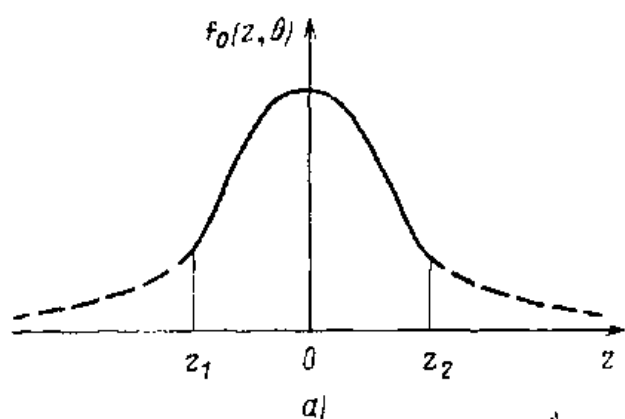


Рис. 4.1

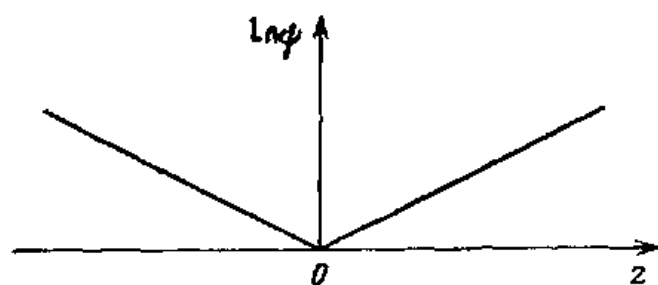


Рис. 4.2

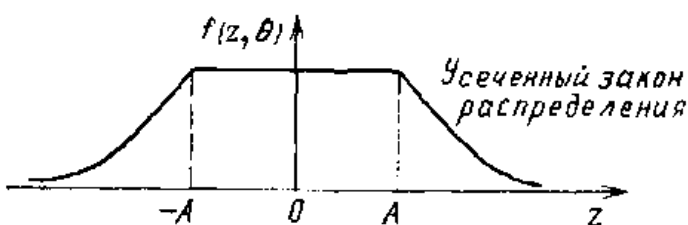
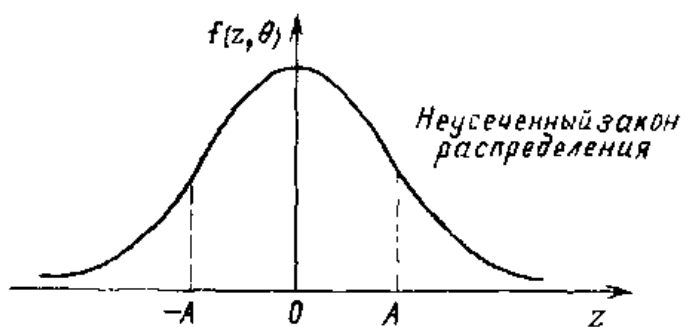


Рис. 4.3

которых описаны в гл. 2. Учитывая возможную несимметричность функции распределения, указанные функции удобно представить в виде суммы

$$\varphi(z) = \varphi_1(z) + \varphi_2(-z),$$

где

$$\varphi_{1,2}(z) = \begin{cases} \varphi(z), & \text{если } z \geq 0, \\ 0, & \text{если } z < 0. \end{cases}$$

Примеры функций  $\varphi(z)$  приведены на рис. 4.5.

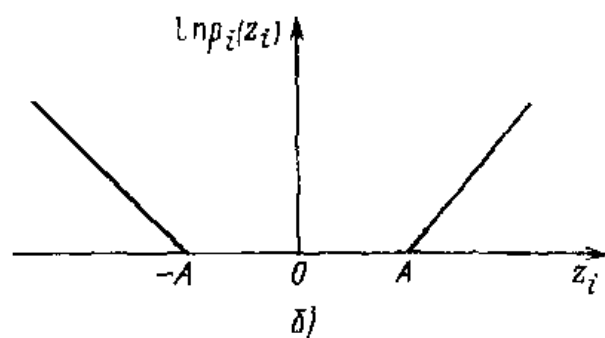
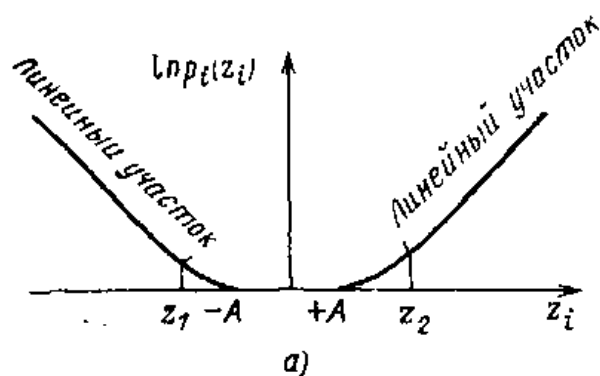


Рис. 4.4

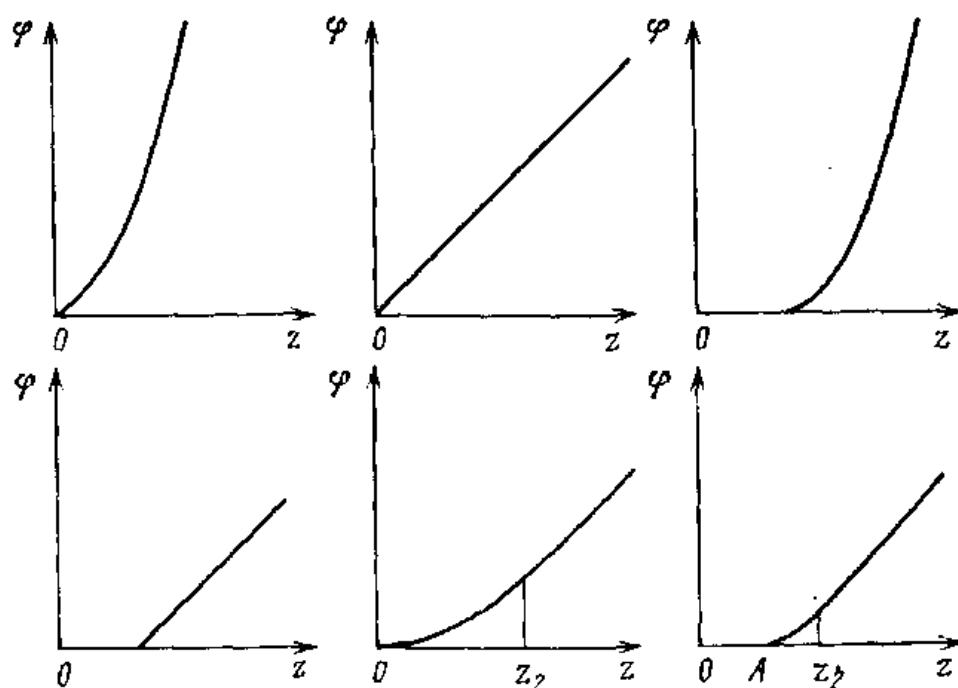


Рис. 4.5

Таким образом, метод ВВКМ становится удобным средством отыскания устойчивых оценок хуберовского типа. Оценки эти имеют вид

$$\theta_{\text{хуб}} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^n \{ -\ln f_{\varepsilon_i}(z_i) \},$$

где  $f_{\varepsilon_i}(z_i)$  — функция плотности, полученная в результате наихудшего загрязнения;  $z_i = b_i - \sum_j a_{ij}x_j$ ,  $x_j = \theta_j - \theta_j^0$ ;  $\mathbf{b} = \|\mathbf{b}_j\|$  — вектор измерений;  $\theta = \|\theta_j\|$  — вектор неизвестных параметров;  $\theta^0 = \|\theta_j^0\|$  — начальное приближение этого вектора.

#### 4.2. ВИДЫ ЭКСПЕРТНЫХ ДАННЫХ

Наиболее распространенными являются следующие виды экспертных данных [14]:

- а) приблизительная оценка количественного значения неизвестной величины;
- б) оценка диапазона, в котором располагается неизвестная величина;
- в) попарное сравнение величин по их «ценности», в результате которого отдается предпочтение одному из них или признается их неразличимость с этой точки зрения;
- г) ранжирование по ценности всех или части объектов;
- д) ранжирование разности ценности для всех или некоторых пар объектов;
- е) определение мер близости для всех или некоторых пар объектов.

В результате обработки экспертных данных либо приписываются числовые характеристики неизвестным объектам либо производится группирование объектов по признаку близости. Первая задача называется задачей шкалирования, вторая — задачей кластеризации. Задача шкалирования может быть одномерной (объекту присваивается скалярная числовая характеристика) или многомерной (объекту присваивается векторная числовая характеристика). Многомерное шкалирование, при котором каждый объект характеризуется точкой  $n$ -мерного пространства, может стать основой, на которой в последующем решается задача кластеризации.

Перечисленные выше виды экспертных оценок могут быть представлены математическими соотношениями следующих трех типов:

- 1) соотношениями типа равенств;
- 2) соотношениями типа неравенств;
- 3) так называемыми ненаправленными соотношениями, характеризующими сходство (различие) объектов.

*Соотношения типа равенств* имеют вид

$$\theta_i + \xi_k = a_{ik}, \quad (4.9)$$

где  $\theta_i$  — ценность  $i$ -го объекта;  $a_{ik}$  — оценка ценности  $i$ -го объекта  $k$ -м экспертом;  $\xi_k$  — ошибка экспертизы.

*Соотношения типа неравенств* могут иметь различный вид в зависимости от их происхождения. Например, при указании диапазона, в котором находится  $\theta_i$ , указанные неравенства имеют вид

$$a_{ik}^{(1)} \leq \theta_i + \xi_k \leq a_{ik}^{(2)}, \quad (4.10)$$

где  $a_{ik}^{(1)}$  и  $a_{ik}^{(2)}$  — верхняя и нижняя границы диапазона, в котором по оценке  $k$ -го эксперта находится величина  $\theta_i$ .

Если в ходе экспертизы установлено, что  $\theta_i < \theta_j$  ( $<$  — знак предпочтения), то это означает, что

$$\theta_i - \theta_j + \xi_{ij} \leq -\epsilon, \quad (4.11)$$

где  $\theta_i$  и  $\theta_j$  — «ценности» объектов, участвовавших в парном сравнении;  $\epsilon$  — «порог чувствительности»;  $\xi_{ij}$  — ошибка экспертизы.

Если по оценке эксперта  $A_i \approx A_j$  (« $\approx$ » — знак неразличимости), то

$$-\epsilon \leq \theta_i - \theta_j + \xi_{ij} \leq \epsilon. \quad (4.12)$$

Если эксперт оценивал разности ценностей, то соотношения, аналогичные (4.10) и (4.11), приобретают вид

$$(\theta_j - \theta_i) - (\theta_k - \theta_l) + \xi \leq -\epsilon, \quad (4.13)$$

$$-\epsilon \leq (\theta_i - \theta_j) - (\theta_k - \theta_l) + \xi \leq \epsilon. \quad (4.14)$$

«Ненаправленные» соотношения, в которых оценивается модуль разности «ценности» объектов или модуль удаления от эталона, имеют вид

$$|\theta_i - \theta_j| \pm \xi = d_{ij} \quad (4.15)$$

или

$$|\theta_i - a| \pm \xi = d_i, \quad (4.16)$$

где  $\theta_i$  и  $\theta_j$  — «ценности» объектов, участвовавших в сравнении;  $a$  — «ценность» эталонного объекта;  $d_i$ ,  $d_{ij}$  — мера близости между объектами по оценке эксперта;  $\xi$  — ошибка экспертизы.

Несмотря на внешнюю похожесть соотношений всех трех типов, по своей внутренней сути они заметно отличаются друг от друга и при их обработке требуют совершенно различных подходов. Данные типа (4.9) обрабатываются так же, как и измерения в их обычной форме, т. е. путем максимизации функции квазиправдоподобия [20]. Данные типа неравенств и «ненаправленные» соотношения типа (4.15), (4.16) обрабатываются специальными методами. Описанию некоторых из них и посвящена настоящая глава.

#### 4.3. ОБРАБОТКА ЭКСПЕРТНЫХ ДАННЫХ, ПРЕДСТАВЛЕННЫХ В ФОРМЕ НЕРАВЕНСТВ

Указание диапазонов, в которых может находиться случайная величина, данные попарных сравнений, частичных и полных ранжировок обрабатываются в соответствии с рядом оригинальных методик, объединенных общим названием «линейные модели» (модели Терстоуна — Мостеллера, Бредли — Терри, Рао — Купера, Гленна — Дэвида и т. д.). В основе этих методик, так же как и при обработке измерений в обычной форме, лежит идея максимизации функций квазиправдоподобия, в которые, наряду с дифференциальными характеристиками распределений (функциями плотности), входят вероятности появления данной совокупности исходов, т. е. интегральные характеристики распределений.

Приняв линейную модель, будем исходить из следующей вероятностной схемы. Имеется множество объектов  $\{A_1, A_2, \dots, A_n\}$ , которому однозначно соответствует множество ценностей  $\theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ . Мы пишем, что  $A_i > A_j$ , если  $i$ -й объект предпочитается  $j$ -му объекту, и  $A_i \approx A_j$ , если они неразличимы по своей ценности. Разница  $\theta_{ij} = \theta_i - \theta_j$  между соответствующими ценностями измеряется экспертом с некоторой ошибкой  $\xi$ , подчиненной закону распределения  $F(z)$ . Причиной появления этой ошибки могут быть: а) незнание или невозможность определения зависимости ценности объекта от его свойств; б) размытость самого понятия «ценность объекта»; в) неоднородность или нестабильность предъявляемых образцов; г) несовершенство человека как измерительного инструмента.

Предполагается, что, наблюдая величину  $y_{ij} = \theta_{ij} + \xi$ , эксперт осознанно или неосознанно руководствуется следующим правилом: если  $|y_{ij}| \leq \epsilon$ , где  $\epsilon$  — некоторая фиксированная величина, он принимает решение о неразличимости объектов; если  $y_{ij} \geq \epsilon$  — о превосходстве  $i$ -го объекта над  $j$ -м; если  $y_{ij} \leq -\epsilon$  — о превосходстве  $j$ -го объекта. Обработчику предъявляются конечные результаты этих сравнений. На основании этих данных требуется приписать числовые характеристики ценности каждому из объектов.

Указанная модель и называется линейной. В рамках этой модели вероятность того, что при разовом сопоставлении  $i$ -го и  $j$ -го объектов лучшим будет признан  $i$ -й объект, равна

$$P_{ij}^1 = P(A_i > A_j) = P(\theta_i - \theta_j + \xi \geq \epsilon) = P(\xi \geq \epsilon + \theta_j - \theta_i) = \\ = 1 - F(\theta_j - \theta_i + \epsilon) = H(\theta_j - \theta_i + \epsilon),$$

где  $H(z) = 1 - F(z)$ ,  $F(z)$  — функция распределения ошибок экспертизы.

Если  $F(z)$  симметрична, то

$$P_{ij}^1 = H(\theta_j - \theta_i + \epsilon) = F(\theta_i - \theta_j - \epsilon).$$

Вероятность того, что  $i$ -й объект будет признан худшим, по причине симметрии  $F(z)$  равна

$$P_{ij}^2 = P(\theta_i - \theta_j + \xi \leq -\epsilon) = F(\theta_j - \theta_i - \epsilon).$$

Наконец, вероятность того, что они будут признаны равноценными, равна

$$P_{ij}^3 = P(-\epsilon \leq \theta_i - \theta_j + \xi \leq \epsilon) = F(\theta_i - \theta_j + \epsilon) - F(\theta_i - \theta_j - \epsilon).$$

Если  $i$ -й и  $j$ -й объекты сопоставлялись между собой  $t_{ij}$  раз, причем  $k_{ij}$  раз лучшим признавался  $i$ -й объект,  $l_{ij}$  —  $j$ -й объект и  $m_{ij}$  раз они признавались неразличимыми, то вероятность такого сложного события

$$P_{ij} = \frac{(t_{ij})!}{(k_{ij})! (l_{ij})! (m_{ij})!} (P_{ij}^1)^{k_{ij}} (P_{ij}^2)^{l_{ij}} (P_{ij}^3)^{m_{ij}}. \quad (4.17)$$

Если взять всю совокупность попарных сравнений в целом, то вероятность появления каждой конкретной реализации будет пропорциональна

$$P = \prod_{i>j} (P_{ij}^1)^{k_{ij}} (P_{ij}^2)^{l_{ij}} (P_{ij}^3)^{m_{ij}}. \quad (4.18)$$

Для определения  $\theta_1, \theta_2, \dots, \theta_n$  из опыта надо максимизировать по  $\theta$  выражение (4.18) или с тем же результатом его логарифм  $L$ :

$$L = \log P = \sum_{i>j} (k_{ij} \log P_{ij}^1 + l_{ij} \log P_{ij}^2 + m_{ij} \log P_{ij}^3). \quad (4.19)$$

Поскольку во все предыдущие формулы входят только разности, то ясно, что каждая величина  $\theta_i$  определяется с точностью до произвольного постоянного слагаемого. Кроме того, половину размера вилки нечувствительности  $\varepsilon$  можно принять за единицу масштаба.

Предположим, например, что функцией плотности случайной величины  $\xi$  является

$$f(z) = F'(z) = \frac{1}{2\beta} \begin{cases} \exp(z/\beta), & \text{если } z < 0, \\ \exp(-z/\beta), & \text{если } z \geq 0, \end{cases} \quad (4.20)$$

где  $\beta$  — параметр закона распределения, так что

$$F(z) = \frac{1}{2} \begin{cases} \exp(|z|/\beta), & \text{если } z < 0, \\ 1 - \exp(-|z|/\beta), & \text{если } z \geq 0. \end{cases} \quad (4.21)$$

В этих условиях [22]

$$\begin{aligned} \log P_{ij}^1 &= \log \frac{1}{2} - \frac{1}{\beta} \times \\ &\times \begin{cases} \theta_j - \theta_i + \varepsilon, & \text{если } \theta_j - \theta_i + \varepsilon \geq 0, \\ -\beta \log \left[ 2 - \exp \left( -\frac{|\theta_j - \theta_i + \varepsilon|}{\beta} \right) \right], & \text{если } \theta_j - \theta_i + \varepsilon < 0; \end{cases} \end{aligned} \quad (4.22)$$

$$\begin{aligned} \log P_{ij}^2 &= \log \frac{1}{2} - \frac{1}{\beta} \times \\ &\times \begin{cases} \theta_i - \theta_j + \varepsilon, & \text{если } \theta_i - \theta_j + \varepsilon \geq 0, \\ -\beta \log \left[ 2 - \exp \left( -\frac{|\theta_i - \theta_j + \varepsilon|}{\beta} \right) \right], & \text{если } \theta_i - \theta_j + \varepsilon < 0; \end{cases} \end{aligned} \quad (4.23)$$

$$\begin{aligned} \log P_{ij}^3 &= \log \frac{1}{2} \left[ 1 - \exp \left( -\frac{2\varepsilon}{\beta} \right) \right] - \frac{1}{\beta} \times \\ &\times \begin{cases} -|\theta_i - \theta_j| + \varepsilon, & \text{если } |\theta_i - \theta_j| \geq \varepsilon, \\ -\beta \log \frac{\exp(\varepsilon/\beta) - \operatorname{ch}(|\theta_i - \theta_j|/\beta)}{\operatorname{sh}(-\varepsilon/\beta)}, & \text{если } |\theta_i - \theta_j| < \varepsilon. \end{cases} \end{aligned} \quad (4.24)$$

Оценкой максимального правдоподобия является точка  $\theta_1, \theta_2, \dots, \theta_n$ , доставляющая максимум функции (4.19), т. е. функции

$$\begin{aligned} L &= \sum_{i>j} (k_{ij} + l_{ij}) \log \frac{1}{2} + \sum_{i>j} m_{ij} \log \frac{1}{2} \left[ 1 - \exp \left( -\frac{2\varepsilon}{\beta} \right) \right] - \\ &- \frac{1}{\beta} \sum_{i>j} (k_{ij} \varphi_{ij}^1 + l_{ij} \varphi_{ij}^2 + m_{ij} \varphi_{ij}^3), \end{aligned} \quad (4.25)$$

где

$$\varphi_{ij}^1 = \begin{cases} \theta_j - \theta_i - \varepsilon, & \text{если } \theta_j - \theta_i - \varepsilon > 0, \\ -\beta \log \left[ 2 - \exp \left( -\frac{|\theta_j - \theta_i - \varepsilon|}{\beta} \right) \right], & \text{если } \theta_j - \theta_i - \varepsilon < 0; \end{cases} \quad (4.26)$$

$$\varphi_{ij}^2 = \begin{cases} \theta_i - \theta_j - \varepsilon, & \text{если } \theta_i - \theta_j - \varepsilon \geq 0, \\ -\beta \log \left[ 2 - \exp \left( -\frac{|\theta_i - \theta_j - \varepsilon|}{\beta} \right) \right], & \text{если } \theta_i - \theta_j - \varepsilon < 0; \end{cases} \quad (4.27)$$

$$\varphi_{ij}^3 = \begin{cases} -|\theta_i - \theta_j| + \varepsilon, & \text{если } |\theta_i - \theta_j| \geq \varepsilon, \\ -\beta \log \frac{\exp(\varepsilon/\beta) - \operatorname{ch}(|\theta_i - \theta_j|/\beta)}{\operatorname{sh}(\varepsilon/\beta)}, & \text{если } |\theta_i - \theta_j| < \varepsilon. \end{cases} \quad (4.28)$$

Графики этих функций показаны на рис. 4.6.

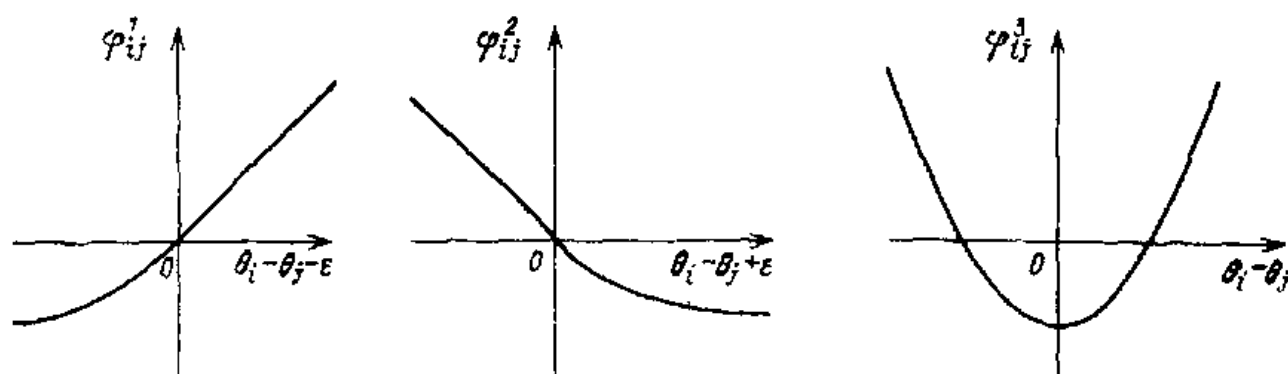


Рис. 4.6

Поскольку первые две суммы, входящие в (4.25), от  $\theta_1, \theta_2, \dots, \theta_n$  не зависят, то для получения оценки максимального правдоподобия необходимо оптимизировать последнее слагаемое.

Заметим при этом, что  $\varphi_{ij}^3$  можно представить в виде суммы  $\varphi_{ij}^3 = \varphi_{ij}^{31} + \varphi_{ij}^{32} - D$ , в которой

$$\varphi_{ij}^{31} = \begin{cases} \theta_j - \theta_i - \varepsilon, & \text{если } \theta_j - \theta_i - \varepsilon \geq 0, \\ -\beta \log \frac{\exp(\varepsilon/\beta) - \operatorname{ch}(|\theta_j - \theta_i - \varepsilon|/\beta)}{\operatorname{sh}(\varepsilon/\beta)}, & \text{если } -\varepsilon \leq \theta_j - \theta_i - \varepsilon \leq 0, \\ D, & \text{если } \theta_j - \theta_i - \varepsilon < -\varepsilon; \end{cases} \quad (4.29)$$

$$\varphi_{ij}^{32} = \begin{cases} \theta_i - \theta_j - \varepsilon, & \text{если } \theta_i - \theta_j - \varepsilon \geq 0, \\ -\beta \log \frac{\exp(\varepsilon/\beta) - \operatorname{ch}(|\theta_i - \theta_j - \varepsilon|/\beta)}{\operatorname{sh}(\varepsilon/\beta)}, & \text{если } -\varepsilon \leq \theta_i - \theta_j - \varepsilon \leq 0, \\ D, & \text{если } \theta_i - \theta_j - \varepsilon < -\varepsilon; \end{cases} \quad (4.30)$$

$$D = -\beta \log \frac{\exp(\varepsilon/\beta) - \operatorname{ch}(\varepsilon/\beta)}{\operatorname{sh}(\varepsilon/\beta)}. \quad (4.31)$$

При малых  $\beta$  все слагаемые, входящие в выражение  $L^* = \sum_{i>j} (k_{ij} \times \times \varphi_{ij}^1 + l_{ij} \varphi_{ij}^2 + m_{ij} \varphi_{ij}^3)$ , имеют одинаковый вид:

$$\varphi_s(y_s) = \begin{cases} y_s, & \text{если } y_s \geq 0, \\ 0, & \text{если } y_s < 0, \end{cases} \quad (4.32)$$

где

$$y_s = \sum_t a_{st} \theta_t + b_s, \quad (4.33)$$

а коэффициенты  $a_{st}$  и  $b_s$  определяются результатами сопоставления объектов.

Пусть, например, сопоставлялись  $i$ -й и  $j$ -й объекты и сильнейшим (более ценным) признан  $i$ -й объект. Тогда  $a_{si} = -1$ ,  $a_{sj} = +1$ ,  $b_s = +1$ , все остальные коэффициенты равны нулю. В случае победы  $j$ -го объекта  $a_{si} = +1$ ,  $a_{sj} = -1$ ,  $b_s = +1$ .

Пусть сопоставлялись  $i$ -й и  $j$ -й объекты и в результате сопоставления объекты признаны неразличимыми. Поскольку функция  $\varphi_{ij}^3$  распадается на две функции  $\varphi_{ij}^{31}$  и  $\varphi_{ij}^{32}$  (плюс постоянное слагаемое  $D$ , не зависящее от  $\theta$ ), то в  $L^*$  появляются два слагаемых вида (4.32), у которых соответственно

$$\begin{aligned} a_{si} &= -1, a_{sj} = +1, b_s = -1 \\ a_{si} &= +1, a_{sj} = -1, b_s = -1 \end{aligned} \quad (i, j = 1, 2, \dots, n).$$

Каждая функция вида (4.32), как показано в гл. 1, может быть мажорирована параболой

$$\psi_s(y_s) = A_s^k - (y_s - C_s)^2 B_s^k / 2, \quad (4.34)$$

где

$$A_s^k = 0, B_s^k = (2|y_s^k|)^{-1}, C_s^k = |y_s^k|, y_s^k = \sum_t a_{st} \theta_t^k + b_s.$$

В качестве начального значения  $\theta_1^0, \theta_2^0, \dots, \theta_n^0$  может быть принят любой набор чисел. Затем в соответствии с общей идеологией вариационно-взвешенных квадратических мажорант строится итерационная процедура, где каждое новое приближение к  $\theta$  есть результат минимизации по  $\theta$  квадратичной формы

$$K = (Y - C^k) P^k (Y - C^k)^T, \quad (4.35)$$

где

$$Y = \|y_s\| = \left\| \sum_t a_{st} \theta_t + b_s \right\|, C^k = \|c_s^k\| = \|-|y_s^k|\|,$$

а  $P^k = \|p_{st}^k\|$  — диагональная матрица с элементами  $p_{st}^k = (4|y_s^k|)^{-1}$  вдоль главной диагонали, так что

$$\theta^{k+1} = \arg \min_{\theta} (Y - C^k) P^k (Y - C^k)^T. \quad (4.36)$$

Сходимость этой процедуры доказана ранее.

Сделаем два важных замечания.

*Замечание 1.* Поскольку все  $\theta_i$  определяются с точностью до произвольной постоянной, то одна из этих величин (например,  $\theta_1$ ) должна быть приравнена нулю и в дальнейшем исключена из числа неизвестных. Это обстоятельство должно быть обязательно учтено при использовании стандартной процедуры метода наименьших квадратов для отыскания минимума суммы мажорирующих парабол.

*Замечание 2.* Если на каком-нибудь этапе итерационного процесса окажется, что одно из  $y_s^k$  равно нулю, то соответствующий диагональный элемент в  $P_k$  должен быть заменен достаточно большим числом, как это следует из общих соображений, положенных в основу ВВКМ.

После того как найден максимум функции правдоподобия с помощью формул (4.32)—(4.36), т. е. в предположении малости  $\beta$ , появляется возможность оценить снизу значение  $\beta$  и воспользоваться более точными формулами (4.26)—(4.31). Нижняя оценка может быть получена при помощи таких рассуждений.

Если  $\theta_{ij} = \theta_i - \theta_j < \varepsilon$  ( $\theta_i > \theta_j$ ), то при отсутствии ошибок экспертизы и в соответствии с принятым решающим правилом результатом «соревнования» между объектами должна быть ничья. Если это не так, то в эксперимент вкралась ошибка, не меньшая по своему абсолютному значению, чем  $-\theta_i + \theta_j + \varepsilon$ , если предпочтение отдано  $i$ -му участнику, и чем  $\theta_i - \theta_j + \varepsilon$ , если —  $j$ -му участнику.

Аналогично, если  $\theta_i \geq \theta_j + \varepsilon$ , то результатом сравнения при отсутствии ошибок экспертизы должна быть победа  $i$ -го объекта. Если это не так, то в эксперимент вкралась ошибка, не меньшая  $\theta_i - \theta_j - \varepsilon$ , если зафиксирована ничья, и не меньшая  $\theta_i - \theta_j + \varepsilon$ , если зафиксирована победа  $j$ -го объекта.

Если результат сравнения соответствует «силе» участников, то нижняя оценка модуля ошибки равна нулю. Находя среднее арифметическое всех нижних границ, получаем нижнюю оценку величины  $\beta$ , поскольку математическое ожидание модуля случайной величины, подчиненной двойному экспоненциальному закону, равно  $\beta$ .

Заметим, что при использовании формул (4.26) — (4.30) функции  $\varphi_{ij}^1$  и  $\varphi_{ij}^2$  имеют одинаковый вид, т. е.

$$\varphi_s^*(y_s) = \begin{cases} y_s, & \text{если } y_s \geq 0, \\ -\beta \log [2 - \exp(-y_s/\beta)], & \text{если } y_s < 0, \end{cases} \quad (4.37)$$

где  $y_s = \theta_j - \theta_i + \varepsilon$  — в случае победы  $i$ -го участника над  $j$ -м;  $y_s = -\theta_i - \theta_j + \varepsilon$  — в противоположном случае.

Так же одинаковый вид

$$\varphi_s^{**}(y_s) = \begin{cases} y_s, & \text{если } y_s \geq 0, \\ -\beta \log \frac{\exp(\varepsilon/\beta) - \operatorname{ch}(|y_s|/\beta)}{\operatorname{sh}(\varepsilon/\beta)}, & \text{если } y_s \leq 0, \end{cases} \quad (4.38)$$

имеют функции  $\varphi_{ij}^{31}$  и  $\varphi_{ij}^{32}$ , зависящие соответственно от аргументов  $y_s = \theta_j - \theta_i - \varepsilon$  и  $y_s = \theta_i - \theta_j - \varepsilon$ .

В качестве коэффициентов мажорирующих парабол (как это следует из гл.1) могут быть назначены

$$B^k = \frac{2}{\beta}, \quad C^k = -|y_s^k| - \frac{\beta}{2} \frac{\exp(|y_s^k|/\beta)}{2 - \exp(|y_s^k|/\beta)}, \quad \text{если } y_s^k < 0, \quad (4.39)$$

$$B^k = (2|y_s^k|)^{-1}, \quad C^k = -|y_s^k|, \quad \text{если } y_s^k > 0,$$

по отношению к функции (4.37) и

$$B^k = (2|y_s^k - D|)^{-1}, \quad C^k = -|y_s^k|, \quad \text{если } y_s^k \leq -\varepsilon,$$

$$B^k = \frac{2}{1 - \exp(-2\varepsilon/\beta)}, \quad C^k = -|y_s^k| - \frac{1}{B^k} \frac{\operatorname{sh}((y_s^k + \varepsilon)/\beta)}{\exp(\varepsilon/\beta) - \operatorname{ch}((y_s^k + \varepsilon)/\beta)}, \quad \text{если } -\varepsilon \leq y_s^k \leq 0, \quad (4.40)$$

$$B^k = (2|y_s^k|)^{-1}, \quad C^k = -|y_s^k|, \quad \text{если } y_s^k > 0,$$

по отношению к функции (4.38).

При переходе от малых  $\beta$  к ее средним и большим значениям может измениться и ранжировка объектов. При малых (по сравнению с  $\varepsilon$ )  $\beta$  легко объяснить появление ничьих, но трудно объяснить появление нетранзитивных троек типа:  $A_i > A_j$ ,  $A_j > A_k$ ,  $A_k > A_i$ . При больших (по сравнению с  $\varepsilon$ )  $\beta$  легко объясняется это явление, но трудно-объяснимыми становятся ничьи.

**Пример 4.1.** Рассмотрим, например, некоторый гипотетический турнир [27]. Итоговая турнирная таблица имеет вид рис. 4.7.

На рисунке справа от таблицы приведены очки, набранные участниками, и их ранжировка, установленная по общепринятой методике. Автор [27] произвел так называемое тонкое ранжирование участников турнира с помощью чисел Бержа

$$b_i = \lim_{n \rightarrow \infty} \left( d_i^n / \sum_k d_k^n \right),$$

где  $d_i^n$  — «сила»  $n$ -го порядка, определяемая как сумма элементов, стоящих в  $i$ -й строке матрицы  $M^n = \underbrace{M \cdot M \cdot \dots \cdot M}_{n \text{ раз}}$  ( $M$  — матрица результатов: 1 — в случае победы,  $1/2$  — в случае ничьей, 0 — в случае поражения). Эта тонкая ранжировка выглядит так: *dfhabceg*.

Шкалирование, основанное на идее минимизации по  $\theta$  выражения

$$\Phi(\theta) = \sum_s \varphi_s(y_s), \quad (4.41)$$

где  $\varphi_s(y_s)$ , в зависимости от результатов встречи, принимает вид (4.37) или (4.38), при малых  $\beta$  привело к следующей шкале:  $\theta_a = 0, \theta_b = -1,001, \theta_c = -1,063, \theta_d = 1,019, \theta_e = -2,106, \theta_f = 1,007, \theta_g = -2,007, \theta_h = 0,605$  с ранжировкой *dfhabcege*.

Оценка снизу величины  $\beta$  дала  $\beta > 0,178$ . Минимизация выражения (4.41) при этом  $\beta$  привела к шкале:  $\theta_a = 0$ ,  $\theta_b = -1,006$ ,  $\theta_c = -1,230$ ,  $\theta_d = 0,987$ ,  $\theta_e = -2,404$ ,  $\theta_f = 1,137$ ,  $\theta_g = -2,230$ ,  $\theta_h = 0,526$  с ранжировкой  $fdhabcge$ .

Анализ полученных результатов показывает, что при  $\beta \gg \epsilon$  на первое место вышел  $f$ . При ранжировании по относительной силе на первое место вышел  $d$ , поскольку  $d$  обыграл более сильного противника  $a$ , в то время как  $f$  сделал с ним ничью. Но ничем не хуже и такие рассуждения, ставящие  $f$  на первое место:  $d$  не сумел обыграть слабого противника  $h$  (удовольствовавшись ничьей), которого, как и надлежит лидеру, обыграл  $f$ .

|          | <i>a</i>      | <i>b</i> | <i>c</i> | <i>d</i>      | <i>e</i> | <i>f</i>      | <i>g</i> | <i>h</i>      | Очки           | Место |
|----------|---------------|----------|----------|---------------|----------|---------------|----------|---------------|----------------|-------|
| <i>a</i> | X             | 1        | 1        | $\frac{1}{2}$ | 1        | 0             | 1        | 0             | $4\frac{1}{2}$ | 4     |
| <i>b</i> | 0             | X        | 1        | 0             | 0        | 0             | 1        | 0             | 2              | 5-6   |
| <i>c</i> | 0             | 0        | X        | 0             | 1        | 0             | 1        | 0             | 2              | 5-6   |
| <i>d</i> | $\frac{1}{2}$ | 1        | 1        | X             | 1        | $\frac{1}{2}$ | 1        | 1             | 6              | 1-2   |
| <i>e</i> | 0             | 1        | 0        | 0             | X        | 0             | 0        | 0             | 1              | 7-8   |
| <i>f</i> | 1             | 1        | 1        | $\frac{1}{2}$ | 1        | X             | 1        | $\frac{1}{2}$ | 6              | 1-2   |
| <i>g</i> | 0             | 0        | 0        | 0             | 1        | 0             | X        | 0             | 1              | 7-8   |
| <i>h</i> | 1             | 1        | 1        | 0             | 1        | $\frac{1}{2}$ | 1        | X             | $5\frac{1}{2}$ | 3     |

Рис. 4.7

Ранжирование участников в рамках линейной модели обладает, по крайней мере, тем преимуществом, что допускает вероятностную трактовку. Согласившись с определенным характером закона распределения ошибок экспертизы, мы обязаны согласиться и с конечным результатом. Если результат нам кажется сомнительным, надо более тщательно рассмотреть модель, приведшую к этому результату.

Одним из преимуществ способов шкалирования, основанных на минимизации (4.41), является то, что они позволяют обрабатывать данные незаконченных турниров, у которых турнирная таблица содержит незаполненные клеточки. Причины этого чаще всего носят экономический характер, желание сэкономить временные и материальные ресурсы. Но бывают и другие причины. Например, нетрудно представить ситуацию, когда эксперт, осуществляющий парное сравнение, попросту незнаком с частью рассматриваемых объектов и поэтому он воздерживается от заполнения некоторых клеток матрицы парных сравнений.

Пусть  $i$ -й и  $j$ -й участники турнира сыграли между собой  $m_{ij}$  партий и, кроме того, во всех встречах  $i$ -й участник набрал  $s_i$  очков. Тогда существует единственный набор чисел  $x_{ij}$  (шансы команд), который удовлетворяет трем условиям:

$$1) \ x_{ij} = \frac{1}{x_{ji}}, \quad 2) \ x_{ih} = x_{ij} x_{jh}, \quad 3) \ p_{ij} = \frac{x_{ij}}{1 + x_{ij}}, \quad (4.42)$$

где  $p_{ij}$  — вероятность победы  $i$ -го противника над  $j$ -м. Условие 3 выражения (4.42) вытекает из следующего определения понятия «шансы»:

$$x_{ij} = p_{ij}/p_{ji} = p_{ij}/(1 - p_{ij}).$$

Полагая  $x_{ij} = \exp(\theta_i - \theta_j)$ , мы удовлетворяем первым двум условиям в (5.34). Чтобы удовлетворить третьему условию, мы должны положить

$$p_{ij} = \frac{\exp(\theta_i - \theta_j)}{1 + \exp(\theta_i - \theta_j)} = \frac{1}{1 + \exp(\theta_j - \theta_i)} = \frac{1}{1 + \exp(-z)}, \quad (4.43)$$

где через  $z$  обозначена разность  $\theta_i - \theta_j$  сил любых двух участников. Учитывая, что в рамках линейной модели при  $\epsilon = 0$

$$p_{ij} = H(z) = \int_{-z}^{\infty} f(x) dx, \quad (4.44)$$

после дифференцирования (4.43) по  $z$  получаем

$$\begin{aligned} f(z) &= \frac{d}{dz} p_{ij} = \frac{d}{dz} \left( \frac{1}{1 + \exp(-z)} \right) = \frac{\exp(-z)}{(1 + \exp(-z))^2} = \\ &= \left( 4 \left( \frac{\exp(-z/2) + \exp(z/2)}{2} \right)^2 \right)^{-1} = \frac{1}{4} \operatorname{sech}^2 \frac{z}{2}. \end{aligned} \quad (4.45)$$

Таким образом, если закон распределения ошибки экспертизы имеет вид (4.45), а стало быть справедлива линейная модель Бредли — Терри, то вероятность победы  $i$ -го участника над  $j$ -м, определяемая по (4.44), будет иметь вид  $p_{ij} = [1 + \exp(\theta_j - \theta_i)]^{-1}$ , а «шансы» на победу в одном матче — вид  $x_{ij} = \exp(\theta_i - \theta_j)$ . Если допустить возможность ничьих (модель Гленна — Дэвида), то при

$$f(z) = \frac{1}{4} \operatorname{sech}^2 \frac{z}{2}, \quad H(z) = \int_{-z}^{\infty} f(z) dz = \frac{1}{1 + \exp(-z)},$$

$$F(z) = 1 - H(z) = \frac{\exp(-z)}{1 + \exp(-z)}$$

имеет место

$$p_{ij}^1 = F(\theta_j - \theta_i - \epsilon) = (1 + \exp[-(\theta_i - \theta_j) + \epsilon])^{-1}, \quad (4.46)$$

$$p_{ij}^2 = F(\theta_i - \theta_j - \epsilon) = (1 + \exp[-(\theta_j - \theta_i) + \epsilon])^{-1}, \quad (4.47)$$

$$\begin{aligned} p_{ij}^3 &= 1 - p_{ij}^1 - p_{ij}^2 = 1 - \frac{1}{1 + \exp(-z + \epsilon)} - \frac{1}{1 + \exp(z + \epsilon)} = \\ &= \frac{\operatorname{sh} \epsilon}{\operatorname{ch} \epsilon + \operatorname{ch} z}. \end{aligned} \quad (4.48)$$

## Функция правдоподобия

$$L = A + \sum_{i>j} k_{ij} \log [1 + \exp [-(\theta_i - \theta_j) + \epsilon]] +$$

$$+ l_{ij} \log [1 + \exp [-(\theta_j - \theta_i) + \epsilon]] + m_{ij} \log [\operatorname{ch} \epsilon + \operatorname{ch} (\theta_i - \theta_j)], \quad (4.49)$$

где  $A = -m_{ij} \log \operatorname{sh} \epsilon = \text{const}$ .

Результатом эксперимента является определенное число побед, ничьих и поражений во встречах отдельных участников и, следовательно, значения коэффициентов  $k_{ij}$ ,  $l_{ij}$  и  $m_{ij}$  в (4.49). Отдельные слагаемые, входящие в (4.49), имеют вид, изображенный на рис. 4.8.

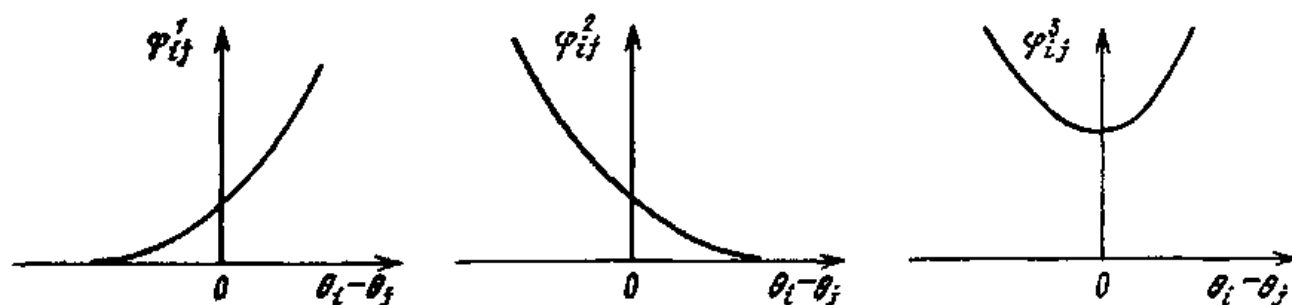


Рис. 4.8

Нетрудно заметить, что графики этих функций очень похожи на графики функций штрафа для двойного экспоненциального закона распределения, изображенных на рис. 4.6, что, конечно, неудивительно, если учесть происхождение этих функций и сходство двойного экспоненциального и гиперсеканс-квадрат-распределений.

В самом деле, функция плотности распределения (4.45) симметрична относительно нуля, вблизи точки  $z = 0$  имеет вид  $f(z) \approx (1 - z^2)/2$  (т. е. похожа на плотность нормального распределения) и обладает экспоненциальными хвостами, так как при  $z \rightarrow \infty$  функция  $f(z) \approx \exp(-z)$ , а при  $z \rightarrow -\infty$  функция  $f(z) \approx \exp z$ . Таким образом, функция плотности гиперсеканс-квадрат-распределения напоминает функцию Хубера (см. § 4.1) и принадлежит, по-видимому, к семейству помехоустойчивых распределений.

Ввиду того, что все функции (4.46) — (4.48) достаточно гладкие и обладают сильной выпуклостью в любом конечном диапазоне, к ним применим метод Ньютона в чистом виде, когда функция (4.49) и все ее слагаемые аппроксимируются (а не мажорируются) квадратичной параболой.

Так, функция  $\varphi^1(z) = \log [1 + \exp (z + \epsilon)]$  аппроксимируется параболой  $\psi(z) = A^k + (z - C^k)^2 B^k / 2$ , где значение  $A^k$  не представляет интереса;

$$B^k = 0,25 \operatorname{sech}^2 [(z + \epsilon)/2],$$

$$C^k = z^k - (B^k [1 + \exp (-z^k - \epsilon)])^{-1}, \quad z^k = \theta_i^k - \theta_j^k,$$

$\theta^k = (\theta_1^k, \theta_2^k, \dots, \theta_n^k)$  — текущее приближение к точке минимума.

Функция  $\varphi^2(z) = \log [1 + \exp(-z + \varepsilon)]$  аппроксимируется аналогичной параболой, у которой

$$B^k = 0,25 \operatorname{sech}^2 [(-z^k + \varepsilon)/2], \quad (4.50)$$

$$C^k = z^k - (B^k [1 + \exp(-z^k + \varepsilon)])^{-1}.$$

Наконец, функция  $\varphi^3(z) = \log (\operatorname{ch} \varepsilon + \operatorname{ch} z)$  аппроксимируется параболой, у которой  $A^k$  по-прежнему не играет роли,

$$B^k = \frac{1 - \operatorname{ch} \varepsilon \operatorname{ch} z^k}{(\operatorname{ch} \varepsilon + \operatorname{ch} z^k)^2}, \quad C^k = z^k - \frac{\operatorname{ch} \varepsilon + \operatorname{ch} z^k}{B^k \operatorname{sh} z^k}.$$

Особенностью методики, включающей предположение о том, что  $f(z) = 0,25 \operatorname{sech}^2(z/2)$  и, следовательно,  $p_{ij} = (1 + \exp(\theta_j - \theta_i))^{-1}$ , является то, что сумма  $\sum m_{ij} p_{ij} = s_i$ , где  $s_i$  — сумма очков, набранная  $i$ -й командой в турнире. Таким образом, шкала относительной силы команд, построенной по методике Рао — Купера (при наличии ничьих) или по методике Бредли — Терри (при запрете на ничьи), не зависит от результатов отдельных встреч команд между собой, а только лишь от количества очков, набранных участниками, и количества игр  $m_{ij}$ , сыгранных между ними. Единственным условием ранжирования всех участников является сравнимость по силе любых двух команд, для чего необходимо, чтобы существовала хотя бы одна цепочка  $i, i_1, i_2, \dots, i_m, j$ , связывающая  $i$  и  $j$  отношением доминирования, подчинения или эквивалентности. Несравнимые объекты ни этим, ни каким-либо другим способом не ранжируются. При этом необязательно, чтобы две цепочки отношений, связывающие  $i$ -й и  $j$ -й объекты, были непротиворечивыми.

|   | a   | b   | c   | d   | e   | f   |
|---|-----|-----|-----|-----|-----|-----|
| a | X   | 7:3 |     |     | 2:1 | 3:3 |
| b | 3:7 | X   | 2:1 |     |     | 3:1 |
| c |     | 1:2 | X   | 0:5 |     |     |
| d |     |     | 5:0 | X   | 6:1 |     |
| e | 1:2 |     |     | 1:6 | X   | 1:1 |
| f | 3:3 | 1:3 |     |     | 1:1 | X   |

Рис. 4.9

**Пример 4.2.** Даны результаты незаконченного турнира без ничьих: первая цифра означает количество побед, вторая — количество поражений (рис. 4.9).

Ранжировка при экспоненциальном распределении такова:  $dabefc$ . Шкала имеет вид  $\theta_a = 0$ ,  $\theta_b = -1,00$ ,  $\theta_c = -2,277$ ,  $\theta_d = 0,572$ ,  $\theta_e = -1,213$ ,  $\theta_f = -1,416$ .

Аналогичная ранжировка при гиперсеканс-квадрат-распределении имеет вид  $dabfec$ , а соответствующая шкала выглядит так:  $\theta_a = 0$ ,  $\theta_b = -0,922$ ,  $\theta_c = -1,618$ ,  $\theta_d = 0,822$ ,  $\theta_e = -1,231$ ,  $\theta_f = -0,982$ .

#### 4.4. ОБРАБОТКА «НЕНАПРАВЛЕННЫХ» СООТНОШЕНИЙ

«Ненаправленными» соотношениями мы договорились называть соотношения вида

$$|\theta_i - \theta_j| = d_{ij}, \quad (4.51)$$

где  $\theta_i$  и  $\theta_j$  — ценности  $i$ -го и  $j$ -го объектов;  $d_{ij}$  — степень сходства объектов по оценке эксперта.

Данные типа (4.51) в настоящее время обрабатываются с помощью различных эвристических приемов, не дающих гарантии того, что получено точное решение. Методы, основанные на минимизации некоторой функции  $\Phi(\Delta_{11}, \Delta_{12}, \dots, \Delta_{nn})$  от невязок  $\Delta_{ij}$ , приводят лишь в некоторый местный минимум этой функции вследствие невыпуклости модуля невязок

$$\Delta_{ij} = ||\theta_i - \theta_j| - d_{ij}| \quad (4.52)$$

по обоим аргументам (рис. 4.10).

Но при отыскании даже этого локального минимума возникают большие трудности, присущие вообще градиентным методам решения многомерных задач с плохим рельефом (затруднения с выбором и корректировкой шага, выбором начального приближения, правилами остановки и т. д.).

В настоящем параграфе описывается метод решения избыточной системы уравнений вида

$$|\theta_i - \theta_j| + \Delta_{ij} = d_{ij}, \quad i, j = 1, 2, \dots, n, \quad (4.53)$$

основанный на методе ВВКМ и гарантирующий как в одномерном, так и в многомерном случае глобальный характер полученного решения.

Итак, пусть задана матрица  $D = ||d_{ij}||$ , характеризующая близость «ценностей» двух объектов  $A_i$  и  $A_j$ . Функция соответствия оценок ценности объектов  $A_i$  ( $i = 1, 2, \dots, n$ ) экспертным данным пусть равна сумме квадратов невязок:

$$\Phi = \sum_{i>j} ||\theta_i - \theta_j| - d_{ij}|^2. \quad (4.54)$$

Обозначим  $y_{ij} = \theta_i - \theta_j$  и  $\varphi_{ij} = ||y_{ij}| - d_{ij}|$ .

Каждая из функций  $\varphi_{ij}$  может быть мажорирована одной из двух функций:

$$\bar{\varphi}_{ij}^n = |y_{ij} - d_{ij}| \quad \text{или} \quad \bar{\varphi}_{ij}^n = |y_{ij} + d_{ij}| \quad (4.55)$$

и минорирована функцией

$$\underline{\varphi}_{ij} = \frac{|y_{ij} - d_{ij}| + |y_{ij} + d_{ij}| + (y_{ij} - d_{ij}) + (y_{ij} + d_{ij})}{2}. \quad (4.56)$$

(рис. 4.11).

Обе мажоранты могут быть описаны выражением одного вида

$$\varphi_{ij}^\alpha = |\alpha_{ij} y_{ij} + d_{ij}|, \quad (4.57)$$

где  $\alpha_{ij} = +1$  для левой мажоранты и  $\alpha_{ij} = -1$  для правой мажоранты.

Если обозначить

$$u(z) = \begin{cases} z, & \text{если } z \geq 0, \\ 0, & \text{если } z < 0, \end{cases} \quad (4.58)$$

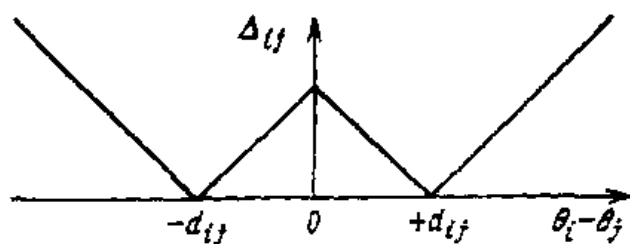


Рис. 4.10

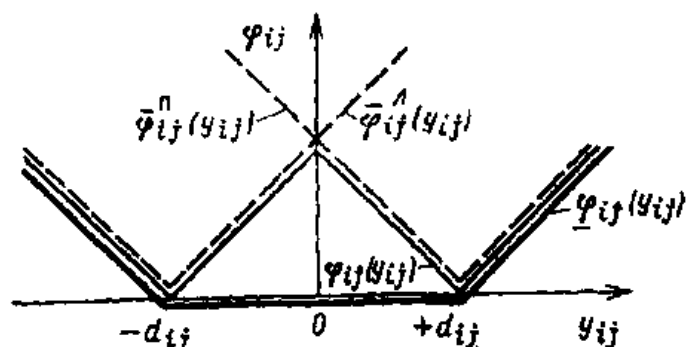


Рис. 4.11

то обе мажоранты и миноранты функции  $\varphi_{ij}(y_{ij})$  могут быть представлены следующим образом:

$$\bar{\varphi}_{ij}^n(y_{ij}) = u(y_{ij} - d_{ij}) + u(-y_{ij} + d_{ij}), \quad (4.59)$$

$$\bar{\varphi}_{ij}^h(y_{ij}) = u(y_{ij} + d_{ij}) + u(-y_{ij} - d_{ij}), \quad (4.60)$$

$$\underline{\varphi}_{ij}(y_{ij}) = u(y_{ij} - d_{ij}) + u(-y_{ij} - d_{ij}). \quad (4.61)$$

Мажорантами и минорантами выражений  $||y_{ij}| - d_{ij}|^2$ , входящих слагаемыми в (4.54), являются соответственно:

$$\begin{aligned} \bar{\psi}_{ij}^n &= v(y_{ij} - d_{ij}) + v(-y_{ij} + d_{ij}), \\ \bar{\psi}_{ij}^h &= v(y_{ij} + d_{ij}) + v(-y_{ij} - d_{ij}), \end{aligned} \quad (4.62)$$

$$\underline{\psi}_{ij} = v(y_{ij} - d_{ij}) + v(-y_{ij} - d_{ij}),$$

где

$$v(z) = \begin{cases} z^2, & \text{если } z \geq 0, \\ 0, & \text{если } z < 0. \end{cases}$$

Поскольку все выражения  $v(z)$ , где  $z = \sum a_j \theta_j + b$ , в свою очередь, мажорируются параболой  $\psi^k(z) = (z - C^k)^2$ , где

$$C^k = \begin{cases} 0, & \text{если } z^k \geq 0, \\ z^k, & \text{если } z^k < 0, \end{cases} \quad (4.63)$$

$$z^k = \sum_j a_j \theta_j^k + b,$$

касающимися мажорируемых функций в точке  $z^k$ , то для минимизации сумм произвольного числа описанных выше мажорант и минорант функции  $\varphi_{ij}(y_{ij})$  может быть эффективно использован метод ВВКМ.

Верхнюю оценку функции (4.54) можно получить следующим образом. Задаемся произвольным набором  $\theta_1^k, \theta_2^k, \dots, \theta_n^k$  (например,  $\theta_1^k = -\theta_2^k = \dots = \theta_n^k = 0$ ). Фиксируем значение  $\theta_1^k$  и в дальнейшем исключаем  $\theta_1^k$  из числа неизвестных. Подсчитываем значения  $y_{ij}^k = \theta_i^k - \theta_j^k$  и по-

лучаем очередное  $(k + 1)$ -е приближение путем минимизации по  $\theta$  суммы мажорант  $\psi(\theta) = \sum_s \psi_s^k(y_s)$ . При этом каждая мажоранта строится с помощью (4.63).

Для соблюдения условий

$$\varphi_{ij}(y_{ij}) = \psi_{ij}(y_{ij})|_{y_{ij}=y_{ij}^k}, \quad \varphi'_{ij}(y_{ij}) = \psi'_{ij}(y_{ij})|_{y_{ij}=y_{ij}^k}, \quad (4.64)$$

требуемых методом ВВКМ для мажорирования слагаемого  $[\varphi_{ij}(y_{ij})]^2$ , используем правую мажоранту  $\bar{\psi}_{ij}^n$ , если  $y_{ij}^k > 0$ , и левую мажоранту  $\bar{\psi}_{ij}^n$ , если  $y_{ij}^k < 0$ . Если же  $y_{ij}^k = 0$ , то можно избрать любую из мажорант. Во всех последующих расчетах для определенности берется правая мажоранта.

С учетом сделанных замечаний в качестве мажоранты изберем

$$\psi^k = \sum_{i < j} |\alpha_{ij}^k y_{ij} + d_{ij}|^2 = \sum_{i < j} v(\alpha_{ij}^k y_{ij} + d_{ij}) + \sum_{i < j} v(-\alpha_{ij}^k y_{ij} - d_{ij}), \quad (4.65)$$

где

$$\alpha_{ij}^k = \begin{cases} 1, & \text{если } y_{ij}^k \geq 0, \\ -1, & \text{если } y_{ij}^k < 0. \end{cases} \quad (4.66)$$

Слагаемые, входящие в (4.65), в свою очередь, мажорируются параболлами

$$\psi^k(z_{ij}) = (z_{ij} - C_{ij}^k)^2 \text{ и } \psi^k(w_{ij}) = (w_{ij} - D_{ij}^k)^2,$$

где

$$\begin{aligned} z_{ij} &= \alpha_{ij}^k y_{ij} + d_{ij}, \quad w_{ij} = -\alpha_{ij}^k y_{ij} - d_{ij}; \\ C_{ij}^k &= \begin{cases} \alpha_{ij}^k y_{ij} + d_{ij}, & \text{если } \alpha_{ij}^k y_{ij} + d_{ij} \geq 0, \\ 0 & \text{в противоположном случае;} \end{cases} \\ D_{ij}^k &= \begin{cases} -\alpha_{ij}^k y_{ij} - d_{ij}, & \text{если } -\alpha_{ij}^k y_{ij} - d_{ij} \geq 0, \\ 0 & \text{в противоположном случае.} \end{cases} \end{aligned}$$

Из общей теории ВВКМ следует, что этот процесс должен сходиться к точке локального минимума. При этом результат решения задачи зависит от того, какие значения приняли коэффициенты  $\alpha_{ij}^k$  в начальной точке. Пусть точкой абсолютного минимума является точка  $\theta_{\min}$ . Этой точке соответствует определенный набор  $\alpha_{ij}^k$ . Задавшись этим набором, получим точку  $\theta^{k+1}$ , в которой  $\Phi(\theta^{k+1}) \leq \Phi(\theta^k)$ . Поскольку  $\theta^k = \theta_{\min}$ , то и  $\theta^{k+1} = \theta_{\min}$ . Отсюда следует, что задача сводится к отысканию некоторого набора  $\alpha_{ij}^k$ , начав с которого детерминированную процедуру ВВКМ, мы обязательно перейдем в точку абсолютного минимума.

Теоретически этот набор можно найти полным перебором всех возможных комбинаций значений  $\alpha_{ij} = \{+1, -1\}$ ,  $(i, j = 1, 2, \dots, n)$ ,  $i < j$ . Ясно, что этот путь связан с непомерно большими затратами машинного времени даже при небольших  $n$ . Поэтому от него следует отказаться с самого начала и искать другие, более эффективные алгоритмы. Одним из таких алгоритмов является метод ветвей и границ.

Для применения этого метода нужно, во-первых, указать способ разбиения всего множества возможных комбинаций на подмножества (способ ветвления) и, во-вторых, научиться оценивать снизу значение оптимизируемой величины в пределах каждого подмножества (указание границы). В данном случае для указания способа ветвления поступим следующим образом: расположим все величины  $d_{ij}$  в порядке убывания их значений и занумеруем пары индексов  $(i, j)$ , соответствующие полученному ряду. В результате получим отображение пар индексов  $(i, j)$  на некоторый отрезок натурального ряда. Пусть теперь множество индексов выглядит так:

$$(i_1, j_1), (i_2, j_2), \dots, (i_h, j_h), \dots \quad (4.67)$$

На первом шаге ветвления множество всех комбинаций значений  $\alpha_{ij}$  разобьем на два подмножества: первое, у которого  $\alpha_{i_1, j_1} = +1$ , и второе, у которого  $\alpha_{i_1, j_1} = -1$ . Для указания нижней границы  $\Phi$  для каждого из подмножеств надо решить две задачи на абсолютный минимум суммы выпуклых вниз функций  $\Phi^n$  и  $\Phi^L$ :

$$\Phi^n = \bar{\varphi}_{i_1, j_1}^n(y_{i_1, j_1}) + \sum_{\substack{i < j \\ (i, j) \neq (i_1, j_1)}} \varphi_{ij}(y_{ij}), \quad (4.68)$$

$$\Phi^L = \bar{\varphi}_{i_1, j_1}^L(y_{i_1, j_1}) + \sum_{\substack{i < j \\ (i, j) \neq (i_1, j_1)}} \varphi_{ij}(y_{ij}) \quad (4.69)$$

Минимальные значения этих функций как раз и используются в качестве нижних оценок значения минимума  $\Phi$  в этих подмножествах. Берем среди этих двух нижних границ наименьшую. Соответствующее ей множество значений  $\alpha_{ij}$  ( $(i, j) \neq (i_1, j_1)$ ) разбиваем еще на два подмножества, характеризуемые  $\alpha_{i_2, j_2} = +1$  и  $\alpha_{i_2, j_2} = -1$ , с которыми поступаем аналогично.

Нижнюю границу для любой комбинации индексов, полученной в результате этих дихотомических операций, будем вычислять по формуле

$$\Phi_{A_1, A_2, A_3} = \min \left\{ \sum_{\substack{i < j \\ (i, j) \in A_1}} \bar{\varphi}_{ij}^n(y_{ij}) + \sum_{\substack{i < j \\ (i, j) \in A_2}} \bar{\varphi}_{ij}^L(y_{ij}) + \sum_{\substack{i < j \\ (i, j) \in A_3}} \varphi_{ij}(y_{ij}) \right\}, \quad (4.70)$$

где  $A_1$  — подмножество пар индексов, для которых  $\alpha_{ij} = +1$ ;  $A_2$  — подмножество пар индексов, для которых  $\alpha_{ij} = -1$ ;  $A_3$  — подмножество пар индексов с незафиксированными  $\alpha_{ij}$ .

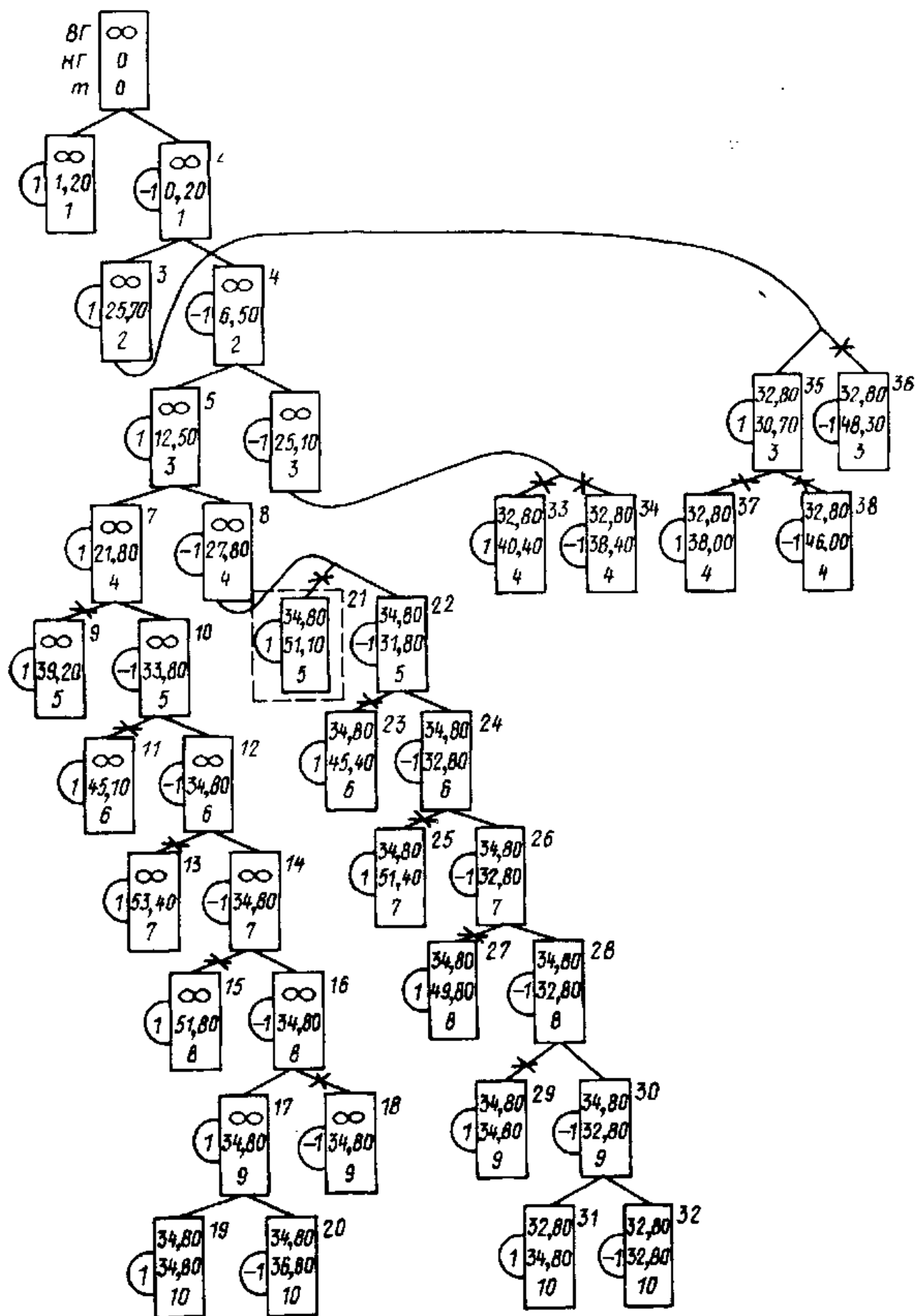


Рис. 4.13

Выбрав на каждом шаге одну из ветвей, мы в конце концов «доберемся» до комбинации, не содержащей незафиксированных индексов. Третья сумма в (4.70) будет отсутствовать. Значение минимума  $\Phi$  на этом подмножестве будет совпадать со значением  $\Phi$  в одной из точек локального минимума. Оно может быть принято в качестве верхней границы искомого абсолютного минимума функции  $\Phi$ . С ее помощью можно обрубить все ветви, в которых нижняя оценка минимума больше этой верхней грани, а одну из «ущевшихся» ветвей развивать до тех пор, пока нижняя грань не превысит верхнюю, или же, в противоположном случае, пока не будет получено новое значение местного минимума и соответственно новое более точное значение верхней границы. В последнем случае появляется возможность «обрубить» дополнительно еще ряд ветвей. В итоге должен остаться один вариант движения по бинарному дереву, приводящий в точку абсолютного минимума.

То, что на верхних ветвях конкретизируются  $\alpha_{ij}$ , соответствующие максимальным  $d_{ij}$ , повышает вероятность отыскания абсолютного минимума уже на первых шагах этого процесса. Процесс отыскания нужной ветви ускоряется еще больше, если одновременно с нижней оценкой строить верхнюю оценку каждого подмножества, для которого характерна фиксация некоторого  $\alpha_{ij}$ . Совпадение верхней и нижней оценок для любого из подмножеств означает, что найдена точная оценка минимума  $\Phi$  на этом подмножестве и производить дальнейшее развитие данной ветви дерева нет необходимости.

**Пример 4.3.** Задана матрица близости пяти объектов, показанная на рис. 4.12.

Ход решения задачи шкалирования представим в виде многошагового процесса, на каждом шаге которого рассчитывается оценка нижней границы и принимается решение о закреплении правой или левой мажоранты. Каждый шаг этого процесса опишем тройкой (ВГ, НГ, Ч), где ВГ — оценка верхней границы; НГ — оценка нижней границы; Ч — число слагаемых, в отношении которых принято решение — мажорировать их правой или левой мажорантой. Совокупность таких троек образует дерево, которое для рассматриваемого примера представлено на рис. 4.13. Информация, описывающая отдельные узлы этого дерева, соответствует указанным тройкам. Так, обозначение выделенного штриховыми линиями узла несет следующую информацию: уже закреплены пять мажорант для слагаемых вида  $\Phi_{ij} = ||y_{ij}| - d_{ij}|$  с наибольшими значениями  $d_{ij}$ , верхняя граница («рекорд», достигнутый к данному шагу) равна 34,80, нижняя граница равна 51,10. Единица в полукруге слева показывает, что данный узел соответствует фиксации правой мажоранты.

|      |      |      |      |      |
|------|------|------|------|------|
|      | 8,5  | 16,0 | 18,0 | 14,7 |
| 8,5  |      | 1,0  | 9,3  | 15,3 |
| 16,0 | 1,0  |      | 9,3  | 1,0  |
| 18,0 | 9,3  | 9,3  |      | 15,3 |
| 14,7 | 15,3 | 1,0  | 15,3 |      |

Рис. 4.12

При построении дерева решений мы придерживались единого правила: каждая левая ветвь соответствует фиксации левой мажоранты, а правая — правой.

Порядок обхода дерева решений в методе ветвей и границ определяется выбранной схемой ветвления. Мы не будем специально рассматривать этот вопрос, так как он подробно изложен в соответствующей литературе [18]. Здесь только отметим, что при решении данного примера использовалась схема ветвления, позволяющая как можно быстрее получить «хороший» рекорд. Суть этой схемы сводится к следующему: на каждом шаге для ветвления выбирается более перспективная из альтернатив, построенных на последнем шаге, пока не будет получен конкретный вариант шкалирования.

Числа, приписанные справа вверху у каждого узла, показывают очередность обхода дерева. Знаком  $\times$  обозначены ветви дерева, забракованные в процессе поиска, так как их оценки нижней границы целевой функции хуже, чем уже полученный «рекорд».

Для более наглядного и компактного изображения дерева решений на рис. 14.13 все ветви располагаются справа налево в порядке их прохождения. Так, самая левая ветвь соответствует первому прохождению дерева с целью получения верхней границы (узел 19; ВГ = 34, 80). До этого момента значение верхней границы положено равным  $+\infty$  (узлы 1...18). В дальнейшем, используя полученное значение верхней границы, удается исключить развитие ряда узлов дерева (18, 15, 13, 9). При повторном прохождении дерева из узла 8 получено значение верхней границы (узел 32; ВГ = 32,80) \*). Попытки улучшить это значение в узлах 6 и 3 оказались безуспешными. Узел 1 не рассматривался в силу симметричности задачи \*\*). В результате при минимальном значении целевой функции, равном 32,80, получена следующая шкала близости объектов:  $\theta_1 = 0$ ;  $\theta_2 = 10,53$ ;  $\theta_3 = 11,53$ ;  $\theta_4 = 20,68$ ;  $\theta_5 = 13,20$ .

Теперь приведем некоторые соображения по совместной обработке данных экспертного опроса, представленных в различной форме: в форме размера интервала, в котором может находиться значение неизвестной величины, в форме парных сравнений и в форме значения близости между некоторыми объектами. Если удастся размеры интервала и значение «близости» выразить через размеры вилки нечувствительности в парных сравнениях, то задача отыскания оценки  $\theta$  сводится к минимизации суммы:

$$\Phi(\theta) = \Phi_1(\theta) + \Phi_2(\theta) + \Phi_3(\theta), \quad (4.71)$$

где  $\Phi_1(\theta)$  — сумма невязок, возникающих при оценках в форме парных сравнений;  $\Phi_2(\theta)$  — сумма невязок, возникающих при оценках в виде указания степени близости;  $\Phi_3(\theta)$  — сумма невязок, порожденных оценками в виде указания интервалов.

Универсальным методом нахождения минимума (4.71) является метод ветвей и границ.

---

\*) Порядок обхода дерева здесь определяется также тем, что в программе, использовавшейся для решения задачи на ЭВМ, информация об активных узлах дерева хранится в стеке, поэтому узлы, записанные в стек первыми, просматриваются последними. Реализуется принцип — «пришедший первым обслуживается последним».

\*\*) Задача допускает два равноценных варианта решения с точностью до перемены знака у всех  $\alpha_{ij}$ .

#### 4.5. ПРИМЕР КОЛИЧЕСТВЕННОЙ ОЦЕНКИ КАЧЕСТВА ИЗДЕЛИЙ МЕТОДОМ ОДНОМЕРНОГО МЕТРИЧЕСКОГО ШКАЛИРОВАНИЯ

Решение задачи количественной оценки качества изделий разобьем на следующие основные этапы:

выбор показателей, характеризующих изделия данного вида, и подготовка исходных данных в виде таблицы «объект — признак»;

нормирование данных таблицы «объект — признак»;

расчет набора матриц близости при использовании различных метрик расстояния;

решение задачи одномерного метрического шкалирования для каждой из матриц близости;

выбор наилучшей шкалы с учетом дополнительной информации.

Предлагаемая методика позволяет формализовать процедуру оценки качества изделий, сведя до минимума информацию, получаемую от экспертов, и использовать современную вычислительную технику.

Будем считать, что задан ряд изделий близкого функционального назначения и соответствующие их характеристики сведены в таблицу «объект — признак». В качестве примера в табл. 4.1 приведем данные о нескольких микроЭВМ (примерно одинаковой производительности и одного объема памяти) [4].

Как правило, признаки, отобранные экспертами для описания изделий, имеют различную размерность, а поэтому и различную единицу масштаба. Чтобы сделать возможным сопоставление признаков и устранить влияние размерности, матрицу исходных данных обычно трансформируют (нормируют), вводя единый для всех признаков масштаб. Самый распространенный вид такой нормировки таблицы «объект — признак» — приведение к стандартной форме, т. е. переход от

значений  $z_{ij}$  к значению  $x_{ij} = \frac{(z_{ij} - \bar{z}_j)}{s_j}$ ,  $i = 1, 2, \dots, N$ ,  $j = 1, 2, \dots, M$ , где  $z_{ij}$  — значение  $j$ -го признака для  $i$ -го объекта;  $\bar{z}_j$  — среднее арифметическое значение  $j$ -го признака;  $s_j$  — среднеквадратическое отклонение  $j$ -го признака.

Такая форма преобразования часто применяется в задачах факторного анализа. В табл. 4.2 исходные данные приведены к стандартной форме.

Для определения близости между оцениваемыми изделиями последовательно используем ряд известных метрик расстояния и получаем набор матриц близости  $\|d_{ij}^v\|$ ,  $i, j = 1, 2, \dots, N$ ,  $v = 1, 2, \dots, R$ , где  $R$  — число различных метрик, а  $N$  — число оцениваемых объектов.

Таблица 4.1

| Номер изделия | Масса, кг | Объем, дм <sup>3</sup> | Наработка между отказами, ч | Потребляемая мощность, Вт |
|---------------|-----------|------------------------|-----------------------------|---------------------------|
| 1             | 4,5       | 9,2                    | 1700                        | 170                       |
| 2             | 3,8       | 8,1                    | 1600                        | 150                       |
| 3             | 2,2       | 4,6                    | 1800                        | 120                       |
| 4             | 13,6      | 19,3                   | 1200                        | 195                       |
| 5             | 7,2       | 13,5                   | 1400                        | 180                       |
| 6             | 12,1      | 18,6                   | 1100                        | 210                       |
| 7             | 11,6      | 15,8                   | 1200                        | 240                       |
| 8             | 40        | 75                     | 400                         | 900                       |

Таблица 4.2

| Номер изделия | Масса  | Объем  | Наработка между отказами | Потребляемая мощность |
|---------------|--------|--------|--------------------------|-----------------------|
| 1             | -0,650 | -0,533 | 0,963                    | -0,419                |
| 2             | -0,712 | -0,585 | 0,722                    | -0,502                |
| 3             | -0,853 | -0,751 | 1,204                    | -0,627                |
| 4             | 0,152  | -0,056 | -0,241                   | -0,315                |
| 5             | -0,412 | -0,330 | 0,241                    | -0,377                |
| 6             | 0,020  | -0,089 | -0,482                   | -0,252                |
| 7             | -0,024 | -0,231 | -0,241                   | -0,127                |
| 8             | 2,480  | 2,576  | -2,167                   | 2,619                 |

На практике чаще всего используются:

- 1) евклидово расстояние  $d_{ij} = d(x_i, x_j) = \left[ \sum_{k=1}^M (x_{ik} - x_{jk})^2 \right]^{1/2}$ ;
  - 2) метрика Минковского  $d_{ij} = d(x_i, x_j) = \sum_{k=1}^M |x_{ik} - x_{jk}|$ ;
  - 3) супремум-норма  $d_{ij} = d(x_i, x_j) = \sup_{k=1,2,\dots,M} \{|x_{ik} - x_{jk}|\}^*$ ;
  - 4) расстояние Махаланобиса  $d_{ij}^2 = d^2(x_i, x_j) = (x_i - x_j)^T W^{-1} (x_i - x_j)$ ,
- где  $x_i$  и  $x_j$  — вектора характеристик  $i$ -го и  $j$ -го изделий,  $W^{-1}$  — весовая матрица.

Можно использовать и другие меры близости, не являющиеся расстоянием с точки зрения классического определения. Это мера Джеффриса — Матуситы

$$d_{ij} = d(x_i, x_j) = \left[ \sum_{k=1}^M (V_{x_{ik}} - V_{x_{jk}})^2 \right]^{1/2}$$

и мера, известная под названием «коэффициент дивергенции»,

$$d_{ij} = d(x_i, x_j) = \left\{ \frac{1}{M} \sum_{k=1}^M \left( \frac{x_{ik} - x_{jk}}{x_{ik} + x_{jk}} \right)^2 \right\}^{1/2}.$$

Кроме того, в качестве меры сходства можно применять обычный коэффициент корреляции. Расстояние Махаланобиса порождает целую совокупность матриц близостей при использовании различных весовых коэффициентов, отражающих, например, мнения отдельных экспертов.

Обозначим  $\theta_i, i = 1, 2, \dots, N$ , количественную характеристику качества  $i$ -го изделия. С учетом этого задачу одномерного метрического шкалирования сформулируем как задачу минимизации функционала

$$\theta = \operatorname{argmin}_{\theta} \sum_{i < j} |\theta_i - \theta_j - d_{ij}|. \quad (4.72)$$

Различным определениям расстояния  $d(x_i, x_j)$  будут соответствовать различные шкалы. Из этих шкал необходимо выбрать ту, которая в большей степени соответствует дополнительной информации.

Источником дополнительной информации могут служить результаты сравнительных испытаний изделий, результаты моделирования, результаты обработки экспертного опроса или просто результаты ретроспективного анализа. Например, информацию о том, что телевизор «Рубин Ц230» имеет лучшее качество, чем «Рубин 714», а, в свою очередь, «Рубин 714» предпочтительнее своих предшественников, следует уже из того, что эти модели сменяли друг друга в сферах производства и потребления. Примеров такого несложного ретроспективного анализа можно привести много в различных областях техники.

Дополнительная информация о качестве изделий может быть представлена в порядковых шкалах (бинарные отношения предпочтительности, ранжирования), в шкале отношений, в других удобных видах. При этом совсем не обязательно, чтобы привлекалась дополнительная информация о всех шкалируемых изделиях напротив, для повышения надежности этой информации можно использовать только ту ее часть, которая не вызывает сомнений. Например, если осуществляет-

\*) Указанные разновидности являются частными случаями так называемой  $l_p$ -нормы  $d_{ij} = \left[ \sum_k |x_{ik} - x_{jk}|^p \right]^{1/p}$ ; при  $p = 2$   $l_p$ -норма приобретает характер евклидова расстояния; при  $p = 1$  — расстояния Минковского, а при  $p = \infty$  — расстояния Чебышева или супремум-нормы.

ся экспертный опрос для выявления предпочтительности изделий, то для получения дополнительной информации в выборку следует включить только те изделия, сравнение которых не вызывает затруднений или вызывает минимальные затруднения.

Лучшей шкалой из полученного набора будем считать ту, которая обеспечивает наибольшее соответствие дополнительной информации.

Учитывая, что и полученная шкала, и дополнительная информация представляют собой бинарные отношения, для установления соответствия между ними воспользуемся мерами близости на отношениях, предложенными в [16].

|                                                    |                                                    |                                                    |                                                    |                                                    |
|----------------------------------------------------|----------------------------------------------------|----------------------------------------------------|----------------------------------------------------|----------------------------------------------------|
| $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ | 0                                                  | 1                                                  | $\frac{1}{2}$                                      | 0                                                  |
| 1                                                  | $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ | 1                                                  | 1                                                  | 1                                                  |
| 0                                                  | 0                                                  | $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ |                                                    | 0                                                  |
| $\frac{1}{2}$                                      | 0                                                  |                                                    | $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ | 0                                                  |
| 1                                                  | 0                                                  | 1                                                  | 1                                                  | $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ |

|                                                    |                                                    |                                                    |                                                    |                                                    |
|----------------------------------------------------|----------------------------------------------------|----------------------------------------------------|----------------------------------------------------|----------------------------------------------------|
| $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ | 0                                                  | 0                                                  | 1                                                  | $\frac{1}{2}$                                      |
| 1                                                  | $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ | $\frac{1}{2}$                                      | 1                                                  | 1                                                  |
| 1                                                  | $\frac{1}{2}$                                      | $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ | 1                                                  | 1                                                  |
| 0                                                  | 0                                                  | 0                                                  | $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ | 0                                                  |
| $\frac{1}{2}$                                      | 0                                                  | 0                                                  | 1                                                  | $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ |

|                                                    |                                                    |                                                    |                                                    |                                                    |
|----------------------------------------------------|----------------------------------------------------|----------------------------------------------------|----------------------------------------------------|----------------------------------------------------|
| $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ |                                                    |                                                    |                                                    |                                                    |
|                                                    | $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ | 1                                                  |                                                    | 1                                                  |
|                                                    | 0                                                  | $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ |                                                    | $\frac{1}{2}$                                      |
|                                                    |                                                    |                                                    | $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ |                                                    |
|                                                    | 0                                                  | $\frac{1}{2}$                                      |                                                    | $\begin{matrix} \diagup \\ \diagdown \end{matrix}$ |

Рис. 4.14

Мера близости между двумя произвольными результатами ранжирования  $P_1$  и  $P_2$  в соответствии с [16] определяется по формуле

$$d(P_1, P_2) = \frac{1}{2} \sum_{i,j=1}^N |p_{ij}^1 - p_{ij}^2| = \sum_{i < j} |p_{ij}^1 - p_{ij}^2|, \quad (4.73)$$

где  $p_{ij}^1$  и  $p_{ij}^2$  — элементы матриц предпочтений, соответствующих ранжированиям  $P_1$  и  $P_2$ .

Приведем пример выбора лучшей шкалы по принципу ее соответствия дополнительной информации. Пусть даны результаты ранжирования  $P_{ш1}$  и  $P_{ш2}$ , соответствующие двум шкалам, и ранжирование  $P_d$ , полученное от экспертов (дополнительная информация):

$$P_{ш1} = \begin{pmatrix} a_2 \\ a_5 \\ a_1 \approx a_4 \\ a_3 \end{pmatrix}, \quad P_{ш2} = \begin{pmatrix} a_2 \approx a_3 \\ a_1 \approx a_5 \\ a_4 \end{pmatrix}, \quad P_d = \begin{pmatrix} a_2 \\ a_5 \approx a_3 \end{pmatrix}.$$

Заменяем эти результаты соответствующими матрицами предпочтений (рис. 4.14).

В результате несложного расчета получаем, что  $d(P_{ш1}, P_d) = 1/2$ ,  $d(P_{ш2}, P_d) = 1$ . Следовательно, шкала 1 более приемлема, так как в большей степени согласуется с дополнительной информацией. Необходимо отметить, что если в матрице (и соответствующей ранжировке), задающей дополнительную информацию, часть отношений между объектами не определена и имеются пустые клетки, то при расчете  $d$  по (4.73) необходимо учитывать только элементы, которые имеются в матрице предпочтений, соответствующей дополнительной информации.

Если дополнительная информация носит метризованный характер, то можно воспользоваться мерами близости, предложенными в [16]. В остальном процедура оценки качества изделия остается прежней.

Завершим рассмотрение примера оценки качества микроЭВМ, основные данные которых приведены в табл. 4.1. Матрица близости, рассчитанная на основе евклидовой метрики, имеет вид

$$\|d_{ij}\| = \begin{vmatrix} 0 & 0,267 & 0,436 & 1,527 & 0,788 & 1,662 & 1,420 & 6,204 \\ & 0 & 0,543 & 1,410 & 0,634 & 1,514 & 1,291 & 6,186 \\ & & 0 & 1,918 & 1,167 & 2,045 & 1,815 & 6,639 \\ & & & 0 & 0,793 & 0,284 & 0,311 & 4,966 \\ & & & & 0 & 0,884 & 0,674 & 5,620 \\ & & & & & 0 & 0,309 & 4,923 \\ & & & & & & 0 & 5,040 \\ & & & & & & & 0 \end{vmatrix}$$

Матрица близости симметрична, поэтому приведена только ее наддиагональная часть. После решения задачи шкалирования (4.73) получаем  $\theta_1 = 0$ ;  $\theta_2 = 0,141$ ;  $\theta_3 = -0,396$ ;  $\theta_4 = 1,522$ ;  $\theta_5 = 0,771$ ;  $\theta_6 = 1,655$ ;  $\theta_7 = 1,420$ ;  $\theta_8 = 6,391$ . Таким образом, каждой из рассматриваемых ЭВМ поставлена в соответствие количественная оценка ее качества. Полученная в результате шкала является шкалой разности и оказывается наилучшей с точки зрения соответствия дополнительной информации. В заключение отметим, что в качестве дополнительной информации использовались результаты решения этой же задачи экспертным путем, приведенные в [4]. Полученная шкала имеет следующую направленность: меньшему показателю  $\theta_i$  соответствует лучшее качество изделия.

## ЗАКЛЮЧЕНИЕ

В основе получения оценок по данным измерений, так же как и в основе задачи шкалирования, лежит соответствующим образом сформулированная задача оптимизации. Происхождение функции, подлежащей оптимизации, носит двоякий характер: это либо логарифм апостериорной вероятности данной совокупности наблюдений, либо штраф за несоответствие результатов наблюдений предположениям о значениях неизвестных величин. Эти две постановки связаны между собой и иногда могут быть выведены друг из друга.

При некоторых естественных предположениях относительно функции распределения ошибок измерений и ошибок экспертизы оптимизируемая функция носит негладкий характер. Аналогичный негладкий характер могут носить и функции штрафа, назначаемые на основе «здравого смысла». В этом случае удобным средством отыскания минимума и, следовательно, средством построения оценок является метод ВВКМ. Иногда метод ВВКМ целесообразно применять и для минимизации суммы гладких функций, так как мажорирование, в отличие от аппроксимации, делает задачу получения оценок нечувствительной к выбору начального приближения.

Во всех случаях метод ВВКМ расширяет возможности обработчика при построении оценок и подходящих шкал. Особенно наглядно это проявляется при получении помехоустойчивых оценок и построении шкал по совокупности «ненаправленных соотношений».

## МАЖОРАНТЫ В ЗАДАЧАХ УСЛОВНОЙ ОПТИМИЗАЦИИ

До сих пор мы рассматривали задачи, в которых на искомые переменные ограничения не налагались. Следует, однако, отметить, что большинство практических задач проектирования и технико-экономического обоснования принимаемых решений содержат ограничения. В данной главе показано, что существует по крайней мере два варианта использования ВВКМ для решения задач с ограничениями.

Во-первых, можно развить идею мажорирования дальше и ввести понятие «мажорирование ограничений», которое предполагает замену исходной системы ограничений другой, более строгой. Одновременное мажорирование целевой функции и ограничений позволяет строить достаточно эффективные итерационные алгоритмы минимизации при наличии ограничений.

Во-вторых, практически любую задачу минимизации выпуклой функции с плохими аналитическими свойствами при наличии ограничений можно преобразовать с помощью метода штрафов в последовательность задач безусловной оптимизации. Для этих целей удобно воспользоваться методом ВВКМ, так как он позволяет вводить недифференцируемые штрафные функции вида  $\psi = \max \{0, \sum_i a_i x_i + b\}$ . На эффективность негладких штрафных функций указывали многие авторы [2, 6, 18, 25], отмечая, что их использование сдерживается отсутствием достаточно эффективных универсальных методов недифференцируемой оптимизации. Метод ВВКМ не относится к числу универсальных методов, однако в ряде случаев при решении задач с выпуклой целевой функцией, в том числе и с негладкой, он дает неплохие результаты.

### 5.1. МАЖОРИРОВАНИЕ ПРЯМОУГОЛЬНЫХ ОГРАНИЧЕНИЙ

Пусть имеются две системы ограничений  $R^k = \{g_i(x) \leq 0, i = 1, 2, \dots, n\}$  и  $R = \{l_i(x) \leq 0, i = 1, 2, \dots, n\}$ . Будем говорить, что система  $R^k$  мажорирует систему ограничений  $R$ , если все решения системы  $R^k$  являются решениями и для системы  $R$ . По самому определению область  $R^k$  является подмножеством области  $R$ . Примеры построения мажорирующих областей для системы ограничений

$$R = \begin{cases} \alpha_1 \leq x_1 \leq \beta_1, \\ \alpha_2 \leq x_2 \leq \beta_2 \end{cases}$$

представлены на рис. 5.1. Для односторонних ограничений одну из границ следует считать бесконечной (рис. 5.1, г). В дальнейшем в качестве мажорант будут использоваться различные выпуклые подмножества  $R^k$  множества  $R$ , причем наиболее удобными с точки зрения

практических применений оказываются эллипсоиды. Как будет показано ниже, применение мажорирующих эллипсоидов позволяет решать исходную задачу условной минимизации с помощью многократного использования метода наименьших квадратов без ограничений [12].

Для работы соответствующего алгоритма нужна произвольная точка  $x^0$ , удовлетворяющая наложенным ограничениям.

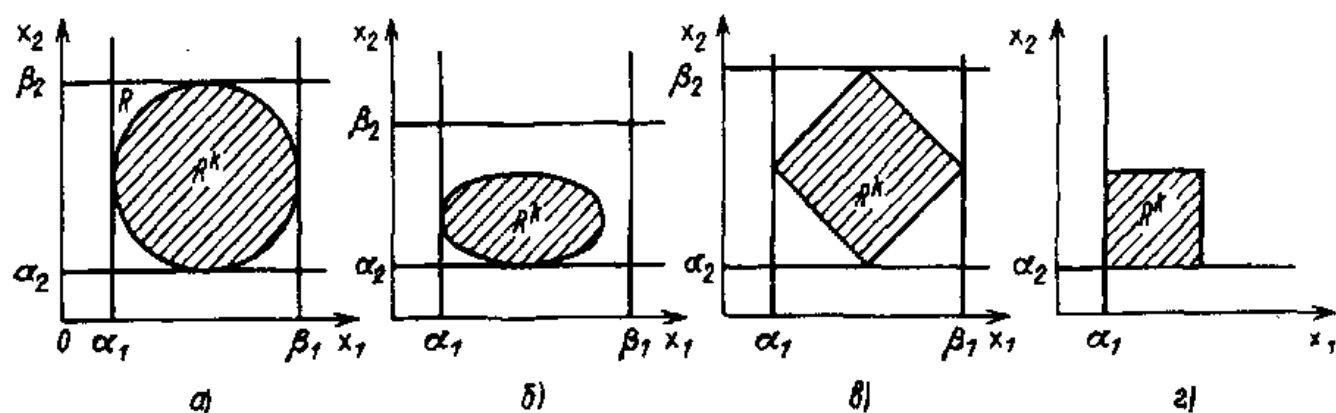


Рис. 5.1

Следует отметить, что получение допустимого решения в общем случае является достаточно сложной самостоятельной задачей, для которой можно эффективно использовать метод ВВКМ. Например, если ограничения имеют вид

$$\sum_{j=1}^n a_{ij}x_j + b_i \leq 0, \quad i = 1, 2, \dots, m,$$

то для отыскания точки, лежащей внутри области ограничений, нужно минимизировать сумму выпуклых вниз функций  $\sum_{i=1}^m \varphi_i(y_i)$ , где

$$\varphi_i(y_i) = \begin{cases} y_i - \sum_{j=1}^n a_{ij}x_j + b_i, & \text{если } y_i \geq 0, \\ 0 & , \text{если } y_i < 0. \end{cases}$$

Такие задачи неоднократно рассматривались нами ранее. В приложении приводится текст программы на языке ПЛ/1, предназначенной для нахождения точки, удовлетворяющей системе линейных ограничений в виде неравенств (программа POISK).

Имея  $x^0$ , построим:

а) параболоид  $\psi(x, x^0)$ , мажорирующий функцию  $f(x)$  и удовлетворяющий условиям

$$\psi(x, x^0) = f(x) \Big|_{x=x^0},$$

$$\nabla_x \psi = \nabla_x f \Big|_{x=x^0};$$

б) эллипсоид с центром в точке  $x^0$ , мажорирующий исходную систему ограничений.

Первой операции посвящено, в сущности, все предшествующее изложение, ее мы неоднократно описывали для самых разнообразных функций  $f(x)$  (см., например, гл. 2). Поэтому теперь основное внимание сосредоточим на построении мажоранты ограничений.

Начнем с прямоугольных ограничений (рис. 5.2). Здесь для построения мажорирующего эллипса сначала необходимо рассчитать минимальные расстояния от точки  $x^k$  до каждого из ограничений. Величины

$$\sigma_i = \max \{ \min (x_i^k - \alpha_i, \beta_i - x_i^k), \varepsilon \},$$

$$i = 1, 2, \dots, n$$

и являются ими с той, однако, оговоркой, что, если расстояние окажется слишком малым, необходима замена  $\sigma_i$  заранее заданным малым положительным числом  $\varepsilon$ , придавая тем самым процедуре мажорирования  $\varepsilon$ -приближенный характер.

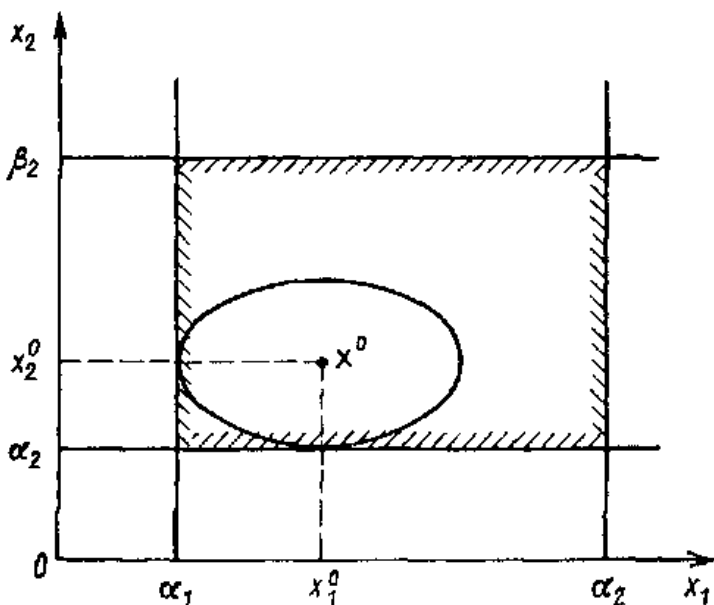


Рис. 5.2

При построении «мажорирующего» эллипсоида эти расстояния определяют размеры соответствующих полуосей. Запишем уравнение мажорирующего эллипсоида

$$R^k = \left\{ x: \frac{(x_1 - x_1^0)^2}{\sigma_1^2} + \frac{(x_2 - x_2^0)^2}{\sigma_2^2} + \dots + \frac{(x_n - x_n^0)^2}{\sigma_n^2} \leq 1 \right\} \quad (5.1)$$

или в более компактной матричной записи

$$R^k = \{ x: F(x, x^0) = (x - x^0)^T Q (x - x^0) \leq 1 \}, \quad (5.2)$$

где

$$Q = \begin{bmatrix} 1/\sigma_1^2 & 0 & \dots & 0 \\ 0 & 1/\sigma_2^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 1/\sigma_n^2 \end{bmatrix}.$$

По построению диагональные элементы матрицы  $Q$  ограничены снизу величиной

$$\max \{ [(x_i^0 - \alpha_i)/2]^{-2}, [(x_i^0 - \beta_i)/2]^{-2} \},$$

а сверху величиной  $1/\varepsilon^2$ .

После построения эллипсоида мы можем заменить систему ограничений  $R$  одним ограничением  $R^k$ . На первый взгляд, новое ограничение в данной задаче имеет более сложный вид и такая замена не сулит особых выгод (при более сложных системах ограничений упрощение, получаемое в результате мажорирования ограничений, более очевидно). Однако переход к одному, хотя и более сложному ограничению, упрощает контроль соблюдения ограничений в ходе поиска минимума целевой функции  $f(x)$  и позволяет организовать описанную ниже итерационную процедуру.

Имея точку  $x^k$  ( $k=0,1,\dots$ ), найдем точку  $y^k = \underset{x}{\operatorname{argmin}} \psi(x, x^k)$ , не заботясь о соблюдении ограничений.

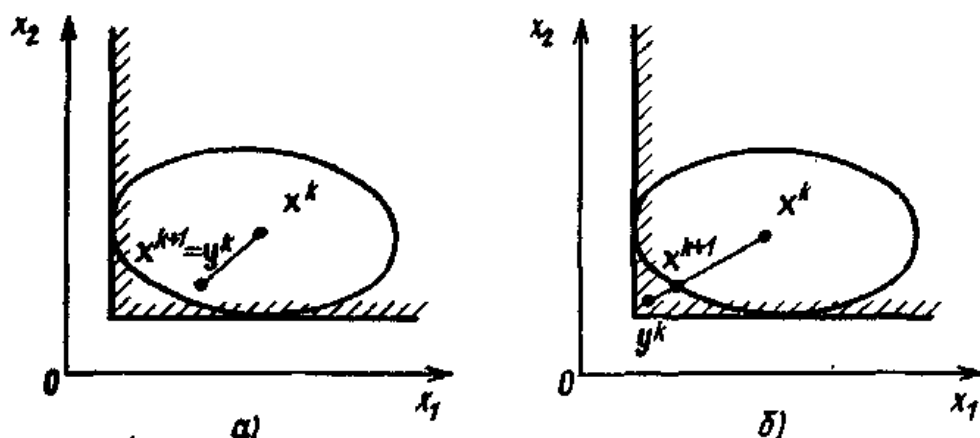


Рис. 5.3

Если точка  $y^k$  лежит внутри эллипсоида  $(x - x^k)^T Q (x - x^k) = 1$  (признаком этого является соблюдение неравенства (5.2)), то в качестве следующего приближения к точке минимума принимаем точку  $x^{k+1} = y^k$  (рис. 5.3, а). Если точка  $y^k$  лежит за пределами эллипсоида  $F(x, x^k) = 1$ , то в качестве следующего приближения надо взять точку, лежащую на границе эллипсоида (рис. 5.3, б). Такой выбор очередного приближения избавляет от необходимости проверять соблюдение системы ограничений исходной задачи, так как, обеспечивая выполнение мажорирующего ограничения, мы никогда не нарушаем исходных ограничений.

Для нахождения такой точки на границе эллипсоида необходимо решить следующую задачу математического программирования: найти

$$\min \psi,$$

где

$$\psi = (Bx - C^k)^T P^k (Bx - C^k) \quad (5.3)$$

при ограничении  $F(x, x^k) = 1$ , где

$$F(x, x^k) = (x - x^k)^T Q^k (x - x^k).$$

Воспользуемся методом множителей Лагранжа, для чего составим функцию Лагранжа  $L = \psi + \lambda F$  и запишем соответствующие уравнения относительно  $x$  и  $\lambda$ :

$$\nabla \psi = -\lambda \nabla F, \quad F = 1$$

или более подробно

$$2B^T P^k (Bx - C^k) = -2\lambda Q^k (x - x^k).$$

Отсюда после несложных преобразований получаем

$$x = (B^T P^k B + \lambda Q^k)^{-1} (B^T P^k C^k + \lambda Q^k x^k) \quad (5.4)$$

и далее

$$\begin{aligned} x - x^k &= (B^T P^k B + \lambda Q^k)^{-1} (B^T P^k C^k + \lambda Q^k x^k) - \\ &- (B^T P^k B + \lambda Q^k)^{-1} (B^T P^k B + \lambda Q^k) x = \\ &= (B^T P^k B + \lambda Q^k)^{-1} (B^T P^k C^k - B^T P^k B x^k) = \\ &= (B^T P^k B + \lambda Q^k)^{-1} B^T P^k (C^k - B x^k). \end{aligned} \quad (5.5)$$

Если в (5.5) положить  $\lambda = 0$ , то

$$x - x^k = (B^T P^k B)^{-1} B^T P^k C^k - x^k, \quad x = \arg \min_x \psi(x, x^k) = y^k.$$

Если в этом выражении положить  $\lambda \rightarrow \infty$ , то правая часть будет стремиться к нулю, а  $x$  к  $x^k$ . При других положительных значениях  $\lambda$  искомая точка минимума будет располагаться на линии «водостока» поверхности  $\psi(x, x^k)$ , проходящей через точку  $x = x^k$ . Для определения конкретного значения  $\lambda$  необходимо подставить (5.5) в формулу  $F(x, x^k) = 1$ . Получим

$$\begin{aligned} (Bx^k - C^k)^T P^k B (B^T P^k B + \lambda Q^k)^{-1} Q^k (B^T P^k B + \lambda Q^k) \times \\ \times B^T P^k (Bx^k - C^k) = 1. \end{aligned} \quad (5.6)$$

К сожалению, это уравнение степени  $2n$  относительно  $\lambda$  не удастся разрешить в конечном виде. Поэтому постараемся найти его приближенное решение.

Пусть на каком-то этапе поиска точка  $x^{k+1}$  стала близка к точке  $x^k$ . Это означает, что точка минимума (5.4) стала близка к точке  $x^k$ , а это возможно тогда, когда  $\lambda \rightarrow \infty$ . Собственные значения матрицы  $P^k$  ограничены сверху некоторой положительной константой, собственные значения диагональной матрицы  $Q^k$  ограничены снизу. Поэтому при приближении к границе области  $R$  имеет место  $x^{k+1} \rightarrow x^k$ ,  $\lambda \rightarrow \infty$ ,  $\lambda Q^k \rightarrow \infty$ ,  $B^T P^k B \rightarrow \text{const}$ , а  $B^T P^k B$  становится значительно меньше матрицы  $\lambda Q^k$ . В силу этого уравнение (5.6) принимает вид

$$(Bx^k - C^k)^T P^k B \lambda^{-1} (Q^k)^{-1} Q^k \lambda^{-1} (Q^k)^{-1} B^T P^k (Bx^k - C^k) = 1$$

или

$$\lambda^2 = (Bx^k - C^k)^T P^k B (Q^k)^{-1} B^T P^k (Bx^k - C^k). \quad (5.7)$$

Таким образом, теперь имеются два основных соотношения (5.4) и (5.7), которые позволяют найти решение задачи (5.3). Напомним, что решением является точка  $x$  на поверхности эллипсоида, которая принимается в качестве очередного приближения к точке минимума. Определяя  $\lambda$  приближенно по (5.7), мы каждый раз будем получать его завышенное значение. Это приводит к тому, что и при расчете координат точки очередного приближения по (5.4) мы также ошибаемся, но ошибаемся так, что искомая точка  $x$  оказывается не точно на границе эллипсоида, а в окрестности этой границы и, что особенно важно, всегда внутри эллипсоида. Из-за этого несколько уменьшается размер шага в направлении минимума и, следовательно, снижается скорость сходимости, но зато получаются дополнительные гарантии того, что, например, из-за ошибок округления мы не выйдем в процессе итераций за пределы мажорирующего ограничения. В дальнейшем описанный алгоритм будем называть алгоритмом комплексного мажорирования (так как одновременно мажорируются и целевая функция, и ограничения).

Работоспособность алгоритма подтвердим решением числового примера.

**Пример 5.1.** Необходимо определить точку  $x (x_1, x_2)$  на плоскости на минимальном суммарном расстоянии от трех заданных точек с координатами  $A_1(a_{11}, a_{12}) = (-1, 1)$ ,  $A_2(a_{21}, a_{22}) = (-1, -0.5)$  и  $A_3(a_{31}, a_{32}) = (1, -1)$ . Такие задачи мы уже рассматривали ранее (см. гл. 3). Это задача Штейнера. Раньше мы не накладывали никаких ограничений на область допустимых значений  $x$ . Теперь же ограничим область возможного размещения искомой точки неравенствами  $1 \leq x_1 \leq 20$ ,  $x_2 \geq 1$ . Целевая функция задачи

$$f(x) = \sqrt{(x_1 + 1)^2 + (x_2 - 1)^2} + \sqrt{(x_1 + 1)^2 + (x_2 + 0.5)^2} + \sqrt{(x_1 - 1)^2 + (x_2 + 1)^2}.$$

В качестве начальной точки возьмем  $x^0 = (x_1^0, x_2^0) = (15, 10)$ .

Воспользуемся ранее полученными результатами и запишем выражение для мажорирующего параболоида в точке  $x^0$ :

$$\begin{aligned} \psi(x, x^0) = & \frac{(x_1 + 1)^2 + (x_2 - 1)^2}{\sqrt{(x_1^0 + 1)^2 + (x_2^0 - 1)^2}} + \\ & + \frac{(x_1 + 1)^2 + (x_2 + 0.5)^2}{\sqrt{(x_1^0 + 1)^2 + (x_2^0 + 0.5)^2}} + \frac{(x_1 - 1)^2 + (x_2 + 1)^2}{\sqrt{(x_1^0 - 1)^2 + (x_2^0 + 1)^2}} \end{aligned}$$

или

$$\psi(x, x^0) = (B^0 x - C^0)^T P^0 (B^0 x - C^0),$$

где

$$B^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C^0 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ -0.5 \\ 1 \\ -1 \end{bmatrix}.$$

$$p_0 = \begin{bmatrix} 1/\sqrt{337} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/\sqrt{337} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{366} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/\sqrt{366} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/\sqrt{317} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/\sqrt{317} \end{bmatrix}$$

Т а б л и ц а 5.1

| Номер шага | $x_1^k$ | $x_2^k$ | $f(x^k)$ | Номер шага | $x_1^k$ | $x_2^k$ | $f(x^k)$ |
|------------|---------|---------|----------|------------|---------|---------|----------|
| 0          | 15,000  | 10,000  | 55,300   | 7          | 1,128   | 1,143   | 6,967    |
| 1          | 12,348  | 5,888   | 42,287   | 8          | 1,043   | 1,049   | 6,657    |
| 2          | 7,714   | 4,728   | 28,466   | 9          | 1,013   | 1,015   | 6,549    |
| 3          | 4,172   | 3,763   | 18,289   | 10         | 1,004   | 1,005   | 6,514    |
| 4          | 2,561   | 2,587   | 12,525   | 11         | 1,001   | 1,001   | 6,504    |
| 5          | 1,757   | 1,800   | 9,362    | 12         | 1,000   | 1,000   | 6,501    |
| 6          | 1,333   | 1,362   | 7,731    | 13         | 1,000   | 1,000   | 6,500    |

Уравнение мажорирующего ограничения эллипса с центром в точке (15,10) запишется так:

$$\begin{pmatrix} x_1 - 15 \\ x_2 - 10 \end{pmatrix}^T \begin{bmatrix} 1.25 & 0 \\ 0 & 1.100 \end{bmatrix} \begin{pmatrix} x_1 - 15 \\ x_2 - 10 \end{pmatrix} = 1,$$

поскольку  $\sigma_1 = 5$ ,  $\sigma_2 = 10$ .

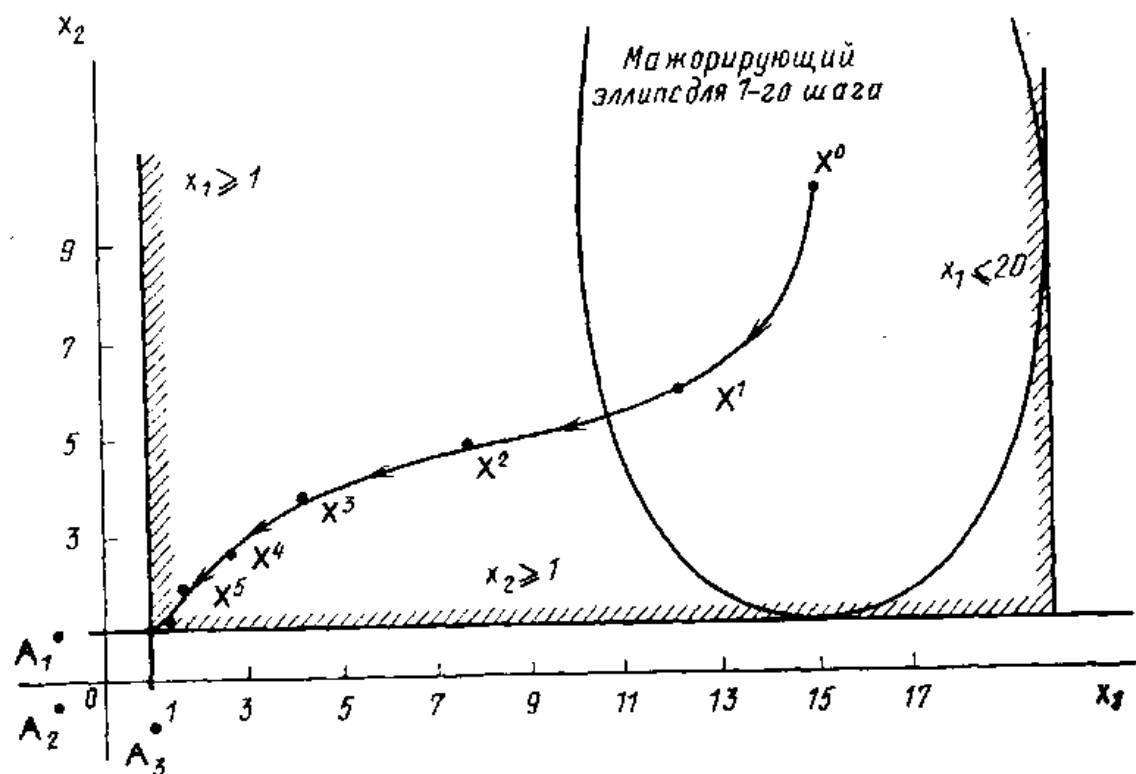


Рис. 5.4

Графическая иллюстрация данного примера представлена на рис. 5.4. Чтобы не загромождать рисунок, мажорирующий эллипс изображен только для первого шага. Хорошо видно, что точка  $x^1$  оказалась не на границе эллипса, а внутри его и вблизи границы. Еще раз отметим, что это результат приближенного определения величины  $\lambda$ . Ход и результаты итерационного процесса для данного примера приведены в табл. 5.1.

## 5.2. МАЖОРИРОВАНИЕ ПРОИЗВОЛЬНОЙ СИСТЕМЫ ЛИНЕЙНЫХ ОГРАНИЧЕНИЙ

Покажем, как усовершенствовать описанный выше алгоритм минимизации выпуклой функции в условиях прямоугольных ограничений для того, чтобы решать задачи с ограничениями более общего вида

$$\sum_{j=1}^n A_{ij} x_j + b_i \leq 0, \quad i = 1, 2, \dots, m, \quad (5.8)$$

где без потери общности будем полагать, что

$$\sum_j A_{ij}^2 = 1, \quad i = 1, 2, \dots, m. \quad (5.9)$$

Как и прежде, выберем точку  $x^0$ , удовлетворяющую системе ограничений (5.8). Если теперь построить эллипсоид с центром в точке  $x^k$ , выбрав направления главных полуосей параллельно осям системы ко-

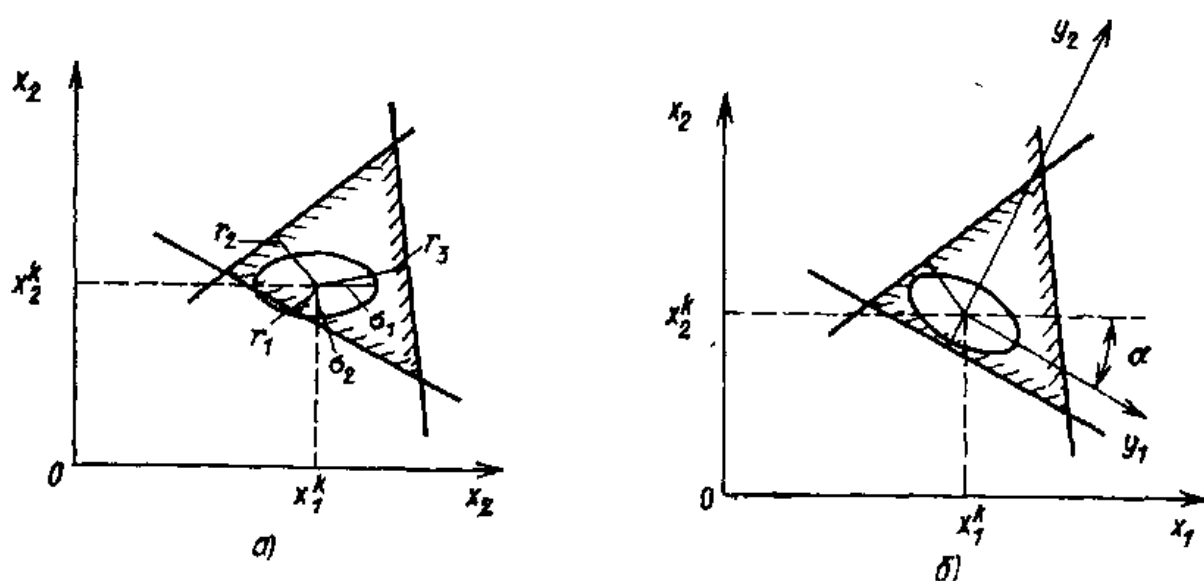


Рис. 5.5

ординат, а их размеры равными минимальным расстояниям до соответствующих ограничений, то полученный эллипсоид чаще всего не будет мажорировать исходную систему ограничений (рис. 5.5). Однако это положение можно исправить, если развернуть малую полуось эллипса перпендикулярно ближайшей линии ограничения, как это сделано на рис. 5.5, б, и выбрать направление и размеры остальных полуосей в соответствии с правилами, излагаемыми ниже.

Обозначим  $\xi^{(1)}$  основание перпендикуляра, опущенного из точки  $x^0$  на ближайшую ограничивающую гиперплоскость. Длина этого перпендикуляра

$$\delta^{(1)} = \| \xi^{(1)} - x^0 \| = \min_i \delta_i^{(1)}, \quad (5.10)$$

где

$$\delta_i^{(1)} = \frac{\| A_i^T x + b_i \|}{\| A_i^T \|} \Big|_{x=x^0}, \quad i = 1, 2, \dots, m,$$

или с учетом (5.9)

$$\delta_i^{(1)} = \| A_i^T x + b_i \|_{x=x^0},$$

а его направляющие косинусы определяются формулой

$$C^{(1)} = A_{i_1}^T / \| A_{i_1}^T \| = A_{i_1}^T,$$

$$\text{где } \| C^{(1)} \| = 1, \text{ а } i_1 = \arg \min_i \delta_i^{(1)}. \quad (5.11)$$

Назовем вектор  $\xi^{(1)} - x^0$  «первым ортом». «Второй орт» построим, воспользовавшись следующим приемом. Возьмем произвольную  $i$ -ю плоскость, входящую в состав исходных ограничений и не совпадающую с  $i_1$ -й плоскостью ( $i \neq i_1$ ). Найдем точку  $\xi_i^{(2)}$ , удовлетворяющую трем условиям:

а) точка лежит в плоскости  $A_i^T x = b_i$ , т. е.

$$A_i^T \xi_i^{(2)} = b_i; \quad (5.12)$$

б) вектор  $\xi_i^{(2)} - x^0$  ортогонален вектору  $C^{(1)}$ , т. е.

$$C^{(1)} (\xi_i^{(2)} - x^0) = 0; \quad (5.13)$$

в) этот вектор имеет наименьшую длину среди всех векторов, удовлетворяющих условиям а) и б):

$$0,5 \| \xi_i^{(2)} - x^0 \| = \min_{\xi_i^{(2)}}.$$

Для отыскания этой точки построим функцию Лагранжа

$$\begin{aligned} L = & 0,5 (\xi_1^{(2)} - x_1^0)^2 + 0,5 (\xi_2^{(2)} - x_2^0)^2 + \dots + 0,5 (\xi_n^{(2)} - x_n^0)^2 - \\ & - \lambda_1^{(2)} [C_1^{(1)} (\xi_1^{(2)} - x_1^0) + C_2^{(1)} (\xi_2^{(2)} - x_2^0) + \dots + C_n^{(1)} (\xi_n^{(2)} - x_n^0)] - \\ & - \lambda_{i_2}^{(2)} [a_{i_1} \xi_1^{(2)} + a_{i_2} \xi_2^{(2)} + \dots + a_{i_n} \xi_n^{(2)} - b_i] \end{aligned}$$

и приравняем нулю ее частные производные по  $\xi_{ij}^{(2)}$  ( $j = 1, 2, \dots, n$ ),  $\lambda_1^{(2)}$ ,  $\lambda_{i_2}^{(2)}$ . Из соотношения  $\partial L / \partial \xi_{ij}^{(2)} = 0$  после несложных преобразований получаем

$$\xi_{ij}^{(2)} - x_j^0 = \lambda_1^{(2)} C_j^{(1)} + \lambda_{i_2}^{(2)} a_{ij}, \quad j = 1, 2, \dots, n. \quad (5.14)$$

Для отыскания неопределенных множителей  $\lambda_{i_1}^{(2)}$  и  $\lambda_{i_2}^{(2)}$  умножим выражение (5.14) на  $C_j^{(1)}$  и просуммируем по всем  $j$ . Учитывая (5.13) и (5.11), имеем

$$\lambda_{i_1}^{(2)} + \lambda_{i_2}^{(2)} [A_i^T C^{(1)}] = 0, \quad i = 1, 2, \dots, n, \quad i \neq i_1. \quad (5.15)$$

Умножим (5.14) на  $a_{ij}$ , просуммируем по  $j$  с учетом (5.9) и получим

$$\lambda_{i_1}^{(2)} [A_i^T C^{(1)}] + \lambda_{i_2}^{(2)} = [A_i^T x^0] + b_i, \quad i = 1, 2, \dots, n, \quad i \neq i_1. \quad (5.16)$$

Разрешив (5.15) и (5.16) относительно  $\lambda_{i_1}^{(2)}$  и  $\lambda_{i_2}^{(2)}$  и подставив полученный результат в (5.14), найдем составляющие вектора  $\xi_i^{(2)} = x^0$ . Обозначим длину этого вектора  $\delta_i^{(2)}$ :

$$\delta_i^{(2)} = |\xi_i^{(2)} - x^0|. \quad (5.17)$$

Расстояние от точки  $x^0$  до ближайшей ограничивающей гиперплоскости в  $(n-1)$ -мерном пространстве  $C^{(1)} (x - x^0) = 0$  найдем по формуле

$$\delta^{(2)} = \min_{i, i \neq i_1} \delta_i^{(2)},$$

а ориентацию отрезка, соединяющего точку  $x^0$  с основанием перпендикуляра, опущенного на эту плоскость, по формуле

$$C^{(2)} = (\xi_{i_2}^{(2)} - x^0) / |\xi_{i_2}^{(2)} - x^0|, \quad (5.18)$$

где

$$i_2 = \arg \min_{i \neq i_1} \delta_i^{(2)}.$$

Вектор  $\xi_{i_2}^{(2)} - x^0$  назовем вторым ортом. Аналогично будем действовать и далее.

Пусть после  $k$ -й итерации становятся известны номера  $i_1, i_2, \dots, i_k$  ограничивающих плоскостей, использованных для построения первых  $k$  ортов, единичные векторы  $C^{(1)}, C^{(2)}, C^{(3)}, \dots, C^{(k)}$ , определяющие ориентацию этих ортов, и расстояния  $\delta^{(1)}, \delta^{(2)}, \delta^{(3)}, \dots, \delta^{(k)}$  от точки  $x^0$  до точки пересечения  $j$ -го орта с соответствующей плоскостью.

Вновь возьмем  $i$ -ю произвольную плоскость  $A_i^T x = b_i, i \notin (i_1, i_2, \dots, i_k)$ , ограничивающую область допустимых решений. Выберем точку  $\xi_i^{(k+1)}$ , удовлетворяющую следующим условиям:

а) точка  $\xi_i^{(k+1)}$  лежит в плоскости  $A_i^T x = b_i$ , т. е.

$$A_i^T x = b_i \quad i = 1, 2, \dots, m, \quad i \neq i_1, \quad i \neq i_2, \dots, i \neq i_k; \quad (5.19)$$

б) вектор  $\xi_i^{(k+1)} - x^0$  ортогонален векторам  $C^{(1)}, C^{(2)}, \dots, C^{(k)}$ , т. е.

$$C^{(1)} (\xi_i^{(k+1)} - x^0) = 0, \quad C^{(2)} (\xi_i^{(k+1)} - x^0) = 0, \dots$$

$$\dots, \quad C^{(k)} (\xi_i^{(k+1)} - x^0) = 0; \quad (5.20)$$



Выпуклая оболочка двух гиперсфер  $\mathcal{T}^{(1)}$  и  $\mathcal{T}^{(2)}$ , где  $\mathcal{T}^{(2)}$  — гиперсфера радиуса  $\delta^{(2)}$  в пространстве  $E^{(n-1)}$  с центром в  $x^0$ , также мажорирует многогранник ограничений. Гиперсфера  $\mathcal{T}^{(1)}$  касается  $i_1$ -го ограничения и не пересекается ни с одной из оставшихся гиперплоскостей. Гиперсфера  $\mathcal{T}^{(2)}$  лежит в плоскости  $A_i^T (x - x^0) = 0$ , параллельной плоскости  $A_{i1}^T x = b_{i1}$ , касается плоскости  $A_{i2}^T x = b_{i2}$  и не пересекается ни с одной из оставшихся гиперплоскостей (иначе след этой плоскости в плоскости  $A_{i1}^T (x - x^0) = 0$  лежал бы от точки  $x^0$  на расстоянии, меньшем  $\delta^{(2)}$ ).

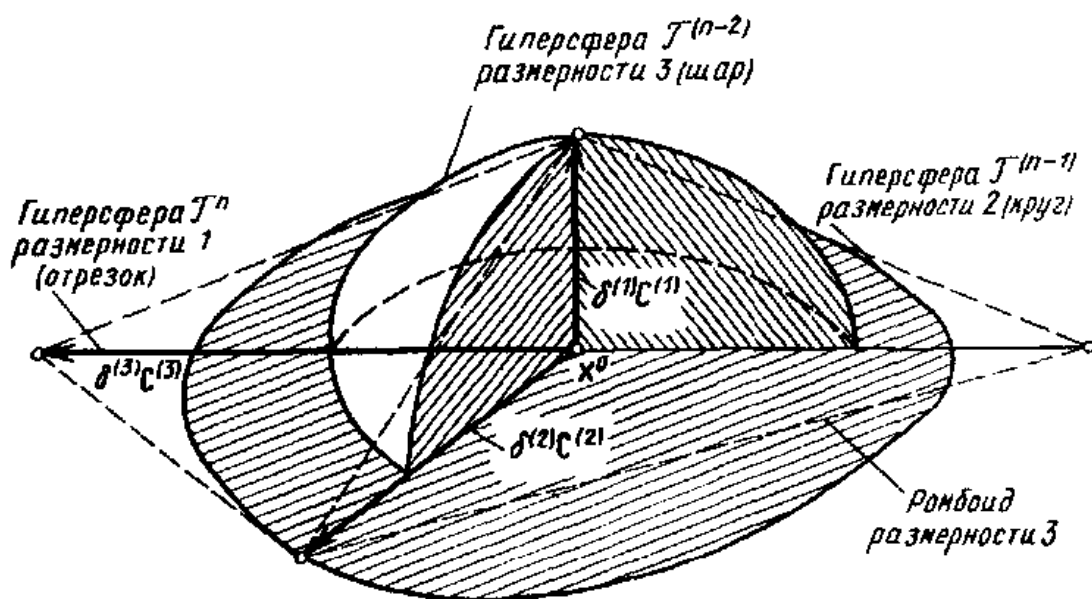


Рис. 5.6

Присоединяя к гиперсферам  $\mathcal{T}^{(1)}$  и  $\mathcal{T}^{(2)}$  гиперсферу  $\mathcal{T}^{(3)}$  радиуса  $\delta^{(3)}$ , расположенную в  $(n-2)$ -мерном пространстве, определяемом ограничениями  $C^{(1)} (x - x^0) = 0$  и  $C^{(2)} (x - x^0) = 0$ , и строя выпуклую оболочку этих трех тел, мы вновь получаем некоторое тело, вписанное в многогранник ограничений, так как  $i_1$ -я,  $i_2$ -я и  $i_3$ -я гиперплоскости касаются этого тела, а остальные не пересекают его. Рассуждая в этом же духе и дальше, мы заключаем, что выпуклая оболочка всех гиперсфер  $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \dots, \mathcal{T}^{(n)}$  заключена внутри многогранника ограничений.

В построенное выпуклое тело можно вписать гиперромб (ромбоид), натянутый на  $n$ -мерный комплекс  $\xi^{(1)} - x^0, \xi^{(2)} - x^0, \dots, \xi^{(n)} - x^0$  (рис. 5.6). Будучи частью этого тела, указанный ромбоид также мажорирует многогранник ограничений.

Формулами перехода от прямоугольных координат  $x_1, x_2, \dots, x_n$  к прямоугольным координатам  $y_1, y_2, \dots, y_n$ , связанным с комплексом  $C^{(1)}, C^{(2)}, \dots, C^{(n)}$ , являются

$$y = C^T (x - x^0). \quad (5.24)$$

Введем прямоугольные координаты  $z_1, z_2, \dots, z_n$ , связанные с координатами  $y_1, y_2, \dots, y_n$  соотношением  $z_i = y_i/\delta_i$  или

$$z = \Delta^{-1} y, \quad (5.25)$$

где

$$\Delta = \begin{vmatrix} \delta^{(1)} & 0 & \dots & 0 \\ 0 & \delta^{(2)} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \delta^{(n)} \end{vmatrix}.$$

Построенный в этом пространстве гиперромб имеет одинаковые диагонали. Теперь в него можно вписать шар радиуса  $1/\sqrt{n}$ . В самом деле, в положительном октанте пространства  $z_1, z_2, \dots, z_n$  плоскостью, ограничивающей гиперромб, будет  $z_1 + z_2 + z_3 + \dots + z_n = 1$ . Расстояние от начала координат до этой плоскости

$$\delta = |(\sum z_i - 1)/\sqrt{n}|_{z=0} = 1/\sqrt{n}. \quad (5.26)$$

На этом же расстоянии располагаются ограничивающие гиперромб плоскости в других октантах. Уравнение гиперсферы максимального радиуса, вписанной в гиперромб, будет иметь вид

$$z^T z = 1/\sqrt{n}. \quad (5.27)$$

Учитывая (5.24)–(5.27), получаем, что эллипсоид

$$(x - x^0)^T Q (x - x^0) = 1,$$

где

$$Q = C \tilde{Q} C^T, \quad C = \| C^{(1)} \ C^{(2)} \ \dots \ C^{(n)} \|, \\ \tilde{Q} = \begin{vmatrix} \sqrt{n}/\delta^{(1)} & 0 & \dots & 0 \\ 0 & \sqrt{n}/\delta^{(2)} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sqrt{n}/\delta^{(n)} \end{vmatrix},$$

мажорирует исходную систему ограничений, вписываясь в многогранный угол, ближайший к точке  $x^0$ .

В процессе реализации данной процедуры необходимо обязательно принять меры предосторожности на случай появления некоторых часто встречающихся вырождений.

1. Если на  $(k+1)$ -й итерации окажется, что вектор  $A_i$ ,  $i \notin (i_1, i_2, \dots, i_k)$ , лежит в пространстве  $C^{(1)}, C^{(2)}, \dots, C^{(k)}$ , то система условий (5.12), (5.13) противоречива, главный определитель системы (5.20) равен нулю и процесс вычислений нарушается.

В этом случае надо положить  $\delta_i^{(k+1)} = M$  ( $M$  — бесконечно большое число) и перейти к следующему  $i$ . Такая ситуация возникает, в частности, тогда, когда ограничения имеют вид  $b_i^{(1)} \leq \sum_{j=1}^n a_{ij} x_j \leq b_i^{(2)}$ .

2. Если окажется, что на  $k$ -й итерации все  $\delta_i^k = M$ ,  $i \notin (i_1, i_2, \dots, i_k)$ , то процесс приостанавливается из-за отсутствия очередного вектора  $\xi^{(k)} = x^0$ . Полученный симплекс  $C^{(1)}, C^{(2)}, \dots, C^{(k)}$  ( $k < n$ ) необходимо дополнить до  $n$ -мерного за счет присоединения любых ( $n - k$ ) ортонормальных векторов. Использовать для этих целей вектор  $\xi_i^{(k)} = x^0$  невозможно.

В этой ситуации можно взять  $n - k$  векторов  $S^{(k+1)}, S^{(k+2)}, \dots, S^{(n)}$ , построенных с помощью датчика случайных чисел, и, используя процедуру Грама — Шмидта, дополнить систему  $C^{(1)}, C^{(2)}, \dots, C^{(k)}$  до  $n$ -мерной. Датчик случайных чисел удобен тем, что при его использовании почти наверняка исключается случай мультиколлинеарности вектора  $S^{(k+1)}$  с векторами  $C^{(1)}, C^{(2)}, \dots, C^{(k)}$  и, наоборот, применение какого-либо регулярного способа формирования вектора  $S^{(k+1)}$  (например,  $S^{(k+1)} = (1, 0, 0, 0, 0, \dots, 0)$  или  $S^{(k+1)} = (1, -1, 1, -1, \dots, \pm 1)$ ) делает эту ситуацию весьма вероятной.

С процедурой ортогонализации системы векторов по Граму — Шмидту можно ознакомиться по любому учебнику линейной алгебры или функционального анализа [30]. Основной формулой в этом алгоритме является

$$C^{(k+1)} = S^{(k+1)} - \sum_{i=1}^k \frac{[C^{(i)} S^{(k+1)}]}{|S^{(i)}|} S^{(i)},$$

где, как и прежде,  $S^{(k+1)}$  — вектор, лежащий вне пространства, определяемого векторами  $C^{(1)}, C^{(2)}, \dots, C^{(k)}$ ;  $[ab]$  — символ векторного произведения;  $|a|$  — норма вектора  $a$ ; индекс в скобках, т. е.  $(k)$  — номер внутреннего цикла, связанного с построением мажорирующего эллипсоида, который в пределах данного параграфа следует отличать от номера  $k$  внешнего цикла, связанного с нахождением точки, приближающей к условному минимуму.

3. Если  $x$  приближается к границе области  $R$ , то  $\delta_i^{(k)} = \min_i \delta_i^{(k)}$  начинает принимать малые значения. Чтобы построить мажорирующий эллипсоид, в качестве одной из его полуосей надо принять  $\sigma^{(k)} = \max \{\delta^{(k)}, \varepsilon\}$ , где  $\varepsilon$  — заданное малое положительное число.

При этом не исключена вероятность того, что мажорирующий эллипсоид выйдет за пределы области ограничений на расстояние, не превышающее  $\varepsilon$ . Следовательно, точка  $x^{k+1}$  также может оказаться вне области  $R$ . Как только это случится, сразу пойдет «вразнос» вычислительный процесс, так как на следующей же итерации новый эллипсоид во всех своих точках перестает удовлетворять системе ограничений. Поэтому, если текущая точка попадает в  $\varepsilon$ -окрестность ограничений

(признаком этого является выбор  $\varepsilon$  в качестве размера одной из полуосей), необходимо проверять соблюдение системы ограничений  $R$  в точке  $x^{k+1}$  и в случае их нарушения за искомую точку минимума брать точку  $x^k$ , полученную на предшествующем шаге. Это будет одним из признаков необходимости останова. Другим признаком, как и ранее, является выполнение условия  $|f(x^{k+1}) - f(x^k)| \leq \varepsilon_1$ , где  $\varepsilon_1$  — заданное малое положительное число.

**Пример 5.2.** Задана система ограничений

$$\begin{aligned} -x_1 + 2x_2 - 14 &\leq 0, \\ x_2 - 9 &\leq 0, \\ -8x_1 - 3x_2 + 24 &\leq 0, \\ x_1 - x_2 - 3 &\leq 0. \end{aligned}$$

Необходимо построить мажорирующий эллипс с центром в точке  $x^0 = (3, 5)$ .

После нормировки в соответствии с (5.9), (5.10) исходную систему неравенств можно записать так:

$$\begin{aligned} -0,4472x_1 + 0,8944x_2 - 6,2610 &\leq 0, \\ x_2 - 9 &\leq 0, \\ -0,9363x_1 - 0,3511x_2 + 2,8090 &\leq 0, \\ 0,7071x_1 - 0,7071x_2 - 2,1213 &\leq 0. \end{aligned}$$

Расстояния от точки  $x^0$  до каждого из ограничений равны:

$$\begin{aligned} \delta_1 &= |A^T x^0 + b_1| = |-0,4472 \times 3 + 0,8944 \times 5 - 6,2610| = 3,1306, \\ \delta_2 &= |5 - 9| = 4, \\ \delta_3 &= |-0,9363 \times 3 - 0,3511 \times 5 + 2,8090| = 1,7554, \\ \delta_4 &= |0,7071 \times 3 - 0,7071 \times 5 - 2,1213| = 3,5355. \end{aligned}$$

Ближайшим к точке  $x^0$  оказалось третье ограничение, поэтому в качестве нормы первого орта возьмем  $\delta^{(1)} = 1,7554$ , а направляющие косинусы этого орта определим по (5.18) с учетом того, что

$$i_1 = 3, \quad C^{(1)} = \frac{A_3}{\|A_3\|} = \begin{pmatrix} -0,9363 \\ -0,3511 \end{pmatrix}.$$

Для нахождения второго орта последовательно перебираем все ограничения (кроме третьего) и для каждого из них составляем систему уравнений вида (5.20). Так, для первого ограничения имеем

$$\begin{aligned} \lambda_1^{(2)} + 0,1047\lambda_2^{(2)} &= 0, \\ 0,1047\lambda_1^{(2)} + \lambda_2^{(2)} &= -3,1305. \end{aligned}$$

Отсюда  $\lambda_1^{(2)} = 0,3313$ ,  $\lambda_2^{(2)} = -3,1652$ . Решение такой системы линейных уравнений легко получить стандартными методами. В работе для этих целей использовались подпрограммы MFG и MDLG из пакета прикладных программ [8]. Имея значения  $\lambda_1^{(2)}$  и  $\lambda_2^{(2)}$ , находим

$$1) \quad \xi_1^{(2)} - x^0 = \begin{pmatrix} 1,1053 \\ -2,9474 \end{pmatrix}, \quad 2) \quad \delta_1^{(2)} = \|\xi_1^{(2)} - x^0\| = 3,1478.$$

Для второго неравенства имеем

$$\begin{aligned} \lambda_{21}^{(2)} - 0,3511\lambda_{22}^{(2)} &= 0, \\ -0,3511\lambda_{21}^{(2)} + \lambda_{22}^{(2)} &= -4, \\ \text{откуда } \lambda_{21}^{(2)} &= -1,6020 \text{ и } \lambda_{22}^{(2)} = -4,5625, \\ \xi_2^{(2)} - x^0 &= \begin{vmatrix} 1,5000 \\ -4,0000 \end{vmatrix}, \quad \delta_2^{(2)} = 4,2720. \end{aligned}$$

И, наконец, для четвертого неравенства получим

$$\begin{aligned} \lambda_{41}^{(2)} - 0,4138\lambda_{42}^{(2)} &= 0, \\ -0,4138\lambda_{41}^{(2)} + \lambda_{42}^{(2)} &= -3,5355, \\ \text{откуда } \lambda_{41}^{(2)} &= -1,7653, \quad \lambda_{42}^{(2)} = -4,2660, \\ \xi_4^{(2)} - x^0 &= \begin{vmatrix} -1,3636 \\ 3,6364 \end{vmatrix}, \quad \delta_4^{(2)} = 3,8836. \end{aligned}$$

В результате 
$$\delta^{(2)} = \min_i \delta_i^{(2)} = 3,1478.$$

Следовательно, в соответствии с (5.21)  $i_2 = \arg \min_{i \neq i_1} \delta_i^{(2)} = 1.$

Направляющие косинусы второго орта определяем по (5.18):

$$C^{(2)} = \frac{\xi_1^{(2)} - x^0}{\|\xi_1^{(2)} - x^0\|} = \begin{vmatrix} 0,3511 \\ -0,9363 \end{vmatrix}.$$

С учетом этих результатов можно записать:

$$\begin{aligned} C &= \begin{vmatrix} -0,9363 & 0,3511 \\ -0,3511 & -0,9363 \end{vmatrix}, \\ \bar{Q} &= \begin{vmatrix} \sqrt{2}/1,7554 & 0 \\ 0 & \sqrt{2}/3,8836 \end{vmatrix} = \begin{vmatrix} 0,6489 & 0 \\ 0 & 0,2018 \end{vmatrix}, \\ Q &= C\bar{Q}C^T = \begin{vmatrix} 0,5938 & -0,3511 \\ 0,3511 & -0,9363 \end{vmatrix}. \end{aligned}$$

Уравнение мажорирующего эллипса имеет вид

$$\begin{vmatrix} x_1 - 3 \\ x_2 - 5 \end{vmatrix}^T \begin{vmatrix} 0,5938 & -0,3511 \\ 0,3511 & -0,9363 \end{vmatrix} \begin{vmatrix} x_1 - 3 \\ x_2 - 5 \end{vmatrix} = 1.$$

Графическая иллюстрация данного примера представлена на рис. 5.7.

Для ускорения сходимости иногда целесообразно увеличить размеры полуосей мажорирующего эллипсоида в  $\sqrt{n}$  раз. При этом, естественно, теряется гарантия того, что точка, лежащая на пересечении этого эллипсоида с линией «водостока» целевой функции, окажется внутри области ограничений (вероятность этого события невелика и убывает с ростом  $n$ ). При использовании такого приема необходимо проверять, лежит ли точка минимума, полученная на текущей итерации, внутри многогранника ограничений. Если вдруг окажется, что

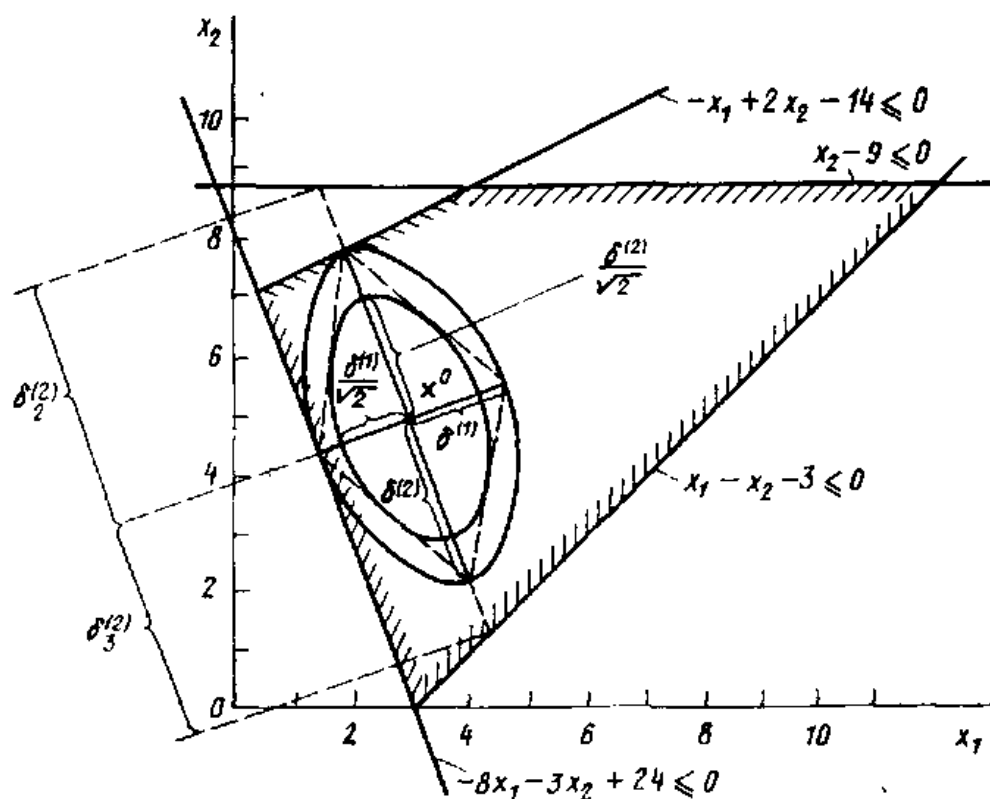


Рис. 5.7

она вышла за пределы этого многогранника, следует вернуться в предшествующую точку и построить «гарантированно мажорирующий» эллипсоид. Такие попытки ускорить поиск можно предпринимать на каждой итерации.

Как видно из рис. 5.7, в предшествующем примере размеры эллипса, построенного в точке  $x^0 = (3, 5)$ , можно увеличить в  $\sqrt{2}$ , и в данном случае мажорирующие свойства эллипса сохраняются в полной мере.

### 5.3. АЛГОРИТМ КОМПЛЕКСНОГО МАЖОРИРОВАНИЯ

Основные особенности алгоритма комплексного мажорирования намечены в § 5.1 при рассмотрении задач минимизации выпуклых функций с прямоугольными ограничениями. Теперь, когда у нас есть надежный способ мажорирования произвольной системы линейных ограничений, опишем этот алгоритм в применении к более общим задачам.

Особенно удобный вид алгоритм комплексного мажорирования приобретает тогда, когда минимизируется линейная функция. В этом случае вспомогательная задача, решаемая на каждом шаге итерационного процесса, имеет следующий вид:

$$\min C^T x \quad (5.28)$$

при  $R^0 = \{(x - x^0)^T Q^0 (x - x^0) \leq 1\}$ , решением которой является точка  $x^1$  на границе эллипсоида, где целевая функция  $\psi(x) = C^T x$  имеет минимальное значение.

Для решения вспомогательной задачи вновь воспользуемся методом множителей Лагранжа. Имеем

$$\nabla \psi = -\lambda \nabla F(x, x^0) \text{ или } C = -2\lambda Q(x - x^0).$$

Отсюда

$$(x - x^0) = -(2\lambda Q)^{-1} C. \quad (5.29)$$

Подставим полученное выражение для  $(x - x^0)$  в уравнение эллипсоида и получим

$$\begin{aligned} [(2\lambda Q)^{-1} C]^T Q [(2\lambda Q)^{-1} C] &= 1, \\ \frac{1}{4\lambda^2} (Q^{-1} C)^T Q Q^{-1} C &= 1, \\ \lambda &= \frac{1}{2} \sqrt{C^T (Q^{-1})^T C}. \end{aligned} \quad (5.30)$$

Легко заметить, что вывод формулы для определения параметра  $\lambda$  аналогичен выводу, приведенному ранее для квадратичной целевой функции, но в случае линейной функции соотношения существенно упрощаются и при этом удается легко получить уже не приближенное, а точное значение  $\lambda$ .

Таким образом, решением задачи (5.28) будет точка

$$x^1 = x^0 - Q^{-1} C / (2\lambda),$$

лежащая на границе эллипсоида.

Во вновь полученной точке  $x^1$  можно повторить процедуру построения мажорирующего эллипсоида и, решив еще раз задачу (5.28), найти точку  $x^2$  с меньшим значением целевой функции, чем в точках  $x^0$  и  $x^1$ :

$$\psi(x^0) \geq \psi(x^1) \geq \psi(x^2).$$

Действуя так и далее, получаем последовательность точек  $x^0, x^1, x^2, \dots, x^k$ , каждая из которых является решением соответствующих вспомогательных задач типа (5.28). Эта последовательность сходится к решению исходной задачи линейного программирования. Доказательство сходимости приводится в [12].

Следует отметить, что методы решения задач линейного программирования с использованием эллипсоидов, предлагаемые в данной работе, отличаются от методов, описанных в [12]. Эти отличия состоят, главным образом, в том, что в зависимости от положения точки текущего приближения меняются не только размеры эллипсоида, но и его ориентация.

В [12] эллипсоид используется в качестве мажоранты только для ограничений вида  $\alpha_j \leq x_j \leq \beta_j$ . Оси этого эллипсоида неизменно

параллельны координатным осям. При наличии в задаче ограничений более общего вида

$$\alpha_i \leq \sum_{j=1}^n a_{ij} x_j \leq \beta_i. \quad (5.31)$$

В [12] используется преобразование

$$\sum_{j=1}^n a_{ij} x_j - y_i = 0, \alpha_i \leq y_i \leq \beta_i. \quad (5.32)$$

В этом случае вспомогательная задача, решаемая на каждом шаге, помимо эллипсоидального ограничения содержит ограничения в виде равенств, а именно, требуется найти

$$\min C^T x$$

при

$$\sum_{j=1}^n a_{ij} x_j = b_i, i = 1, 2, \dots, m, x \in R^k = \left\{ x: \sum_{j=1}^n (x_j - x_j^k)^2 / (\sigma_j^k)^2 \leq 1 \right\}.$$

Преобразование (5.32) повышает размерность вспомогательной задачи и приводит к затруднениям при определении множителей Лагранжа. Задача становится несколько сложнее, чем вспомогательная задача (5.28), которую приходится решать при изменяющейся ориентации эллипсоида.

**Пример 5.3.** Решить следующую задачу линейного программирования:  
 $\min (-x_1 - 2x_2)$  при

$$\begin{aligned} -x_1 + 2x_2 &\leq 14, \\ 8x_1 + 3x_2 &\geq 24, \\ x_1 - x_2 &\leq 3, \\ x_1 + x_2 &\leq 10. \end{aligned}$$

Решение задачи легко получить геометрически (рис. 5.8). Очевидно, что оптимальной является точка (2,8), а соответствующее ей значение целевой функции равно — 18.

Это же решение можно получить и при использовании алгоритма комплексного мажорирования. В качестве точки начального приближения возьмем  $x^0 = (3,5)$ . Ход итерационного процесса и результаты решения примера представлены в табл. 5.3 и на рис. 5.8. Как уже отмечалось, иногда можно ускорить поиск, увеличив размеры мажорирующего эллипсоида в  $\sqrt{n}$  раз. В данном примере это удастся сделать на каждом из шагов. В таблице приведены результаты как для обычного эллипса, так и для увеличенного, причем во втором случае для решения потребовалось меньше итераций, так как поиск осуществляется более крупным шагом. На рис. 5.8 показаны траектории перемещения теку-

Т а б л и ц а 5.3

| Номер шага $k$ | Алгоритм с обычным эллипсом |         |          | Алгоритм с увеличенным эллипсом |                    |                      |
|----------------|-----------------------------|---------|----------|---------------------------------|--------------------|----------------------|
|                | $x_1^k$                     | $x_2^k$ | $f(x)^k$ | $x_{1\text{ув}}^k$              | $x_{2\text{ув}}^k$ | $f(x_{\text{ув}}^k)$ |
| 0              | 3,000                       | 5,000   | -13,000  | 3,000                           | 5,000              | -13,000              |
| 1              | 2,545                       | 6,571   | -15,687  | 2,357                           | 7,222              | -16,801              |
| 2              | 2,261                       | 7,330   | -16,921  | 2,052                           | 7,823              | -17,698              |
| 3              | 2,122                       | 7,687   | -17,496  | 2,023                           | 7,953              | -17,929              |
| 4              | 2,057                       | 7,854   | -17,765  | 2,002                           | 7,990              | -17,982              |
| 5              | 2,027                       | 7,932   | -17,890  | 2,002                           | 7,997              | -17,996              |
| 6              | 2,012                       | 7,968   | -17,949  | 2,000                           | 7,999              | -17,999              |
| 7              | 2,006                       | 7,985   | -17,976  | 2,000                           | 8,000              | -18,000              |
| 8              | 2,003                       | 7,993   | -17,989  |                                 |                    |                      |
| 9              | 2,001                       | 7,997   | -17,995  |                                 |                    |                      |
| 10             | 2,001                       | 7,998   | -17,998  |                                 |                    |                      |
| 11             | 2,000                       | 7,999   | -17,999  |                                 |                    |                      |
| 12             | 2,000                       | 8,000   | -18,000  |                                 |                    |                      |

щей точки в процессе поиска и мажорирующие эллипсы для первого шага. Точки  $x^1$  и  $x_{\text{ув}}^1$  выбираются точно на границах эллипсов и поэтому  $|x^1 - x^0| < |x_{\text{ув}}^1 - x^0|$ .

**Пример 5.4.** Даны координаты семи объектов на плоскости;  $A_1(-1, 1)$ ,  $A_2(-1, -0,5)$ ,  $A_3(10, -1)$ ,  $A_4(-1, 5)$ ,  $A_5(1, 4)$ ,  $A_6(-9, 12)$ ,  $A_7(-5, 9)$ . Считаем, что все объекты имеют одинаковую важность.

Необходимо разместить новый объект на минимальном суммарном расстоянии от заданных с учетом следующих ограничений:

$$\begin{aligned}
 x_1 + x_2 - 10 &\leq 0, \\
 -x_1 - x_2 + 4 &\leq 0, \\
 x_1 &\geq 0, \\
 x_2 &\geq 0,
 \end{aligned}
 \tag{5.33}$$

В данном случае целевая функция

$$f(x_1, x_2) = \sum_{i=1}^7 d_i(x_1, x_2) = \sum_{i=1}^7 \sqrt{(a_{i1} - x_1)^2 + (a_{i2} - x_2)^2}.$$

Для отыскания точки, удовлетворяющей системе линейных ограничений (5.33), воспользуемся программой POISK. Точка  $x(15, 100)$  выбрана произвольно. Результаты поиска допустимой точки приведены в табл. 5.4.

Т а б л и ц а 5.4

| Номер шага $k$ | $x_1^k$ | $x_2^k$ | Суммарная невязка | Номер шага $k$ | $x_1^k$ | $x_2^k$ | Суммарная невязка |
|----------------|---------|---------|-------------------|----------------|---------|---------|-------------------|
| 0              | 15,000  | 100,000 | 105,000           | 2              | 2,253   | 15,023  | 7,275             |
| 1              | 5,419   | 36,128  | 31,547            | 3              | 1,289   | 8,597   | 0,000             |

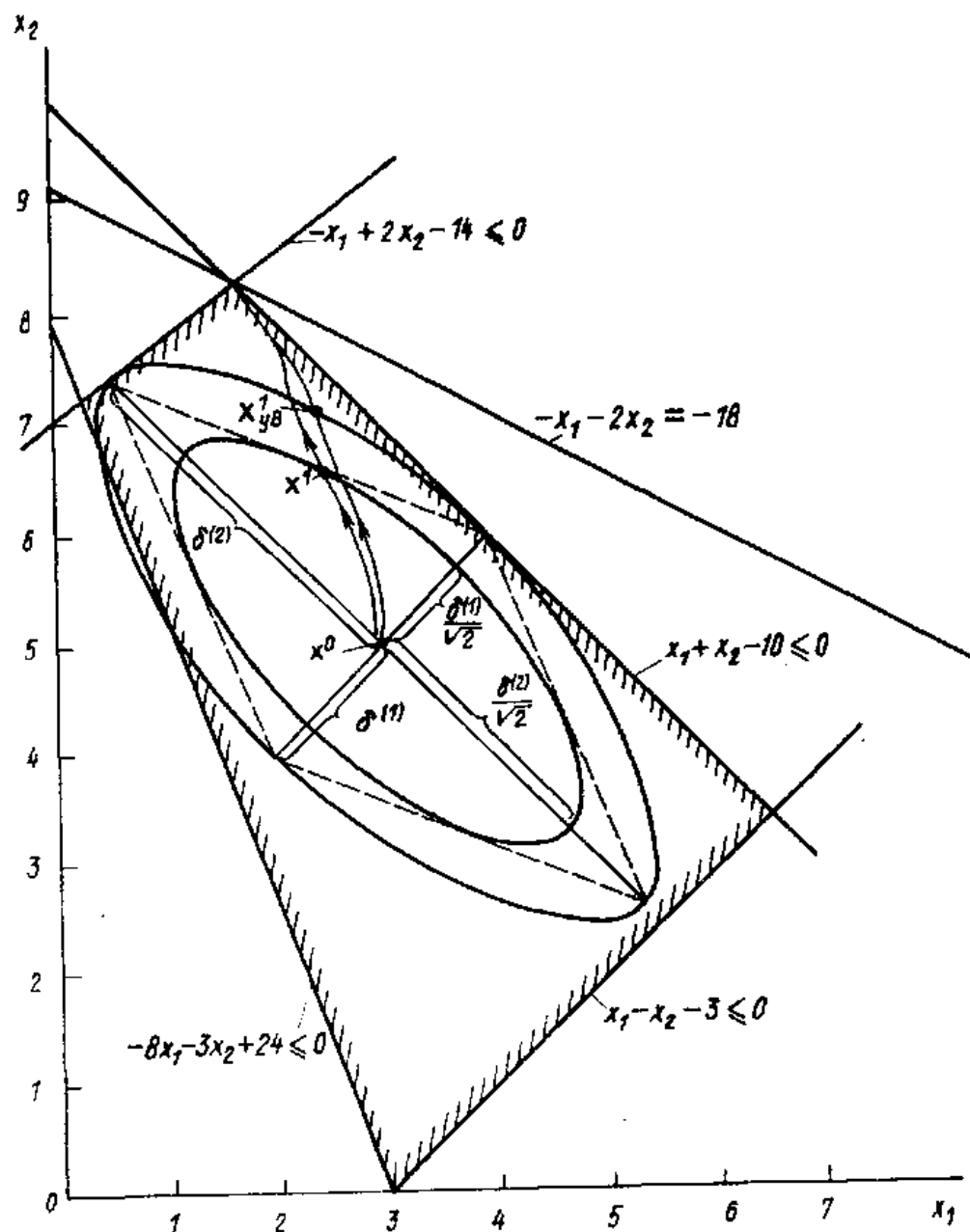


Рис. 5.8

8

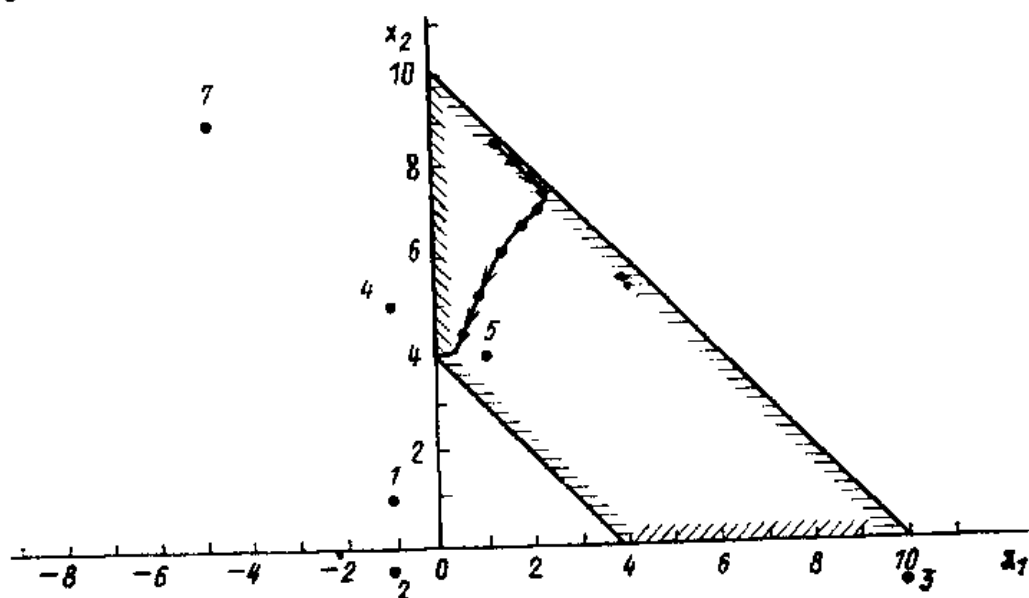


Рис. 5.9

Т а б л и ц а 5.5

| Номер шага $k$ | Алгоритм с обычным эллипсом |         |          | Алгоритм с увеличенным эллипсом |             |             |
|----------------|-----------------------------|---------|----------|---------------------------------|-------------|-------------|
|                | $x_1^k$                     | $x_2^k$ | $f(x^k)$ | $x_{1ув}^k$                     | $x_{2ув}^k$ | $f(x_{ув})$ |
| 0              | 1,289                       | 8,597   | 56,286   | 1,289                           | 8,597       | 56,286      |
| 1              | 1,719                       | 8,154   | 55,607   | 1,719                           | 8,154       | 55,607      |
| 2              | 2,093                       | 7,761   | 55,197   | 2,129                           | 7,718       | 55,147      |
| 3              | 2,327                       | 7,492   | 54,961   | 2,345                           | 7,444       | 54,850      |
| 4              | 2,404                       | 7,329   | 54,649   | 2,367                           | 7,260       | 54,288      |
| 5              | 2,363                       | 7,195   | 54,056   | 2,231                           | 7,048       | 53,131      |
| 6              | 2,233                       | 7,028   | 53,071   | 1,929                           | 6,710       | 51,057      |
| 7              | 2,005                       | 6,778   | 51,515   | 1,408                           | 6,156       | 47,761      |
| 8              | 1,646                       | 6,396   | 49,196   | 0,903                           | 5,109       | 43,563      |
| 9              | 1,131                       | 5,852   | 46,100   | 0,612                           | 4,423       | 41,574      |
| 10             | 0,902                       | 4,918   | 43,129   | 0,403                           | 4,112       | 40,935      |
| 11             | 0,662                       | 4,406   | 41,634   | 0,214                           | 4,034       | 40,686      |
| 12             | 0,481                       | 4,138   | 41,050   | 0,085                           | 4,017       | 40,556      |
| 13             | 0,311                       | 4,059   | 40,799   | 0,016                           | 4,015       | 40,497      |
| 14             | 0,178                       | 4,032   | 40,649   | 0,004                           | 3,999       | 40,482      |
| 15             | 0,086                       | 4,025   | 40,560   | 0,000                           | 4,001       | 40,480      |
| 16             | 0,032                       | 4,021   | 40,512   | 0,000                           | 4,000       | 40,479      |
| 17             | 0,014                       | 4,003   | 40,492   |                                 |             |             |
| 18             | 0,004                       | 4,004   | 40,483   |                                 |             |             |
| 19             | 0,002                       | 4,001   | 40,481   |                                 |             |             |
| 20             | 0,001                       | 4,000   | 40,480   |                                 |             |             |
| 21             | 0,000                       | 4,000   | 40,479   |                                 |             |             |

Полученную в допустимой области точку с координатами (1,289; 8,597) используем в качестве стартовой для работы алгоритма комплексного мажорирования. Ход итерационного процесса отражен в табл. 5.5 и на рис. 5.9. Для сравнения приведены результаты решения задачи алгоритмом с обычными размерами мажорирующего эллипса и алгоритмом, в котором на каждом шаге предпринимается попытка ускорить поиск за счет построения эллипса с увеличенными в  $\sqrt{n}$  раз полуосями.

Как видим, во втором случае для получения решения с точностью  $\varepsilon = 0,001$  потребовалось на пять итераций меньше, при этом время решения на ЭВМ ЕС-1045 сократилось с 2,92 до 2,37 с. Следует отметить, что в задачах большой размерности и при более благоприятном выборе стартовой точки выигрыш во времени и числе итераций может оказаться существенным.

Рассмотрим еще один пример использования алгоритма комплексного мажорирования для решения задачи аппроксимации с ограничениями. Пример заимствован из [17].

**Пример 5.5.** Имеется вектор наблюдений

|         |      |      |      |      |
|---------|------|------|------|------|
| $t_i^*$ | 0,25 | 0,50 | 0,50 | 0,80 |
| $f_i^*$ | 0,50 | 0,60 | 0,70 | 1,20 |

Необходимо найти аппроксимирующую линию  $f(t) = x_1 t + x_2$  с учетом следующих ограничений:  $f'(t) \geq 0$ ,  $f(0) \geq 0$ ,  $f(1) \leq 1$ .

В качестве критерия качества аппроксимации используем минимум суммы квадратов отклонений. Представим задачу следующим образом: найти

$$\min_{x_1, x_2} \sum_{i=1}^4 (x_1 t_i^* + x_2 - f_i^*)^2$$

при  $x_1 \geq 0$ ,  $x_2 \geq 0$ ,  $x_1 + x_2 \leq 1$ .

В данном случае нет необходимости в построении специальной мажоранты для целевой функции. Целевая функция

$$\Psi(x) = (Bx - C)^T P (Bx - C),$$

где

$$B = \begin{bmatrix} 0,25 & 1 \\ 0,50 & 1 \\ 0,50 & 1 \\ 0,80 & 1 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 0,50 \\ 0,60 \\ 0,70 \\ 1,20 \end{bmatrix}.$$

сама является квадратичной формой, которую легко преобразовать к виду (5.3). Поэтому использование алгоритма комплексного мажорирования в таких задачах упрощается.

Сначала нужно отыскать точку внутри области ограничений. Здесь это легко сделать из чисто геометрических соображений. Возьмем  $x = (0,224 \ 0,281)$ . В более сложных задачах можно воспользоваться подпрограммой POISK.

Ход вычислений по методу комплексного мажорирования и полученные результаты сведены в табл. 5.6. Применялся алгоритм с попытками ускорения поиска на каждом шаге.

Т а б л и ц а 5.6

| Номер шага $k$ | $x_1^k$ | $x_2^k$ | $\Psi(x_1^k, x_2^k)$ | Номер шага $k$ | $x_1^k$ | $x_2^k$ | $\Psi(x_1^k, x_2^k)$ |
|----------------|---------|---------|----------------------|----------------|---------|---------|----------------------|
| 0              | 0,224   | 0,281   | 0,711                | 4              | 0,573   | 0,405   | 0,124                |
| 1              | 0,273   | 0,391   | 0,386                | 5              | 0,611   | 0,384   | 0,117                |
| 2              | 0,340   | 0,481   | 0,210                | 6              | 0,621   | 0,379   | 0,115                |
| 3              | 0,452   | 0,470   | 0,152                | 7              | 0,621   | 0,379   | 0,114                |

Уравнение аппроксимирующей прямой имеет вид  $f(t) = 0,621 t + 0,379$ .

Алгоритм комплексного мажорирования применялся нами для решения задач с линейными ограничениями в форме неравенств. Круг решаемых задач можно расширить, включив в число ограничений выпуклые тела, описываемые кривыми второго порядка. Соответствующий алгоритм также предполагает построение  $n$ -мерного симплекса, который позволяет последовательно вписать в область ограничений ряд гиперсфер, затем их выпуклую оболочку, затем  $n$ -мерный ромбоид и, наконец, искомый  $n$ -мерный мажорирующий эллипсоид.

Метод штрафных функций основан на преобразовании исходной задачи с ограничениями в последовательность задач безусловной оптимизации. Штрафная функция добавляется к целевой функции исходной задачи таким образом, что нарушение любого из ограничений становится невыгодным с точки зрения трансформированной задачи безусловной оптимизации. Штрафные функции обеспечивают либо быстрое возвращение в допустимую область, либо невозможность выхода за ее пределы. В зависимости от этого различают метод собственно штрафных и метод барьерных функций.

Рассмотрим следующую задачу: найти

$$\min f(x) \quad (5.34)$$

при  $R_i(x) \leq 0, i = 1, 2, \dots, m$ .

Преобразуем эту задачу в задачу без ограничений: найти

$$\min T(x, t) = f(x) + tQ(x), \quad (5.35)$$

где  $T(x, t)$  — вспомогательная функция;  $Q(x)$  — специальным образом построенная штрафная функция;  $t$  — некоторое положительное число, называемое коэффициентом штрафа.

Штрафная функция  $Q(x)$  должна удовлетворять следующим условиям:

а)  $Q(x) = 0$  внутри и на границе допустимой области, т. е. при  $x$ , удовлетворяющих  $R_i(x) \leq 0$ ;

б)  $Q(x) > 0$ , если нарушается хотя бы одно из ограничений  $R_i(x) \leq 0, i = 1, 2, \dots, m$ .

Оптимизация с помощью штрафных функций состоит в следующем [8]. Выбирается такая последовательность  $\{t_k\}, t_k \geq 0, k = 1, 2, 3, \dots$ , что  $t_k \rightarrow \infty$  при  $k \rightarrow \infty$ . Для первого элемента этой последовательности ( $k = 1$ ) с помощью одного из методов безусловной оптимизации отыскивается безусловный минимум вспомогательной функции  $T(x, t_1)$ . Пусть этот минимум достигается в точке  $x^1$ . Используя вектор  $x^1$  в качестве начального приближения для минимизации  $T(x, t_2)$ , где  $t_2 > t_1$ , получаем  $x^2$  и т. д. Таким образом, решается последовательность задач безусловной минимизации вспомогательной функции  $T(x, t_k), k = 1, 2, 3, \dots$ , при этом результат решения предыдущей задачи  $x^k$  выбирается в качестве начального приближения для получения  $x^{k+1}$ . При определенных условиях для бесконечно возрастающей последовательности  $t_k$  безусловные минимумы функций  $T(x, t_k)$  приближаются к допустимой области и последовательность  $x^k, k = 1, 2, 3, \dots$ , сходится к оптимуму, расположенному внутри или на границе допустимой области.

Что касается выбора конкретных разновидностей штрафных функций, то для задач с ограничениями вида  $R_i(x) = \sum_{j=1}^n a_{ij}x_j + b_i \leq 0$ ,

$i = 1, 2, \dots, m$ , может быть использована любая из следующих вспомогательных функций:

$$T(x, t) = f(x) + t \sum_{i=1}^m \max\{0, R_i(x)\}; \quad (5.36)$$

$$T(x, t) = f(x) + t \sum_{i=1}^m [\max\{0, R_i(x)\}]^2; \quad (5.37)$$

$$T(x, t) = f(x) + t \max\{0, R_1(x), R_2(x), \dots, R_m(x)\}; \quad (5.38)$$

$$T(x, t) = f(x) + t [\max\{0, R_1(x), R_2(x), \dots, R_m(x)\}]^2. \quad (5.39)$$

Чаще предпочтение отдается квадратичным штрафам (5.37) и (5.39), при этом в качестве главной причины большинство авторов отмечают отсутствие достаточно эффективных универсальных алгоритмов безусловной негладкой оптимизации [2, 19, 25]. Однако решение вспомогательной задачи (5.35) при использовании штрафной функции (5.37) будет приближаться к оптимальному решению исходной задачи (5.34) только при достаточно большом  $t_k$ . В то же время выбор больших  $t_k$  часто приводит к непреодолимым вычислительным трудностям: усиливается овражность вспомогательной функции и снижается эффективность методов безусловной оптимизации (приходится решать слишком длинные серии подзадач безусловной оптимизации). В отличие от функции (5.37) негладкая штрафная функция (5.36) хороша тем, что уже при конечном значении  $t_k$  обеспечивается совпадение безусловного минимума суммы  $f(x) + t_k Q(x)$  с решением исходной задачи [6, 19, 25]. Использование негладкого штрафа вдвойне оправдано в тех задачах, где сама целевая функция недифференцируема.

Если в задачах со штрафами в качестве метода безусловной минимизации применяется метод ВВКМ, то безразлично, какую функцию штрафа применять. Метод успешно использовался нами ранее для минимизации как функций  $\max\{0, y\}$ , так и функций  $[\max\{0, y\}]^2$ . Однако, учитывая приведенные выше соображения в пользу негладкой штрафной функции, будем ориентироваться главным образом на использование негладкого штрафа.

Рассмотрим применение метода штрафных функций в задаче Штейнера с ограничениями. Пусть на плоскости задана совокупность точек  $A(a_{11}, a_{12})$ ,  $A_2(a_{21}, a_{22}), \dots, A_N(a_{N1}, a_{N2})$  с весовыми коэффициентами  $w_1, w_2, \dots, w_N$ . Требуется найти точку  $x(x_1, x_2)$  на минимальном суммарном расстоянии от этих точек при наличии линейных ограничений на область размещения искомой точки.

Можно записать задачу более формально: найти

$$\min f(x) \quad (5.40)$$

при  $R_i(x) = c_{i1}x_1 + c_{i2}x_2 + b_i \leq 0, i = 1, 2, \dots, m,$

где

$$f(x) = \sum_{i=1}^N w_i \sqrt{(x_1 - a_{i1})^2 + (x_2 - a_{i2})^2}.$$

В качестве функции штрафа возьмем

$$Q(x) = \sum_{i=1}^m \max(0, c_{i1} x_1 + c_{i2} x_2 + b_i)$$

и составим вспомогательную функцию для безусловной минимизации в виде

$$T(x, t_k) = f(x) + Q(x) t_k = f(x) + t_k \sum \max(0, c_{i1} x_1 + c_{i2} x_2 + b_i), \quad (5.41)$$

где  $t_k$  — коэффициент функции штрафа.

Зафиксировав параметр  $t_k$ , построим мажоранты для каждого из слагаемых вспомогательной функции в начальной точке  $x^0$ . При решении задачи Штейнера без ограничений итерационный процесс обычно начинают из точки средних координат («центра тяжести») уже размещенных объектов [36]. Это позволяет сократить время поиска оптимальной точки, так как во многих задачах (но далеко не во всех) оптимальная точка и центр тяжести близки. Следовательно, определяем

$$x_1^0 = \sum_{i=1}^N w_i a_{i1} / \sum_{i=1}^N w_i, \quad x_2^0 = \sum_{i=1}^N w_i a_{i2} / \sum_{i=1}^N w_i.$$

Используем ранее приобретенный опыт построения мажорант (§ 3.1) для штрафной функции типа «срезка»:  $\max\{0, \sum c_{ij} x_j + b_i\}$  (пример 2.3). Мажорирующий параболоид для нее имеет вид

$$\psi_1(x_1, x_2) = \sum_{i=1}^N \frac{w_i [(x_1 - a_{i1})^2 + (x_2 - a_{i2})^2]}{2 \sqrt{(x_1^0 - a_{i1})^2 + (x_2^0 - a_{i2})^2} + \epsilon}, \quad (5.42)$$

где  $\epsilon$  — заданное малое положительное число. Мажорантой для второго слагаемого вспомогательной функции (5.35) в точке  $x^0$  будет

$$\begin{aligned} \psi_2(x_1, x_2) = & \sum_{i=1}^m \frac{t_k}{4|c_{i1} x_1^0 + c_{i2} x_2^0 + b_i| + \epsilon} \times \\ & \times |c_{i1} x_1 + c_{i2} x_2 + b_i - |c_{i1} x_1^0 + c_{i2} x_2^0 + b_i||^2. \end{aligned} \quad (5.43)$$

Сумма мажорант (5.42) и (5.43) является мажорантой для вспомогательной функции (5.35). Введем следующие обозначения:

$$p_{1i}^0 = w_i (2 \sqrt{(x_1^0 - a_{i1})^2 + (x_2^0 - a_{i2})^2} + \epsilon)^{-1}, \quad (5.44)$$

$$p_{2i}^0 = t_k (2 |c_{i1} x_1^0 + c_{i2} x_2^0 + b_i| + \epsilon)^{-1} \quad (5.45)$$

и запишем выражение мажоранты для вспомогательной функции в целом

$$\Phi(x_1, x_2) = \sum_{i=1}^N p_{1i}^0 (x_1 - a_{1i})^2 + \sum_{i=1}^N p_{2i}^0 (x_2 - a_{2i})^2 + \\ + \sum_{i=1}^m p_{2i}^0 (c_{i1} x_1 + c_{i2} x_2 + b_i - |c_{i1} x_1^0 + c_{i2} x_2^0 + b_i|)^2. \quad (5.46)$$

Это выражение описывает параболоид, коэффициенты которого  $p_{1i}^0$  и  $p_{2i}^0$  являются функциями координат точки начального приближения  $x^0 = (x_1, x_2)$ . В соответствии с общей идеей метода ВВКМ необходимо найти минимум этого параболоида, используя программу обычного метода наименьших квадратов. В точке минимума вновь строится подобный параболоид и отыскивается решение по методу наименьших квадратов. Многократно повторяя эти действия, мы в конце концов получим точку минимума мажорируемой функции.

Таким образом, для определения очередного приближения к точке минимума вспомогательной функции надо решить систему из  $2N + m$  линейных уравнений, при этом весовые коэффициенты для первых  $2N$  уравнений определяются по (5.44), а для оставшихся  $m$  уравнений — по (5.45). В качестве критерия остановки процесса поиска можно использовать абсолютную разность значений целевой функции, рассчитанных на  $k$ -м и  $(k + 1)$ -м шагах.

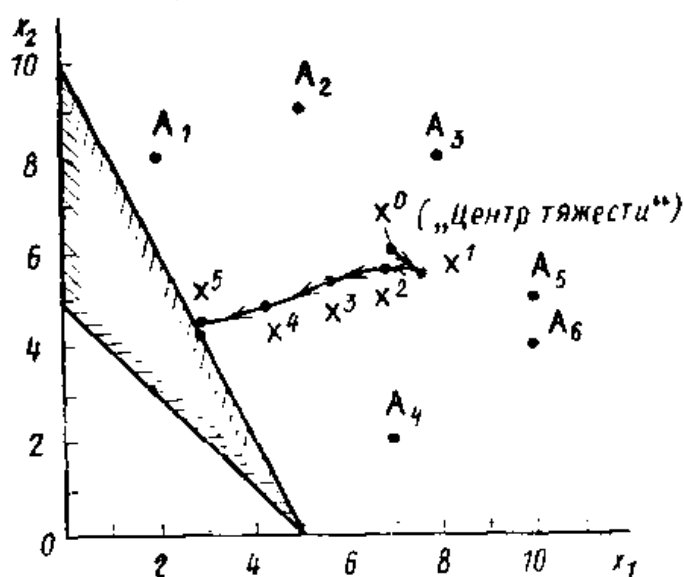
Еще раз напомним, что описанный выше алгоритм минимизации вспомогательной функции справедлив для  $t_k = \text{const}$ . Поэтому для решения задачи (5.40) методом штрафных функций необходимо решить последовательность задач минимизации вспомогательных функций при увеличивающихся значениях  $t_k$ . Это делается для того, чтобы подобрать нужное значение параметра  $t_k$ . Здесь, в отличие от гладких штрафных функций, существует конечное пороговое значение  $t_d$  такое, что решение, полученное в результате минимизации вспомогательной функции  $T(x, t_k)$ , будет точкой ее безусловного минимума при любом  $t_k > t_d$ . Следовательно, здесь нет необходимости неограниченно увеличивать  $t_k$ , следует только подобрать его хорошее значение.

К сожалению, надежных методов для определения порогового значения параметра нет и приходится пользоваться некоторыми оценками. Рекомендации по определению  $t_k$  приведены в [6]. В частности, на практике часто подходит  $t = \max \{100, 100|f(x^0)|\}$ . Если известно хорошее начальное приближение  $x^0$ , то предпочтительнее взять меньшее значение  $t = \max \{1, |f(x^0)|\}$ . Уменьшение штрафной добавки  $tQ(x)$  до нуля свидетельствует о том, что полученное решение находится внутри или на границе допустимой области, где штрафная функция не оказывает влияния, и, следовательно,  $T(x, t_k) = f(x)$ .

**Пример 5.6.** На плоскости заданы шесть точек  $A_1(2,8)$ ,  $A_2(5,9)$ ,  $A_3(8,8)$ ,  $A_4(7,2)$ ,  $A_5(10,5)$ ,  $A_6(10,4)$ . Необходимо найти точку на минимальном суммарном расстоянии от совокупности этих точек при заданной системе линейных ограничений:

$$\begin{aligned}x_1 &\geq 0, \\2x_1 + x_2 - 10 &\leq 0, \\x_1 + x_2 - 5 &\geq 0.\end{aligned}$$

Считаем, что все точки имеют одинаковую важность  $w_i = 1, i = 1, \dots, 6$ , а в качестве стартовой точки выбираем «центр тяжести»  $x^0(7,6)$ . Геометрическая иллюстрация примера представлена на рис. 5.10. Ход решения и результаты сведены в табл. 5.7. Как видим, задача решена при конечном значении  $t = 5$ . Точка  $x^0$  выбрана произвольно, она не входит в последовательность решений вспомогательной задачи. Этим объясняется ее расположение в стороне от траектории движения частных решений вспомогательной задачи к минимуму.



Т а б л и ц а 5.7

| Номер шага $k$ | $t(k)$ | $x^k$   |         | $f(x)^k$ | $T(x^k, t)^k$ |
|----------------|--------|---------|---------|----------|---------------|
|                |        | $x_1^k$ | $x_2^k$ |          |               |
| 0              | 0,1    | 7,641   | 5,995   | 21,720   | 22,848        |
| 1              | 1      | 6,790   | 5,772   | 22,279   | 31,581        |
| 2              | 2      | 5,651   | 5,371   | 24,267   | 37,613        |
| 3              | 3      | 4,294   | 4,900   | 28,161   | 38,894        |
| 4              | 4      | 2,854   | 4,474   | 34,042   | 34,745        |
| 5              | 5      | 2,780   | 4,440   | 34,384   | 34,384        |

Рис. 5.10

В заключение приведем соотношения для решения многомерной задачи размещения при линейных ограничениях. При фиксированном  $t_k$  очередное приближение к точке минимума вспомогательной задачи можно определить следующим образом:

$$\begin{aligned}x^{k+1} = \operatorname{argmin}_x \left\{ \sum_{i=1}^n \sum_{j=1}^N P_{1i}^k (x_j - a_{ji})^2 + \right. \\ \left. + \sum_{i=1}^m P_{2i}^k \left( \sum_{j=1}^n c_{ij} x_j + b_i - \left| \sum_{j=1}^n c_{ij} x_j^k + b_i \right| \right) \right\},\end{aligned}$$

где  $n$  — размерность пространства, а соответствующие весовые коэффициенты рассчитываются по формулам:

$$P_{1i}^k = \frac{w_i}{2 \left| \sqrt{\sum_{j=1}^n (x_j^k - a_{ji})^2} \right| + \epsilon}, \quad P_{2i}^k = \frac{t_k}{4 \left| \sum_{j=1}^n c_{ij} x_j^k + b_i \right| + \epsilon}.$$

Круг задач, которые могут быть решены методом ВВКМ с использованием штрафных функций, не исчерпывается задачами размещения. Такой подход возможен и в других задачах квадратического и линейного программирования [25, 31].

### 5.5. ПОИСКОВЫЕ МЕТОДЫ С МАЖОРИРОВАНИЕМ ОГРАНИЧЕНИЙ

К поисковым относят методы, которые не требуют вычисления производных, т. е. методы нулевого порядка. В соответствующих алгоритмах направления движения к минимуму определяются на основе последовательных вычислений целевой функции  $f(x)$  в пробных точках. Алгоритмы отличаются друг от друга, главным образом, числом и способом назначения этих точек. Большинство поисковых методов носит эвристический характер. Сходятся они, как правило, медленнее, чем градиентные методы и методы второго порядка, особенно в задачах без ограничений. Однако на практике методам, использующим производные, присущ ряд недостатков. Например, даже при не очень большом числе переменных сложно выписать аналитические выражения для производных, а при использовании разностных схем для оценки производных вносятся ошибки, ограничивающие возможности этого подхода, особенно в окрестности точки минимума. Для поисковых методов не требуется гладкости функции. Кроме того, методы первого и второго порядка в отличие от поисковых требуют больших вычислительных затрат на подготовку задачи. В силу перечисленных выше причин алгоритмы прямого поиска часто оказываются более подходящими с точки зрения пользователей.

В задачах с ограничениями поисковые алгоритмы испытывают затруднения, связанные, в основном, с проверкой допустимости получаемых в процессе поиска точек. Попытки приспособить эти методы для решения задач на условный экстремум пока не дают ощутимых результатов. Алгоритмы становятся сложнее, а эффективность поиска резко снижается.

Один из возможных способов усовершенствования алгоритмов поискового типа для решения задач с ограничениями основан на использовании в процессе поиска мажорирующих эллипсоидов (с целью упрощения проверки допустимости получаемых точек). Рассмотрим два соответствующих алгоритма.

**Алгоритм случайного поиска.** В произвольной точке  $x_0$ , удовлетворяющей системе линейных ограничений  $x_0 \in R$ , построим мажорирующий эллипсоид. Порядок и правила построения такого эллипсоида рассматривались ранее (см. § 5.1, 5.2). С помощью датчика случайных чисел генерируем  $p$  точек, равномерно распределенных в объеме эллипсоида  $R^0 = \{x: \sum_{i=1}^n (x_i - x_i^0)^2 / \sigma_i^2 \leq 1\}$ .

Для генерирования случайной точки в пределах мажорирующего эллипсоида достаточно:

а) используя датчик случайных чисел, получить  $n$ -мерный случайный вектор  $y$ , такой, что его составляющие  $y_i$  равномерно распределены в интервалах  $[x_{0i} - \sigma_i, x_{0i} + \sigma_i]$ ,  $i = 1, 2, \dots, n$ , где  $\sigma_i$  — соответствующие длины полуосей эллипсоида, а  $x_{0i}$  — координаты центра эллипсоида;

б) проверить, соответствует ли полученный вектор неравенству

$$(y - x_0)^T Q^{-1} (y - x_0) \leq 1,$$

где

$$Q = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n^2 \end{bmatrix}.$$

Рекомендуемое число пробных точек обычно равно  $2n$ .

Выбираем из них точку с минимальным значением целевой функции и строим в ней новый мажорирующий эллипсоид. В этом эллипсоиде вновь случайным образом назначаем требуемое число пробных точек, рассчитываем в них значения целевой функции и выбираем точку с наименьшим значением  $f(x)$  для организации очередного шага. Эти действия повторяем до тех пор, пока удастся улучшить значение целевой функции. Если в пределах некоторого эллипсоида не удастся за определенное число попыток найти точку, «лучшую», чем центр эллипсоида, то этот центр и принимается в качестве экстремальной точки.

**Алгоритм случайного поиска с ускорением.** При построении этого алгоритма заимствован принцип ускорения поиска заменой «самой плохой» точки на симметричную ей относительно «центра тяжести» остальных. Этот прием используется в так называемом комплексном методе и в методе многогранника [6,32].

Выберем точку начального приближения в допустимой области  $x_0^k \in R$ . Построим эллипсоид, мажорирующий исходную систему ограничений, с центром в точке  $x_0^k$ . С помощью датчика случайных чисел генерируем  $p = 2n$  точек  $x_i^k$ , расположенных внутри эллипсоида, и рассчитываем в каждой из этих точек и точке  $x_0^k$  значение целевой функции.

Находим среди этого набора точки  $x_{\max}^k$  и  $x_{\min}^k$ , в которых целевая функция принимает наибольшее и наименьшее значения:

$$f(x_{\max}^k) = \max \{f(x_0^k), f(x_1^k), \dots, f(x_n^k)\},$$

$$f(x_{\min}^k) = \min \{f(x_0^k), f(x_1^k), \dots, f(x_n^k)\}.$$

Рассчитываем координаты центра «тяжести» всех полученных точек за исключением точки с «наихудшим» значением  $f(x)$ :

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

Определим точку  $x_{отр}^k$ , симметричную  $x_{max}^k$  относительно рассчитанного центра «тяжести»:

$$x_{отр}^k = x_c^k + \alpha (x_c^k - x_{max}^k),$$

где  $\alpha$  — коэффициент отражения (рекомендуемое значение  $\alpha = 1,3$ ). Если окажется, что  $f(x_{отр}^k) \leq f(x_{min}^k)$ , и, кроме того, точка  $x_{отр}^k$  находится внутри эллипсоида, т. е. соблюдается условие

$$(x_{отр}^k - x_0^k)^T Q^{-1} (x_{отр}^k - x_0^k) \leq 1,$$

то принимаем  $x_{отр}^k$  в качестве начальной точки для следующего шага. Если при  $f(x_{отр}^k) \leq f(x_{min}^k)$  точка  $x_{отр}^k$  вышла за пределы эллипсоида, то в качестве исходной точки для организации очередного шага попытаемся взять точку  $x_n^k$  пересечения вектора  $x_{отр}^k - x_{max}^k$  с границей эллипсоида (рис. 5.11).

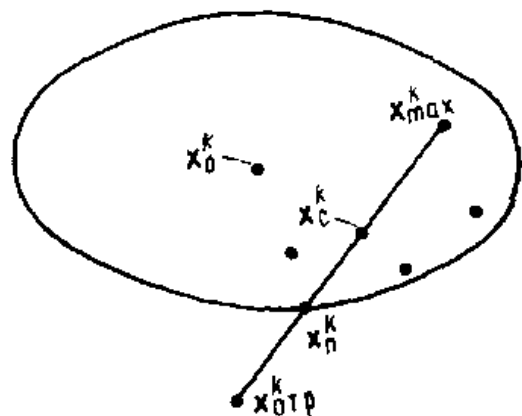


Рис. 5.11

Для определения координат пересечения вектора  $x_{отр}^k - x_{max}^k$  с границей эллипсоида запишем уравнения

$$\begin{aligned} (x - x_0)^T Q^{-1} (x - x_0) &= 1, \\ x &= x_0 + \lambda (x_{отр}^k - x_{max}^k). \end{aligned} \quad (5.47)$$

Обозначим  $x_{отр}^k - x_{max}^k = z$  и подставим значение  $x$  из второго уравнения (5.47) в первое. Имеем

$$(x_c + \lambda z - x_0)^T Q^{-1} (x_c + \lambda z - x_0) = 1$$

или

$$(x_c^T Q^{-1} + \lambda z^T Q^{-1} - x_0^T Q^{-1}) (x_c + \lambda z - x_0) = 1.$$

После несложных преобразований получаем квадратное уравнение относительно  $\lambda$ :

$$\begin{aligned} \lambda^2 z^T Q^{-1} z + \lambda (z^T Q^{-1} x_c + x_c^T Q^{-1} z - x_0^T Q^{-1} z - z^T Q^{-1} x_0) + \\ + (x_c^T Q^{-1} x_c - x_0^T Q^{-1} x_c - x_c^T Q^{-1} x_0 + x_0^T Q^{-1} x_0 - 1) = 0, \end{aligned} \quad (5.48)$$

положительный корень которого является необходимым размером шага вдоль вектора  $z = x_{отр}^k - x_{max}^k$  для определения точки на границе эллипсоида.

Оценим значение  $f(x_n^k)$  и, если  $f(x_n^k) < f(x_{min}^k)$ , то возьмем эту точку в качестве исходной для следующего шага.

В том случае, когда  $f(x_{отр}^k) \geq f(x_{min}^k)$  или  $f(x_n^k) \geq f(x_{min}^k)$ , считаем, что попытка ускорить поиск неудачна и в качестве точки для построения очередного мажорирующего эллипсоида возьмем  $x_{min}^k$ . В точке,

выбранной для очередного шага, повторяем действия предшествующего шага: строим эллипсоид, генерируем  $p = 2n$  точек внутри эллипсоида, пытаемся ускорить поиск за счет отражения самой «плохой» из точек и выбираем «лучшую» точку. Если в течение нескольких шагов не удастся улучшить значение целевой функции  $f(x_{\min}^k)$ , то точку  $x_{\min}^k$  можно считать решением задачи.

## ЗАКЛЮЧЕНИЕ

Описанные в данной главе алгоритмы во многих случаях носят экспериментальный характер. Вместе с тем проведенные эксперименты, как и результаты, полученные в [12], показывают, что методы решения задач условной оптимизации, в которых используются квадратичные мажоранты целевой функции и ограничений, имеют хорошую перспективу. К достоинствам этих алгоритмов можно отнести ясный геометрический смысл, простоту машинной реализации и возможность использования стандартного программного обеспечения.

## Приложение.

### ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МЕТОДА ВВКМ

Часто некоторые алгоритмы, очень привлекательные «на бумаге», оказываются совсем не таковыми на практике вследствие сложности и неэффективности их машинной реализации. И напротив, другие, теоретически не столь красивые алгоритмы, могут оказаться более «практичными», поскольку их реализация проста и требует небольших затрат труда программистов. К числу таких практических алгоритмов с уверенностью можно отнести и алгоритмы, опирающиеся на метод ВВКМ. Ниже приводятся некоторые советы по составлению программ для решения ранее рассмотренных задач и тексты программ на языке ПЛ/1.

В представленных программах использованы наиболее общие средства алгоритмического языка ПЛ/1, а сами программы носят главным образом учебно-иллюстративный характер, однако их можно применять и для решения практических и исследовательских задач.

#### 1. Краткое описание программ

В случае решения задач безусловной оптимизации основу любого из рассмотренных ранее алгоритмов составляет итерационный процесс, на каждом шаге которого решается система уравнений методом наименьших квадратов. В состав программного обеспечения современных ЭВМ, как правило, входит стандартная подпрограмма метода наименьших квадратов. В дальнейшем мы будем ориентироваться на подпрограмму MLSQ, включенную в пакет научных подпрограмм (ПНП) на языке ПЛ/1 [8] и предназначенную для решения избыточной системы уравнений вида  $Ax = b$ .

Решение любой из приведенных в книге задач методом ВВКМ можно изобразить в виде схемы, представленной на рис. П.1. Чтобы легче было ориентироваться в машинных программах, рассмотрим отдельные элементы этой схемы.

Ввод и распечатка исходных данных. Все программы используют ввод, управляемый списком (оператор GET LIST), что позволяет упростить процесс подготовки исходных данных. Печать матриц производится подпрограммой, обращение к которой осуществляется оператором

CALL PTMATR ('матрица A', A);

Формирование матрицы A и вектора b для системы уравнений  $Ax = b$ . Формирование матриц A и b во всех задачах определяется существом решаемой задачи, однако везде — это присвоение исходных значений элементам массива. Этап

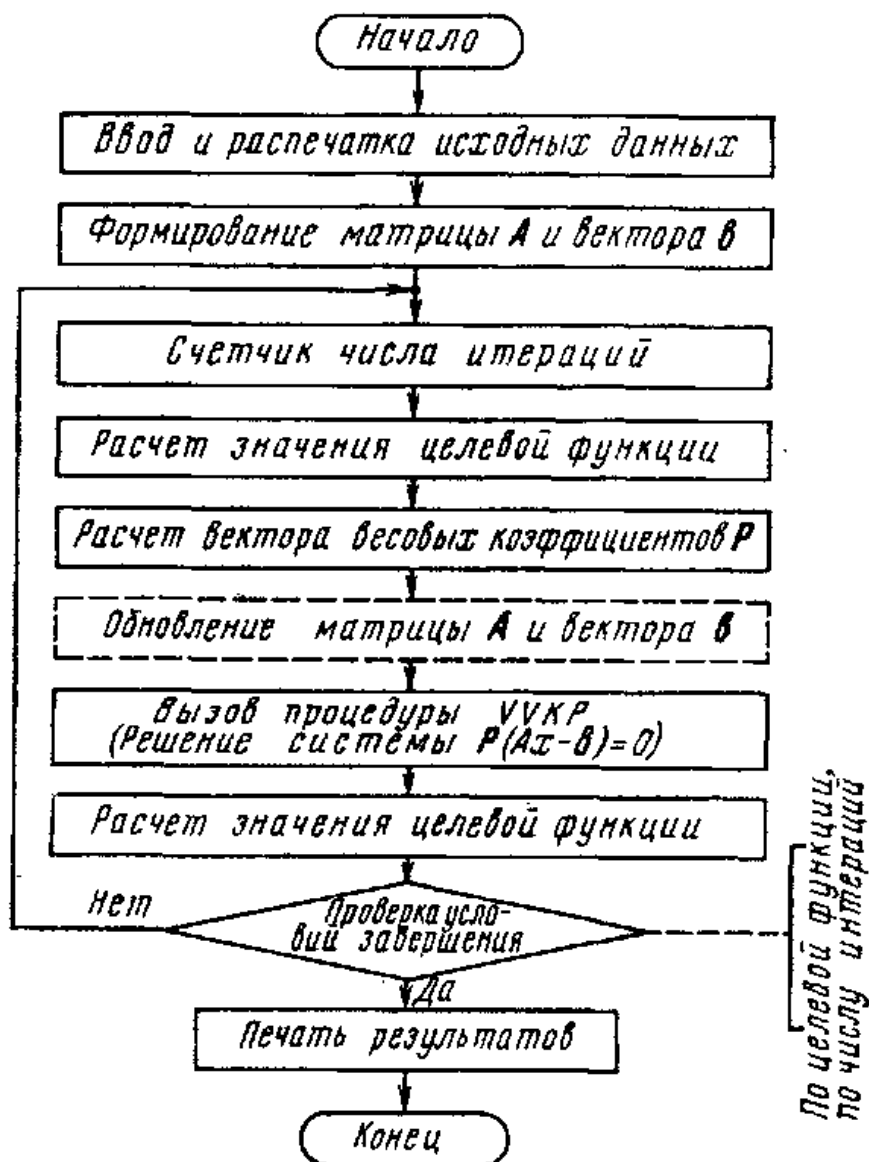


Рис. П.1

выделен в программах соответствующими комментариями. С целью согласования описателей типа данных с подпрограммами ПНП ПЛ/1 массивы A и b объявлены BIN FLOAT, а переменные размерности массивов — BIN FIXED.

Так как размерность задач непостоянна, то часть переменных массивов объявлена CONTROLLED. В ходе вычислений массивы A и b, как правило, остаются постоянными, а все изменения в итерационный процесс вносятся пересчетом вектора весовых коэффициентов P. Однако могут встретиться алгоритмы, в которых матрица A и вектор b обновляются от итерации к итерации. Именно так обстоит дело в алгоритме отыскания точки на минимальном расстоянии от совокупности отрезков (программа HITRAS). Этому случаю соответствует показанный пунктиром на схеме этап обновления массивов A и b.

**Вызов подпрограммы VVKP.** Данная подпрограмма реализует метод взвешенных наименьших квадратов, причем ее основу составляет стандартная подпрограмма MLSQ. Отказ от применения MLSQ в чистом виде обусловлен, главным образом, необходимостью сохранения исходных массивов **A** и **B** в течение всего итерационного процесса. Стандартная подпрограмма MLSQ не позволяет сохранять массивы **A** и **B**, так как на их место засылается информация, получаемая в ходе вычислений. Вызов подпрограммы VVKP производится оператором

CALL VVKP (A, B, Z, P, N, M, OD);

где **A(N, M)** — матрица;

**B(N, OD)** — вектор-столбец ( $OD = 1$ );

**P(N)** — вектор-столбец весовых коэффициентов;

**Z(M)** — вектор-столбец решения.

**Проверка условий завершения итерационного процесса.** Во всех программах итерационный процесс завершается, как только на двух следующих друг за другом шагах изменение абсолютного значения целевой функции становится меньше некоторого наперед заданного числа, которое также является мерой точности нахождения экстремума. Кроме этого, предусмотрена защита от заикливания путем ограничения числа итераций **IT**, которое задается пользователем.

**Печать результатов.** Для вывода результатов итерационного процесса в табличной форме используется подпрограмма TABL. Обращение к этой подпрограмме осуществляется оператором

CALL TABL (N, TX, TF, IT);

где **TX(IT, M)** — совокупность векторов искомым неизвестных на **IT** шагах итерационного процесса;

**TF(IT)** — массив значений целевой функции на **IT** шагах итерационного процесса.

Подпрограмма TABL рассчитана для случаев, когда размерность вектора **x** меньше или равна 10. В остальных случаях нужно внести в подпрограмму соответствующие изменения.

**1. Программа LIDER.** Программа используется для решения задачи одномерного шкалирования при совместной обработке разнородной экспертной информации: полных и неполных матриц парных сравнений и ранжировок.

Входные параметры:

**NB** — число матриц парных сравнений;

**NR** — число ранжировок;

**M** — число объектов в матрице (ранжировке);

**R(NR, M)** — массив ранжировок;

**TRR1(NB, M, M)** — массив матриц парных сравнений.

Выходные параметры:

**H** — номер итерации;

**X(M)** — вектор значений искомой шкалы;

**F** — значение целевой функции на **H**-м шаге итерационного процесса.

Пример подготовки входной информации для программы LIDER представлен в табл. П.1. Входные параметры следующие:  $NB = 1$ ,  $NR = 3$ ,  $M = 4$ ; ранжирования: 4321, 4231 и 4132; матрица имеет вид

|     |   |     |     |
|-----|---|-----|-----|
| ×   |   | 0,5 | 1   |
|     | × | 1   | 0   |
| 0,5 | 0 | ×   | 0,5 |
| 0   | 1 | 0,5 | ×   |

Т а б л и ц а П.1

| Номер<br>перфокарты | Номер позиции на перфокарте |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|---------------------|-----------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
|                     | 1                           | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1                   | 1                           |   | 3 |   | 4 |   |   |   |   |    |    |    |    |    |    |
| 2                   | 4                           |   | 3 |   | 2 |   | 1 |   | 4 |    | 2  |    | 3  |    | 1  |
| 3                   | 4                           |   | 1 |   | 3 |   | 2 |   |   |    |    |    |    |    |    |
| 4                   | 1                           |   | — | 5 |   | 0 |   | 5 |   | 1  |    |    |    |    |    |
| 5                   | —                           | 5 |   | 1 |   | 1 |   | 0 |   |    |    |    |    |    |    |
| 6                   | 0                           |   | 5 |   | 0 |   | 1 |   | 0 |    | 5  |    |    |    |    |
| 7                   | 0                           |   | 1 |   | 0 |   | 5 |   | 1 |    |    |    |    |    |    |

В п. 2 (с. 129) приведен текст программы (с картами задания) для случая, когда подпрограммы VVKP, PTMATR, TIMES, TABLE, RAZMER используются в виде исходных модулей, а подпрограмма MLSQ в виде загрузочного модуля.

**2. Программа STEINER.** Программа предназначена для решения задачи Штейнера — нахождения точки на минимальном расстоянии от совокупности точек в евклидовой метрике.

Входные параметры:

NI — число исходных точек;

N — размерность пространства;

KRD (NI, N) — массив координат исходных точек;

W (NI) — вектор весовых коэффициентов.

Выходные параметры:

H — номер итерации;

X (N) — координаты искомой точки на H-й итерации;

NEV — суммарное расстояние от искомой точки до заданных на H-й итерации.

Пример подготовки входной информации для программы STEINER представлен в табл. П.2. Входные параметры соответствуют примеру, результаты решения которого приведены в гл. 3 (табл. 3.5 и 3.6).

Т а б л и ц а П.2

| Номер<br>перфокарты | Номер позиции на перфокарте |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|---------------------|-----------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
|                     | 1                           | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1                   | 5                           |   | 2 |   |   |   |   |   |   |    |    |    |    |    |    |
| 2                   | 1                           |   | 2 |   | 3 |   | 1 |   | 6 |    | 1  |    | 7  |    | 0  |
| 3                   | 5                           | 6 |   | 5 | 0 |   |   |   |   |    |    |    |    |    |    |

**3. Программа HITRAS.** Программа предназначена для нахождения точки, минимально удаленной от совокупности отрезков.

Входные параметры:

N — число отрезков;

KRD (N, 4) — массив координат отрезков размерностью  $N \times 4$ .

Выходные данные:

H — номер итерации;

X (2) — координаты искомой точки на H-й итерации;

NEV — суммарное расстояние на H-й итерации.

Т а б л и ц а П.3

| Номер перфокарты | Номер позиции на перфокарте |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|------------------|-----------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
|                  | 1                           | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1                | 5                           |   | 0 |   | 0 |   | 0 |   | 1 |    |    |    |    |    |    |
| 2                | 1                           |   | 1 |   | 1 |   | 2 |   | 2 |    | 2  |    | 2  |    | 3  |
| 3                | 1                           |   | 0 |   | 2 |   | 0 |   | 2 |    | 1  |    | 3  |    | 1  |

Пример подготовки входной информации для программы HITRAS представлен в табл. П.3. Входные параметры соответствуют примеру, результаты решения которого приведены в гл. 3 (табл. 3.10, 3.11).

**4. Программа MINMAX.** Программа предназначена для решения задачи размещения точечного объекта, когда требуется минимизировать максимальное евклидово расстояние от размещаемой точки до любой из заданных (см. § 3.2).

Входные параметры:

NOB — число заданных точек;

N — размерность пространства;

IT — предельное число итераций;

EPS — константа останова итерационного процесса;

A (NOB, N) — координаты заданных точек.

Выходные параметры:

ST — показатель степени вспомогательной функции  $f_m = \sum_{i=1}^n r_i^m$ ;

X (N) — координаты искомой точки;

R (NOB) — вектор расстояний от искомой точки до каждой из заданных.

При больших степенях возможно возникновение ситуации «переполнение». В этом случае необходимо изменить масштаб входных данных.

Пример подготовки входной информации для программы MINMAX представлен в табл. П.4. Входные параметры соответствуют примеру 3.9.

Для решения задач условной оптимизации нами использовался алгоритм комплексного мажорирования, работу которого удобно показать при помощи схемы, представленной на рис. П.2. В зависимости от особенностей решаемой задачи эта схема может уточняться и детализироваться, но этапы отыскания точки в допустимой области и построения мажорирующего эллипсоида необходимы в любом случае. Поэтому ниже приводится описание и тексты подпрограмм POISK и

Т а б л и ц а П.4

| Номер перфокарты | Номер позиции на перфокарте |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|------------------|-----------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
|                  | 1                           | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1                | 5                           |   | 3 |   | 1 | 0 | 0 | 0 |   |    |    |    |    |    | 0  |
| 2                | 0                           | . | 0 | 0 | 0 | 0 | 0 | 0 | 1 |    | 1  |    | 1  |    |    |
| 3                |                             | 0 |   | 0 | . | 9 | 5 |   | 1 |    | 0  | .  | 1  |    |    |
| 4                | 0                           | . | 1 |   | 0 | . | 1 |   | 0 |    | 4  |    |    |    |    |
| 5                | 0                           | . | 3 | 4 | 5 |   | 0 | . | 2 | 2  | 5  |    |    |    |    |
| 6                | 3                           |   | 4 |   | 5 |   |   |   |   |    |    |    |    |    |    |

ELLIPS, которые соответствуют этим этапам. Кроме того, в качестве примера программы, реализующей алгоритм комплексного мажорирования и использующей подпрограммы POISK и ELLIPS, дается описание программы АКМ для решения задачи Штейнера при наличии линейных ограничений в форме неравенств.

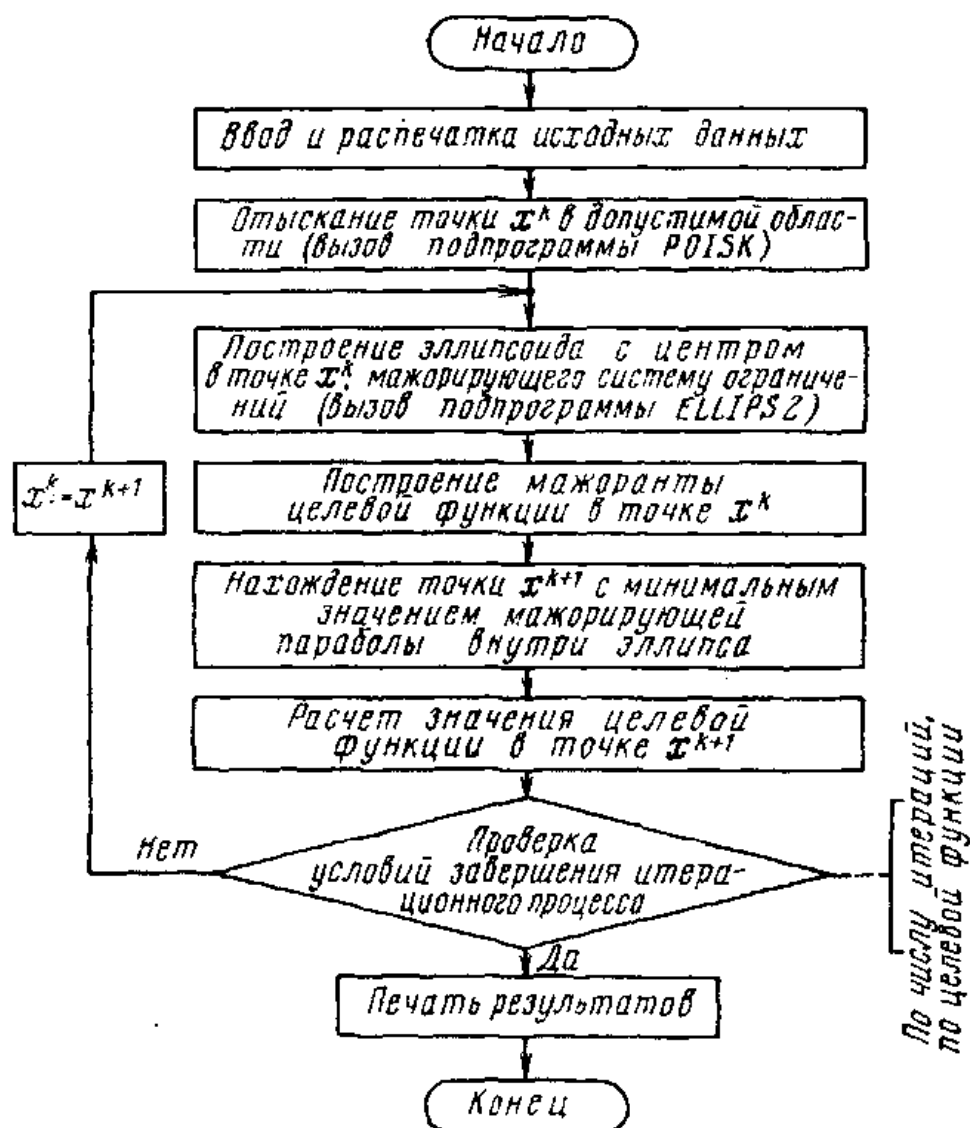


Рис. П.2

5. Подпрограмма POISK. Подпрограмма предназначена для нахождения точки, удовлетворяющей системе линейных неравенств.

Входные параметры:

$N$  — размерность пространства;

$L$  — число ограничений неравенств;

$KO(L, N + 1)$  — матрица коэффициентов ограничений в форме  $Ax + b \leq 0$ , элементы которой равны

$$KO(L, N + 1) = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} & b_1 \\ a_{21} & a_{22} & \dots & a_{2N} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{L1} & a_{L2} & \dots & a_{LN} & b_L \end{bmatrix};$$

$X(N)$  — вектор координат точки, из которой начинается поиск.

Выходные параметры:

$X(N)$  — вектор координат точки, удовлетворяющей системе линейных неравенств.

Обращение к подпрограмме осуществляется оператором CALL POISK (L, N, KO, X). Входной вектор координат исходной точки  $X(N)$ , в ходе вычислений заменяется вектором координат точки  $X(N)$ , удовлетворяющей ограничениям. Параметры N, L и KO (L, N + 1) во время выполнения подпрограммы не изменяются и возвращаются в вызывающую процедуру. Подпрограмма POISK используется в программе АКМ, предназначенной для решения задач размещения с линейными ограничениями.

В том случае, когда система ограничений несовместна и, естественно, не существует точки ей удовлетворяющей, подпрограмма выдает сообщение «Внутренняя точка не найдена: нарушено ограничение I», где I — номер ограничения.

**6. Подпрограмма ELLIPS.** Подпрограмма предназначена для построения эллипсоида с центром в точке  $x^k$ , мажорирующего систему линейных неравенств.

Входные параметры:

N — размерность пространства;

$X(N)$  — координаты центра эллипсоида  $x^k$ ;

OGR (L, N + 1) — матрица коэффициентов ограничений в форме  $Ax + b \leq 0$ , где

$$\text{OGR}(L, N+1) = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} & b_1 \\ a_{21} & a_{22} & \dots & a_{2N} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{L1} & a_{L2} & \dots & a_{LN} & b_L \end{bmatrix};$$

L — число ограничений;

Z — параметр, определяющий размер эллипсоида: если  $Z = 1$  — эллипс увеличенных размеров (он может и не быть мажорирующим); если  $Z = N$ , то эллипс гарантированно мажорирующий (см. § 5.2).

Выходные параметры:

Q (N, N) — матрица, определяющая размеры и ориентацию осей мажорирующего эллипсоида  $(x - x^k)^T Q (x - x^k) = 1$ .

Подпрограмма в свою очередь обращается к подпрограммам MINV (расчет определителя), MMGG (умножение матриц), MFG и MDLG (решение системы линейных уравнений) из пакета ПНП ПЛ/1 [8]. Кроме того, используются подпрограммы ORTGRI (ортогонализация Грама—Шмидта), SORT (упорядочение массива по величине от большего к меньшему), TRANSP (транспонирование матриц), RANDOM (датчик случайных чисел), текст которых приводится ниже в п.2.

**7. Программа АКМ.** Программа предназначена для решения задачи Штейнера — нахождения точки на минимальном суммарном расстоянии от совокупности заданных точек в евклидовой метрике при наличии линейных ограничений в форме неравенств.

Входные параметры:

N1 — число точек;

N — размерность пространства;

KRD (N1, N) — массив координат точек;

W (N1) — вектор «важности» (весомости) точек;

L — число ограничений;

OGR (L, N + 1) — матрица коэффициентов ограничений в форме,  $Ax + b \leq 0$ ;

IT — предельное число итераций;

EPS — константа останова процесса.

Выходные параметры:

$X(N)$  — координаты искомой точки на N-й итерации;

NEVF — суммарное расстояние на N-й итерации;

N — номер итерации.

Программа оформлена в виде главной процедуры. Кроме подпрограмм POISK и ELLIPS2 вызываются подпрограммы TPR2 (для приближенного нахождения точки внутри эллипсоида с минимальным значением мажоранты целевой функции), UMV (умножение матрицы на вектор), TABL и TIMES.

Пример подготовки входной информации для программы АКМ представлен в табл. П.5. Входные параметры соответствуют примеру 5.4.

Т а б л и ц а П.5

| Номер<br>перфокарты | Номер позиции на перфокарте |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|---------------------|-----------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
|                     | 1                           | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1                   | 7                           |   | 2 |   | 4 | 0 |   | 0 | . | 0  | 0  | 1  |    | 4  |    |
| 2                   | —                           | 1 |   | 1 |   | — | 1 |   | — | 0  | .  | 5  |    | 1  | 0  |
| 3                   |                             | — | 1 |   | — | 1 |   | 5 |   | 1  |    | 4  |    | —  | 9  |
| 4                   |                             | 1 | 2 |   | — | 5 |   | 9 |   | 1  |    | 1  |    | 1  |    |
| 5                   | 1                           |   | 1 |   | 1 |   | 1 |   | 1 | 1  |    |    | —  | 1  | 0  |
| 6                   |                             | — | 1 |   | — | 1 |   | 4 |   | —  | 1  |    | 0  |    | 0  |
| 7                   |                             | 0 |   | — | 1 |   | 0 |   | 1 | 5  |    | 1  | 0  | 0  |    |

Все программы, приведенные в п. 2 приложения, отлажены на ЭВМ ЕС-1045. Отладку любой из программ следует начинать, используя рассмотренные в тексте числовые примеры. Это, безусловно, сэкономит время и усилия потенциальных пользователей.

В заключение отметим, что в данном приложении приведены минимально необходимые сведения о программной реализации метода ВВКМ. Всестороннее изложение вопросов, связанных с практическим решением задач оптимизации на ЭВМ, содержится в [25].

## 2. Тексты программ

# Текст программы LIDER

```
//LIDER JOB (
// MSGLEVEL=(0,0)
// EXEC PL1LFCLG,PARM='S'
//PL1L SYSIN DD *
    LIDER:PROC OPTIONS(MAIN),
/*****
/* ПРОГРАММА LIDER ПРЕДНАЗНАЧЕНА ДЛЯ РЕШЕНИЯ ЗАДАЧИ
/* ОДНОМЕРНОГО ШИФРОВАНИЯ
/* ВХОДНЫЕ ПАРАМЕТРЫ
/* NB -ЧИСЛО МАТРИЦ ПАРНЫХ СРАВНЕНИЙ,
/* NR -ЧИСЛО РАНЖИРОВОК,
/* M -ЧИСЛО ОБЪЕКТОВ В МАТРИЦЕ(РАНЖИРОВКЕ),
/* R(NR,M) -МАССИВ РАНЖИРОВОК,
/* TRR1(NB,M,M) -МАССИВ МАТРИЦ ПАРНЫХ СРАВНЕНИЙ,
/* IT -ПРЕДЕЛЬНОЕ ЧИСЛО ИТЕРАЦИЙ
/* ВЫХОДНЫЕ ПАРАМЕТРЫ:
/* X(M) -ВЕКТОР ЗНАЧЕНИЙ ИСХОДНОЙ ШИФРЫ НА N-ЕЙ ИТЕРАЦИИ,
/* F -ЗНАЧЕНИЕ ЦЕЛЕВОЙ ФУНКЦИИ(НЕВЯЗКА) НА N-ЕЙ ИТЕРАЦИИ,
/* N -НОМЕР ИТЕРАЦИИ
/* ИСПОЛЗУЮТСЯ ПОДПРОГРАММЫ VVVF,PTMATR,TIMES,TABL,RAZMER,
/* ИЗ ПАКЕТА ОНП ПЛ-1 ПОДПРОГРАММА ML5Q
/*****
DCL (F1,NNEV,BETA,F,FA,EPS,EPL,DD
(I,I1,M,N,MM,MB1,OD,K,H,NHN
((X,XXX)(M),A(MM,MB1),(IND,P,Y,Y1)(MM),TRR(N,M,M),TF(IT),
TRR1(NB,M,M),(BS,B)(MM,OD),BB(M,M),TX(IT,M),XD(M-1),
R(NR,M)
) BIN FLOAT,
) BIN FIXED,
) BIN FLOAT CTL,
NAP FORMAT(SKIP(2),COL(10),A,X(1),F(3)),
RAZM FORMAT(SKIP,A,A,A);
/* ВВОД И РАСПЕЧАТКА ИСХОДНЫХ ДАННЫХ
DAN ON ENDFILE(SYSIN) GO TO KON;
CALL TIMES;
GET LIST(NB,NR,M,IT)COPY;
N=NB+NR;
ALLOCATE BB,R,XXX,TRR;
PUT EDIT((40)'*', 'ИСХОДН ДАННЫЕ', (40)'*')(R(RAZM));
PUT EDIT('ЧИСЛО РАНЖИРОВОК...',NR)(R(NAP));
PUT EDIT ('ЧИСЛО МАТРИЦ...',NB)(R(NAP));
PUT EDIT ('ЧИСЛО ОБЪЕКТОВ В РАНЖИРОВКЕ(МАТРИЦЕ)...',M)(R(NAP));
IF NR>0 THEN GET LIST(R),
ELSE GO TO BINO;
PUT EDIT ('РАНЖИРОВКИ')(SKIP(5),COL(10),A);
DO I=1 TO NR,
XXX=R(I,*);
CALL SRATUR(XXX,BB,M);
TRR(I,*,*)=BB(*,*);
PUT EDIT ('РАНЖИРОВКА...',I)(R(NAP));
PUT EDIT((XXX(I2) DO I2= 1 TO M))(SKIP,COL(10),(M)F(3));
CALL PTMATR (BB,'МАТРИЦА ДЛЯ ДАННОЙ РАНЖИРОВКИ');
END;
BINO IF NB>0 THEN DO,
ALLOCATE TRR1;
GET LIST (TRR1);
END;
PUT EDIT('МАТРИЦЫ БИНАРНЫХ ОТНОШЕНИЙ')(SKIP(5),COL(10),A);
DO J=1 TO NB,
BB(*,*)=TRR1(J,*,*);
TRR(NR+J,*,*)=TRR1(J,*,*);
CALL PTMATR (BB,'МАТРИЦА');
END;
FREE BB,TRR1;
PUT EDIT((40)'*', 'РЕЗУЛЬТАТЫ РЕШЕНИЯ', (40)'*')(R(RAZM));
CALL RAZMER(TRR,M,MM,N);
MB1=M-1;I1=0,OD=1;ALLOCATE A,B,BS,IND;A=0;NHN=0;
ФОРМИРОВАНИЕ МАТРИЦА А И НАЧАЛЬНОЙ МАТРИЦЫ В
DO JA=1 TO N;
DO I=1 TO M;
```

|                                                       |            |
|-------------------------------------------------------|------------|
| DO K=1+1 TO M.                                        | LDR 0068   |
| I1=I1+1.                                              | LDR 0069   |
| IF TRR(JA,I,K)=1 THEN DO.                             | LDR 0070   |
| IND(I1)=1.                                            | LDR 0071   |
| IF I=1 THEN DO.                                       | LDR 0072   |
| A(I1,K-1),B(I1,1)=1.                                  | LDR 0073   |
| END.                                                  | LDR 0074   |
| ELSE DO.                                              | LDR 0075   |
| A(I1,K-1),B(I1,1)=1.                                  | LDR 0076   |
| A(I1,I-1)=-1.                                         | LDR 0077   |
| END.                                                  | LDR 0078   |
| END.                                                  | LDR 0079   |
| IF TRR(JA,I,K)=0 5 THEN DO.                           | LDR 0080   |
| IND(I1),IND(I1+1)=0.                                  | LDR 0081   |
| NHN=NHN+1.                                            | LDR 0082   |
| IF I=1 THEN DO.                                       | LDR 0083   |
| A(I1,K-1),B(I1,1)=-1.                                 | LDR 0084   |
| I1=I1+1.                                              | LDR 0085   |
| A(I1,K-1)=1, B(I1,1)=-1.                              | LDR 0086   |
| END.                                                  | LDR 0087   |
| ELSE DO.                                              | LDR 0088   |
| A(I1,I-1)=1.                                          | LDR 0089   |
| A(I1,K-1),B(I1,1)=-1.                                 | LDR 0090   |
| I1=I1+1.                                              | LDR 0091   |
| A(I1,K-1)=1.                                          | LDR 0092   |
| A(I1,I-1),B(I1,1)=-1.                                 | LDR 0093   |
| END.                                                  | LDR 0094   |
| END.                                                  |            |
| IF TRR(JA,I,K)=0 THEN DO.                             | LDR 0095   |
| IND(I1)=1.                                            | LDR 0096   |
| IF I=1 THEN DO.                                       | LDR 0097   |
| A(I1,K-1)=-1.                                         | LDR 0098   |
| B(I1,1)=1.                                            | LDR 0099   |
| END.                                                  | LDR 0100   |
| ELSE DO.                                              | LDR 0101   |
| A(I1,I-1),B(I1,1)=1.                                  | LDR 0102   |
| A(I1,K-1)=-1.                                         | LDR 0103   |
| END.                                                  | LDR 0104   |
| END.                                                  | LDR 0105   |
| IF TRR(JA,I,K)=-5 THEN I1=I1-1.                       | LDR 0106   |
| END.                                                  | LDR 0107   |
| END.                                                  | LDR 0108   |
| END.                                                  | LDR 0109   |
| ALLOCATE X,Y,Y1,TX,P,XD,TF.                           | LDR 0110   |
| P=0, X=0, H=0, XD=0; F1=10**10.                       | LDR 0111   |
| ЦИКЛ H=H+1; /* СЧЕТЧИК ЧИСЛА ИТЕРАЦИЙ                 | */LDR 0112 |
| Y1=0; Y=0, F=0; EPS=1, EPL=1E-10.                     | LDR 0113   |
| DO J=1 TO MM.                                         | LDR 0114   |
| DO I=1 TO MB1.                                        | LDR 0115   |
| Y1(J)=Y1(J)+A(J,I)*XD(I).                             | LDR 0116   |
| X(I+1)=XD(I).                                         | LDR 0117   |
| END.                                                  | LDR 0118   |
| Y(J)=Y1(J)+B(J,1).                                    | LDR 0119   |
| END.                                                  | LDR 0120   |
| DO J=1 TO MM.                                         | LDR 0121   |
| F=F+0.5*(ABS(Y(J))+Y(J)).                             | LDR 0122   |
| P(J)=0.25/(ABS(Y(J))+EPL).                            | LDR 0123   |
| BS(J,1)=-B(J,1)-ABS(Y(J)).                            | LDR 0124   |
| END.                                                  | LDR 0125   |
| TF(H)=F/TX(H.*)=X.                                    | LDR 0126   |
| /* ВЫЗОВ ПРОЦЕДУРЫ VVKP                               | */LDR 0128 |
| CALL VVKP(A,BS,XD,P,MM,MB1,0D).                       | LDR 0129   |
| /* ПРОВЕРКА УСЛОВИЯ ЗАВЕРШЕНИЯ ИТЕРАЦИОННОГО ПРОЦЕССА | */LDR 0130 |
| IF ABS(F-F1) > 0.01*H THEN DO.                        | LDR 0131   |
| F1=F.                                                 | LDR 0132   |
| GO TO CIKL.                                           | LDR 0133   |
| END.                                                  | LDR 0134   |
| ELSE GO TO OUTPUT.                                    | LDR 0135   |
|                                                       | LDR 0136   |
| OUTPUT.                                               |            |

```

/* ПЕЧАТЬ РЕЗУЛЬТАТОВ РАСЧЕТА
CALL TABL(M,TX,TF,H);
BETA=F/(MM-NHN);
PUT EDIT('BETA=',BETA)(SKIP,COL(10),A,X(1),F(7,4));
PUT EDIT((120)('*'))(SKIP,A);
/*****
/* РЕШЕНИЕ С BETA- ВТОРОЙ ЭТАП */
P=0; X=0; H=0; XD=0; F1=10**10;
CIK2 H=H+1; /* СЧЕТЧИК ЧИСЛА ИТЕРАЦИЙ
Y1=0; Y=0; F=0;
IF BETA=0 THEN BETA=0.001;
DO J=1 TO MM;
DO I=1 TO MB1;
Y1(J)=Y1(J)+A(J,I)*XD(I);
X(I+1)=XD(I);
END;
Y(J)=Y1(J)+B(J,1);
END;
DO J=1 TO MM;
F=F+((ABS(Y(J))+Y(J))/2);
IF Y(J)<0 THEN F=F-IND(J)*BETA*LOG(2-EXP(Y(J)/BETA));
IF IND(J)=0 & -EPS<=Y(J) & Y(J)<=0 THEN
F=F-BETA*LOG((1-EXP(-EPS/BETA)*COSH((Y(J)+EPS)/BETA))/
(EXP(-EPS/BETA)*SINH(EPS/BETA)));
IF IND(J)=0 & Y(J)<-EPS THEN
F=F-BETA*LOG(2*(1-EXP(-EPS/BETA))/(1-EXP(-2*EPS/BETA)));
US1 IF IND(J)=1 & Y(J)<= 0 THEN DO,
P(J)=1/BETA;
BS(J,1)=B(J,1)-Y(J)+BETA*0.5*EXP(Y(J)/BETA)/
(2-EXP(Y(J)/BETA));
END;
US2 IF IND(J)=1 & Y(J)>0 THEN DO,
P(J)=0.25/Y(J);
BS(J,1)=B(J,1)+Y(J);
END;
US3 IF IND(J)=0 & Y(J)< -EPS THEN DO;
DD=-BETA*LOG(2*(1-EXP(-EPS/BETA))/(1-EXP(-2*EPS/BETA)));
P(J)=0.25/ABS(Y(J)-DD);
BS(J,1)=B(J,1)+ABS(Y(J));
END;
US4 IF IND(J)=0 & Y(J)>0 THEN DO,
P(J)=0.25/Y(J);
BS(J,1)=B(J,1)+Y(J);
END;
US5 IF IND(J)=0 & -EPS<=Y(J) & Y(J)<=0 THEN DO;
P(J)=1/(BETA*(1-EXP(-2*EPS/BETA)));
BS(J,1)=B(J,1)+ABS(Y(J))+(1/P(J))*SINH((Y(J)+EPS)/BETA)*0.5/
(EXP(EPS/BETA)-COSH((Y(J)+EPS)/BETA));
END;
BS=-BS; TF(H)=F; TX(H,*)=X;
/* ВЫЗОВ ПРОЦЕДУРЫ VVKP
CALL VVKP(A,BS,XD,P,MM,MB1,DD);
/* ПРОВЕРКА УСЛОВИЙ ЗАВЕРШЕНИЯ ИТЕРАЦИОННОГО ПРОЦЕССА
IF ABS(F-F1)>1E-3 & H<IT THEN DO,
F1=F;
GO TO CIK2;
END;
ELSE GO TO*OUTFIN;
/* ПЕЧАТЬ РЕЗУЛЬТАТОВ РАСЧЕТА
OUTFIN
CALL TABL(M,TX,TF,H);CALL TIMES;PUT EDIT((120)('*'))(SKIP,A);
GO TO DAN;
/*****
SR#TUR PROC(X,TRR,M);
/* ПОДПРОГРАММА SR#TUR ПРЕОБРАЗУЕТ СТРОГУЮ РАМКУ В
В ТУРНИРНУЮ ТАБЛИЦУ
DCL (TRR(*),X(*)) BIN FLOAT.(I,J,M) BIN FIXED;
DO I=1 TO M;
J=X(M+1-I);

```

```

*/LDR 0137
LDR 0138
LDR 0139
LDR 0140
LDR 0141
LDR 0142
LDR 0143
LDR 0144
*/LDR 0145
LDR 0146
LDR 0147
LDR 0148
LDR 0149
LDR 0150
LDR 0151
LDR 0152
LDR 0153
LDR 0154
LDR 0155
LDR 0156
LDR 0157
LDR 0158
LDR 0159
LDR 0160
LDR 0161
LDR 0162
LDR 0163
LDR 0164
LDR 0165
LDR 0166
LDR 0167
LDR 0168
LDR 0169
LDR 0170
LDR 0171
LDR 0172
LDR 0173
LDR 0174
LDR 0175
LDR 0176
LDR 0177
LDR 0178
LDR 0179
LDR 0180
LDR 0181
LDR 0182
LDR 0183
LDR 0184
LDR 0185
LDR 0186
LDR 0187
*/LDR 0190
LDR 0191
*/LDR 0192
LDR 0193
LDR 0194
LDR 0195
LDR 0196
LDR 0197
*/LDR 0198
LDR 0199
LDR 0200
LDR 0203
*/LDR 0204
LDR 0205
LDR 0206
*/LDR 0207
LDR 0208
LDR 0209
LDR 0210

```

```

      TRR(*,J)=0;
      TRR(J,*)=1;
    END,
  END SR#TUR,
/*****
  VVKP PROC(A,B,X,P,M,N,OD),
  DCL ((A,B)(*,*), (X,P)(*)) BIN FLOAT,
  (M,N,OD,I) BIN FIXED,
  (A1(M,N),B1(M,OD))BIN FLOAT CTL,
  ALLOCATE A1,B1,
  DO I=1 TO M,
    A1(I,*)= A(I,*)*SQRT(P(I)),
    B1(I,1)= B(I,1)*SQRT(P(I)),
  END,
  CALL MLSQ(A1,B1,M,N,OD),
  DO I=1 TO N,
    X(I)=B1(I,1),
  END,
  FREE A1,B1,
  END VVKP;
/*****
  TIMES:PROC,
  DCL A CHAR(9),B CHAR(120),
  A=TIME,
  B=(92)' 'SUBSTR(A,1,2)' ' YAC 'SUBSTR(A,3,2)
    'MMH 'SUBSTR(A,5,2)' ' '
    SUBSTR(A,7)' ' CEK',
  PUT SKIP LIST(B),
  END TIMES,
/*****
  PTMATR:PROC(MATR,SHAPKA),
  DCL MATR(*,*) BIN FLOAT,SHAPKA CHAR(*),I,J,NMATR,MMATR,
  NMATR=DIM(MATR,2), MMATR=DIM(MATR,1),
  PUT EDIT (SHAPKA) (SKIP,A),
  PUT EDIT ((MATR(I,J) DO J=1 TO NMATR)
  DO I=1 TO MMATR))
  (SKIP,(NMATR)(X(2),F(7,3))),
  END PTMATR,
/*****
  TABL PROC(M,TX,TF,H),
  DCL (TF(*),TX(*,*)) BIN FLOAT,
  U CHAR(120) INIT((120)'~'),
  (I,J,M,H) BIN FIXED,
  PUT EDIT(U)(SKIP,COL(2),A(24+8*M)),
  PUT EDIT(' ','HOMEF WAGA',' ',('X',1,' ' DO I=1 TO M),
  'HEB3KA',' ')(SKIP,COL(2),A,A,A,(M)(X(2),A,F(2,0),X(2),A),
  X(1),A,X(2),A),
  PUT EDIT(U)(SKIP,COL(3),A(22+8*M)),
  DO I=1 TO H,
    PUT EDIT(' ',I,' ',(TX(I,J),' ' DO J=1 TO M),TF(1),' ')
    (SKIP,COL(2),A,X(4),F(2,0),X(4),A,(M)(F(7,3),A),F(10,3),A),
  PUT EDIT(U)(SKIP,COL(3),A(22+8*M)),
  END,
  PUT EDIT(U)(SKIP(0),COL(2),A(24+8*M)),
  END TABL,
/*****
  RAZMER:PROC(TRR,M,MM,N),
  DCL TRR(*,*,*) BIN FLOAT,(I,K,M,MM,JA) BIN FIXED, MM=0,
  DO JA=1 TO N;
    DO I=1 TO M,
      DO K=I+1 TO M,
        IF TRR(JA,I,K)=1 | TRR(JA,I,K)=0 THEN MM=MM+1,
        IF TRR(JA,I,K)= 5 THEN MM=MM+2,
      END,
    END,
  END,
  END RAZMER,
/*****
KON
  END LIDER,

```

```

LDR 0211
LDR 0212
LDR 0213
LDR 0214
LDR 0215
LDR 0216
LDR 0217
LDR 0218
LDR 0219
LDR 0220
LDR 0221
LDR 0222
LDR 0223
LDR 0224
LDR 0225
LDR 0226
LDR 0227
LDR 0228
LDR 0229
LDR 0230
LDR 0231
LDR 0232
LDR 0233
LDR 0234
LDR 0235
LDR 0236
LDR 0237
LDR 0238
LDR 0239
LDR 0240
LDR 0241
LDR 0242
LDR 0243
LDR 0244
LDR 0245
LDR 0246
LDR 0247
LDR 0248
LDR 0249
LDR 0250
LDR 0251
LDR 0252
LDR 0253
LDR 0254
LDR 0255
LDR 0256
LDR 0257
LDR 0258
LDR 0259
LDR 0260
LDR 0261
LDR 0262
LDR 0263
LDR 0264
LDR 0265
LDR 0266
LDR 0267
LDR 0268
LDR 0269
LDR 0270
LDR 0271
LDR 0272
LDR 0273
LDR 0274
LDR 0275
LDR 0276
LDR 0277
LDR 0310
LDR 0311

```

```

//LKED SYSLIB DD DSN=SYS1 PL1LIB,DISP=SHR
// DD DSN=PNPPL LOAD,DISP=SHR
//GO SYSPRINT DD SYSOUT=A,OUTLIM=1000
//GO SYSLIB DD *
1 3 4 100
4 3 2 1 4 2 3 1 4 1 3 2 1 -5 0 5 1 -5 1 1 0 0 5 0 1 0 5 0 1 0 5 1
//

```

## Текст программы STEINER

```

STEINER PROC OPTIONS(MAIN),
/*****/
/* ПРОГРАММА STEINER ПРЕДНАЗНАЧЕНА ДЛЯ РЕШЕНИЯ ЗАДАЧИ ШТЕЙНЕРА-
/* НАХОЖДЕНИЯ ТОЧКИ НА МИНИМАЛЬНОМ СУММАРНОМ РАССТОЯНИИ ОТ СОВО-
/* КУПНОСТИ ТОЧЕК В ЕВКЛИДОВОЙ МЕТРИКЕ.
/* ВХОДНЫЕ ПАРАМЕТРЫ
/* N1 -ЧИСЛО ИСХОДНЫХ ТОЧЕК,
/* N -РАЗМЕРНОСТЬ ПРОСТРАНСТВА,
/* KRD(N1,N)-МАССИВ КООРДИНАТ ИСХОДНЫХ ТОЧЕК,
/* W(N1) -ВЕКТОР ВЕСОВЫХ КОЭФФИЦИЕНТОВ,
/* IT -ОГРАНИЧЕНИЕ ПО ЧИСЛУ ИТЕРАЦИЙ,
/* EPS -КОНСТАНТА ОСТАНОВКИ ПРОЦЕССА.
/* ВЫХОДНЫЕ ПАРАМЕТРЫ.
/* X(N) -КООРДИНАТЫ ИСХОДНОЙ ТОЧКИ НА N-ЕЙ ИТЕРАЦИИ,
/* NEVF -СУММАРНОЕ РАССТОЯНИЕ НА N-ЕЙ ИТЕРАЦИИ,
/* H -НОМЕР ИТЕРАЦИИ
/* ИСПОЛЬЗУЮТСЯ ПОДПРОГРАММЫ VV&P,PTMATR,TIMES,TABL,
/* ИЗ ПАКЕТА ПНП ПЛ-1 ПОДПРОГРАММА MLSD
/*****/
DCL (KRD(N1,N),(W,NEVI,NEVB)(N1),X(N),A(M,N),B(M,OD),P(M),
TF(IT),TX(IT,X))
BIN FLOAT CTL,
(EPS,F,NEVF) BIN FLOAT,(N1,N,IT,M,OD,H,JJ) BIN FIXED,
/* ВВОД И РАСПЕЧАТКА ИСХОДНЫХ ДАННЫХ
DAN ON ENDFILE(SYSLIB) GO TO KON,
CALL TIMES,
GET LIST(N1,N,IT,EPS) COPY,
NAP FORMAT(SKIP(2),COL(10),A,F(3)),
PUT EDIT('ЧИСЛО ТОЧЕК=' ,N1)(R(NAP)),
PUT EDIT('РАЗМЕРНОСТЬ ПРОСТРАНСТВА = ',N)(R(NAP)),
M=N1*N; OD=1; H=0; F=10**10;
ALLOCATE KRD,W,X,A,B,NEVI,NEVB,P,TF,TX,
GET LIST(KRD,W,X);
CALL PTMATR(KRD,'КООРДИНАТЫ ТОЧЕК'),
PUT SKIP LIST('ВЕСА ТОЧЕК',W);
PUT SKIP LIST('КООРДИНАТЫ ИСХОДНОЙ ТОЧКИ',X);
/* ФОРМИРОВАНИЕ МАТРИЦЫ A И ВЕКТОРА B ДЛЯ СИСТЕМЫ УРАВНЕНИЯ AX=B
JJ=0; A=0; B=0;
DO I=1 TO M BY N,
DO J=1 TO N,
A(I+J-1,J)=1;
END;
END;
DO I=1 TO N1,
DO J=1 TO N;
JJ=JJ+1;
B(JJ,1)=KRD(I,J);
END;
END;
CICL H=H+1, /* СЧЕТЧИК ЧИСЛА ИТЕРАЦИИ
JJ=0, NEVI=0,
DO I=1 TO N1,
NEVB(I)=SQRT(SUM((KRD(I,*)-X(*))**2)),;
NEVI(I)=NEVB(I)*W(I);
DO J=1 TO N,
JJ=JJ+1;
P(JJ)=W(I)/(NEVB(I)+1E-5);
END;
END;
NEVF=SUM(NEVI);
IF(H)=NEVF, TX(H,*)=X,

```

```

/*      ПРОВЕРКА УСЛОВИЯ ЗАВЕРШЕНИЯ ИТЕРАЦИОННОГО ПРОЦЕССА
      IF ABS(NEVF-F)<EPS ; H=IT THEN GOTO OUT,
      F=NEVF;
/*      ВЫЗОВ ПРОЦЕДУРЫ VVKP
      GOTO CIKL;
/*      ПЕЧАТЬ РЕЗУЛЬТАТОВ РАСЧЕТА
OUT    CALL TABL(N,TX,TF,H);
      FREE KRD,W,X,A,B,NEV1,NEVB,P,TF,TX,
      CALL TIMES,
      GO TO DAN;
/*      *****
/*      ЗДЕСЬ НЕОБХОДИМО ПОМЕСТИТЬ ПОДПРОГРАММЫ VVKP,PTMATR,TIMES,
/*      TABL, ТЕКСТ КОТОРЫХ ПРИВЕДЕН В ПРОГРАММЕ LIDER
/*      *****
KON
      END STEINER,

```

## Текст программы HITRAS

```

      HITRAS PROC OPTIONS(MAIN),
/*      *****
/*      ПРОГРАММА HITRAS ПРЕДНАЗНАЧЕНА ДЛЯ НАХОЖДЕНИЯ ТОЧКИ,
/*      МИНИМАЛЬНО УДАЛЕННОЙ ОТ СОВОКУПНОСТИ ОТРЕЗКОВ
/*      ВХОДНЫЕ ПАРАМЕТРЫ
/*      N      - ЧИСЛО ОТРЕЗКОВ,
/*      X(2)   - КООРДИНАТЫ ИСХОДНОЙ ТОЧКИ,
/*      KRD(N,4) - МАССИВ КООРДИНАТ ОТРЕЗКОВ,
/*      IT     - ПРЕДЕЛЬНОЕ ЧИСЛО ИТЕРАЦИЙ
/*      ВЫХОДНЫЕ ПАРАМЕТРЫ
/*      X(2)   - КООРДИНАТЫ ИСХОДНОЙ ТОЧКИ НА H- ЕМ ШАГЕ,
/*      NEV    - СУММАРНОЕ РАССТОЯНИЕ НА H- ЕЙ ИТЕРАЦИИ,
/*      H      - НОМЕР ИТЕРАЦИИ
/*      ИСПОЛЬЗУЮТСЯ ПОДПРОГРАММЫ VVKP,PTMATR,TIMES,TABL
/*      ИЗ ПАКЕТА ЛНП ПЛ-1 MLSD
/*      *****
      DCL (RA,RB,A,ALFA,C1,C2,DL)(N) BIN FLOAT CTL,
      (AA(2*N,2),B(2*N,1),P(2*N),KRD(N,4)) BIN FLOAT CTL,
      (X(2),NEV,SNEV) BIN FLOAT,
      (H,I,N,J,OD,DVA,N2) BIN FIXED,
      (TF(IT),TX(IT,2)) BIN FLOAT CTL,
/*      ВВОД И РАСПЕЧАТКА ИСХОДНЫХ ДАННЫХ
DAN    ON ENDFILE(SYSIN) GO TO KON,
      CALL TIMES,GET LIST(N),
      H=0,OD=1,DVA=2,SNEV=0,EPS=1E-10,N2=N*2,
      ALLOCATE RA,RB,A,ALFA,C1,C2,DL,AA,B,P,KRD,
      GET LIST(KRD,X),
      PUT EDIT('ЧИСЛО ОТРЕЗКОВ=',N)(R(NAP)),
NAP    FORMAT(SKIP(2),COL(10),A,F(3)),
      CALL PTMATR(KRD,'КООРДИНАТЫ ОТРЕЗКОВ'),
CIKL   H=H+1, NEV=0, /* СЧЕТЧИК ЧИСЛА ИТЕРАЦИЙ
      DO I=1 TO N,
      RA(I)=SQRT((X(1)-KRD(I,1))**2+(X(2)-KRD(I,2))**2),
      RB(I)=SQRT((X(1)-KRD(I,3))**2+(X(2)-KRD(I,4))**2),
      A(I)=SQRT((KRD(I,1)-KRD(I,3))**2+(KRD(I,2)-KRD(I,4))**2),
      ALFA(I)=(1/(A(I)**2+EPS))*((KRD(I,1)-KRD(I,3))*(KRD(I,1)-X(1))+
      (KRD(I,2)-KRD(I,4))*(KRD(I,2)-X(2))),
      C1(I)=ALFA(I)*KRD(I,3)+(1-ALFA(I))*KRD(I,1),
      C2(I)=ALFA(I)*KRD(I,4)+(1-ALFA(I))*KRD(I,2),
      DL(I)=SQRT((X(1)-C1(I))**2+(X(2)-C2(I))**2),
      END,
      J=0,
      DO I=1 TO 2*N-1 BY 2,
      J=J+1,
/*      ФОРМИРОВАНИЕ МАТРИЦЫ AA
      AA(I,1)=1, AA(I,2)=0, AA(I+1,1)=0, AA(I+1,2)=1,
/*      ФОРМИРОВАНИЕ И КОРРЕКТИРОВКА ВЕКТОРА B
      IF ABS(RA(J)**2-RB(J)**2)<=A(J)**2 THEN DO,
      B(I,1)=C1(J),
      B(I+1,1)=C2(J),

```

```

P(I)=1/(DL(J)+EPS); HITR0051
P(I+1)=P(I); HITR0052
NEV=NEV+DL(J); HITR0053
END; HITR0054
IF ABS(RA(J)**2-RB(J)**2)>A(J)**2 & RA(J)**2>RB(J)**2 THEN DO; HITR0055
B(I,1)=KRD(J,1); HITR0056
B(I+1,1)=KRD(J,2); HITR0057
P(I)=1/(RA(J)+EPS); HITR0058
P(I+1)=P(I); HITR0059
NEV=NEV+RA(J); HITR0060
END; HITR0061
IF ABS(RA(J)**2-RB(J)**2)>A(J)**2 & RA(J)**2>RB(J)**2 THEN DO; HITR0062
B(I,1)=KRD(J,3); HITR0063
B(I+1,1)=KRD(J,4); HITR0064
P(I)=1/(RB(J)+EPS); HITR0065
P(I+1)=P(I); HITR0066
NEV=NEV+RB(J); HITR0067
END; HITR0068
END; HITR0069
TF(H)=NEV; TX(H,*)=X(*); HITR0070
/* ПРОВЕРКА УСЛОВИЙ ЗАВЕРШЕНИЯ ИТЕРАЦИОННОГО ПРОЦЕССА */ HITR0071
IF ABS(NEV-SNEV)<0.001 & H=IT THEN GO TO OUT; HITR0072
CALL VVKP(AA,B,X,P,N2,DVA,00); /* ВЫЗОВ ПРОЦЕДУРЫ VVKP */ HITR0073
SNEV=NEV; GO TO CIKL; HITR0074
/* ПЕЧАТЬ РЕЗУЛЬТАТОВ РАСЧЕТА */ HITR0075
OUT CALL TABL(DVA,TX,TF,H); CALL TIMES; HITR0076
FREE RA,RB,A,ALFA,C1,C2,DL,AA,B,P,KRD,TF,TX; HITR0077
GO TO DAN; HITR0078
/*****/ HITR0079
/* ЗДЕСЬ НЕОБХОДИМО ПОМЕСТИТЬ ПОДПРОГРАММЫ VVKP,PTMATR,TIMES, */ HITR0080
/* TABL, ТЕКСТ КОТОРЫХ ПРИВЕДЕН В ПРОГРАММЕ LIDER */ HITR0081
/*****/ HITR0082
KON; HITR0083
END HITRAS; HITR0084

```

## Текст программы MINMAX

```

MINMAX PROC OPTIONS(MAIN); MIMA0001
/*****/ MIMA0002
/* ПРОГРАММА MINMAX ПРЕДНАЗНАЧЕНА ДЛЯ РЕШЕНИЯ МИНИМАКСНОЙ ЗАДАЧИ */ MIMA0003
/* РАЗМЕЩЕНИЯ ТОЧЕЧНОГО ОБЪЕКТА */ MIMA0004
/* ВХОДНЫЕ ПАРАМЕТРЫ */ MIMA0005
/* NOB - ЧИСЛО ТОЧЕК, */ MIMA0006
/* N - РАЗМЕРНОСТЬ ПРОСТРАНСТВА, */ MIMA0007
/* IT - ПРЕДЕЛЬНОЕ ЧИСЛО ИТЕРАЦИЙ, */ MIMA0008
/* EPS - КОНСТАНТА ОСТАНОВА ИТЕРАЦИОННОГО ПРОЦЕССА, */ MIMA0009
/* A(NOB,N)- МАССИВ КООРДИНАТ ТОЧЕК */ MIMA0010
/* ВЫХОДНЫЕ ПАРАМЕТРЫ */ MIMA0011
/* X(N)- КООРДИНАТЫ ИСКОМОЙ ТОЧКИ, */ MIMA0012
/* ST - СТЕПЕНЬ ВСПОМОГАТЕЛЬНОЙ ФУНКЦИИ, */ MIMA0013
/* R(NOB)- РАСТОЯНИЯ ОТ ИСКОМОЙ ТОЧКИ ДО КАЖДОЙ ИЗ ЗАДАНЫХ */ MIMA0014
/*****/ MIMA0015
DCL (X,XNK,SXMA)(N),(A,XMA)(NOB,N),(R,VPF,PPF,BI,VES)(NOB); MIMA0016
BIN FLOAT(53) CTL,(TF(IT),TX(IT,N)) BIN FLOAT CTL; MIMA0017
(ZNM,EPS,FC,F,RNK) BIN FLOAT(53);(N,NOB,H,ST,I,J) BIN FIXED; MIMA0018
ON ENDFILE(SYSIN) GO TO KON; MIMA0019
RPT CALL TIMES; H=0; MIMA0020
GET LIST(NOB,N,IT,EPS) COPY; MIMA0021
ALLOCATE X,XNK,SXMA,A,XMA,R,VPF,PPF,BI,VES; MIMA0022
GET LIST(A); MIMA0023
CALL PTMATR(A,'МАТРИЦА A'); MIMA0024
X=0; MIMA0025
DO ST=2 TO 130 BY 4; MIMA0026
H=0; FC=1E10; PUT SKIP LIST('СТЕПЕНЬ = ',ST); MIMA0027
CIKL H=H+1; MIMA0028
/* РАСЧЕТ РАСТОЯНИЙ ОТ ТЕКУЩЕЙ ТОЧКИ ДО КАЖДОЙ ИЗ ЗАДАНЫХ */ MIMA0029
DO I=1 TO NOB; MIMA0030
R(I)=SQRT(SUM((X(*)-A(I,*))**2)); MIMA0031
VPF(I)=ST*(ST-1)*R(I)**(ST-2); MIMA0032
PPF(I)=ST*R(I)**(ST-1); MIMA0033

```

```

END,
F=SUM(R(*)**ST),
IF ABS(F-FC)<=EPS*F THEN GO TO OUT;
ELSE FC=F;
/* НАХОЖДЕНИЕ НЬУТОНОВСКОЙ ТОЧКИ
SUMMA=SUM(R(*)**ST-2),
SXMA=0,
DO I=1 TO NOB;
VES(I)=R(I)**(ST-2)/SUMMA,
XMA(I,*)=X-A(I,*);
SXMA=SXMA+XMA(I,*)*VES(I);
END,
XNK=X-SXMA/(ST-1), SXMA=0,
/* НАХОЖДЕНИЕ РАССТОЯНИЙ ОТ НЬУТОНОВСКОЙ ТОЧКИ ДО КАЖДОЙ ИЗ
ЗАДАННЫХ*/
DO I=1 TO NOB,
RNK=SQRT(SUM((XNK(*)-A(I,*))**2)),
ZNM=R(I)-RNK;
IF ZNM=0 THEN ZNM=1E-15,
BI(I)=MAX(VPF(I),(PPF(I)-ST*RNK**ST-1)/ZNM);
SXMA=SXMA+XMA(I,*)*R(I)**(ST-2),
END;
X=X-ST*SXMA/SUM(BI(*)),
GO TO CIKL,
OUT. PUT DATA(H,X),
PUT SKIP LIST('R(I)',R),
END,
FREE X,XNK,SXMA,A,XMA,R,VPF,PPF,BI,VES,
GO TO RPT;
KON. PUT SKIP LIST('КОНЕЦ РАСЧЕТА'),
END MINMAX,

```

## Текст подпрограммы POISK

```

/*****
POISK PROC(L,N,KO,X),
/*****
/* ПОДПРОГРАММА POISK ПРЕДНАЗНАЧЕНА ДЛЯ НАХОЖДЕНИЯ ТОЧКИ,
/* УДОВЛЕТВОРЯЮЩЕЙ ЗАДАННОЙ СИСТЕМЕ ЛИНЕЙНЫХ ОГРАНИЧЕНИЯ
/* ВИДА  $AX+B \leq 0$ 
/* ВХОДНЫЕ ДАННЫЕ
/* L - ЧИСЛО ОГРАНИЧЕНИЯ,
/* N - РАЗМЕРНОСТЬ ПРОСТРАНСТВА,
/* KO(L,N+1) - КОЭФФИЦИЕНТЫ ОГРАНИЧЕНИЯ,
/* X(N) - КООРДИНАТЫ ИСХОДНОЙ ТОЧКИ.
/* ВЫХОДНЫЕ ПАРАМЕТРЫ
/* X(N) - КООРДИНАТЫ ТОЧКИ В ДОПУСТИМОЙ ОБЛАСТИ
/*****
DCL (KO(*,*),NEV,F,EPS,X(*) )BIN FLOAT,
(A(L,N),B(L,OD),(P,NEVI)(L),PRR(N),TF(IT),TX(IT,N))
BIN FLOAT CTL,
(H,I,J,L,N,OD,IT )BIN FIXED;
F=10000, IT=100; EPS=1E-3, OD=1; H=0;
ALLOCATE A,B,P,NEVI,PRR,TF,TX;
A=0, B=0,
DO J=1 TO N,
A(*,J)=KO(*,J),
END,
B(*,1)=-KO(*,N+1),
CIKL:H=H+1,
DO I=1 TO L,
DO J=1 TO N,
PRR(J)=A(I,J)*X(J),
END,
NEVI(I)=SUM(PRR)+KO(I,N+1),
P(I)=0.25/(ABS(NEVI(I))+EPS);
P(I,OD)=-KO(I,N+1)-ABS(NEVI(I));
IF NEVI(I)<=0 THEN NEVI(I)=0;
END,
NEV=SUM(NEVI);
IF (H)=NEV TX(H,*)=X(*);

```

```

IF ABS(NEV)=0 THEN GO TO OVT,
  IF ABS(NEV-F)<EPS I N=IT THEN GO TO OVT,
  F=NEV,
  CALL VVKP(A,B,X,P,L,N,OD),
  GO TO CIKL,
OVT.PUT SKIP EDIT('ПОИСК ТОЧКИ,УДОВЛЕТВОРЯЮЩЕЙ ОГРАНИЧЕНИЯМ',
  (COL(5),A),
  CALL TABL(N,IX,TF,H),
  DO I=1 TO L,
    IF NEVI(I)>0 THEN PUT EDIT('ВНУТРЕННЯЯ ТОЧКА НЕ НАЙДЕНА
    'НАРУШЕНО ОГРАНИЧЕНИЕ',I)(COL(5),A,X(2),A,X(1),F(3)),
  END,
  FREE A,B,P,NEVI,PRR,TF,IX,
  END POISK;

```

## Текст подпрограммы ELLIPS2

```

/*****/00000001
  ELLIPS2.PROC(X,N,OCR,L,Q,Z), 00000002
/*****/00000003
/* ПОДПРОГРАММА ELLIPS СТРОИТ МАЖОРИРУЮЩИЙ ЭЛЛИпсоИД ДЛЯ */00000004
/* СИСТЕМЫ ЛИНЕЙНЫХ ОГРАНИЧЕНИЙ */00000005
/* ВХОДНЫЕ ПАРАМЕТРЫ */00000006
/* X(N) - КООРДИНАТЫ ЦЕНТРА ЭЛЛИпсоИДА, */00000007
/* N - РАЗМЕРНОСТЬ ПРОСТРАНСТВА, */00000008
/* L - ЧИСЛО ОГРАНИЧЕНИЙ, */00000009
/* OCR(L,N+1)- КОЭФФИЦИЕНТЫ ОГРАНИЧЕНИЯ, */00000010
/* Z - ПАРАМЕТР,ОПРЕДЕЛЯЮЩИЙ РАЗМЕРЫ ЭЛЛИпсоИДА */00000011
/* ВЫХОДНЫЕ ПАРАМЕТРЫ */00000012
/* Q(N,N) - МАТРИЦА,ОПРЕДЕЛЯЮЩАЯ РАЗМЕРЫ ЭЛЛИпсоИДА И */00000013
/* ОРИЕНТАЦИЮ ЕГО ОСЕЙ */00000014
/* */00000015
/*****/00000016
  DCL (X(*),(OCR,Q)(*,*),BOL,Z ) BIN FLOAT, 00000017
  ((T,TT,QT)(N,N),(PRR,PRR1)(N),(RASST,DOPR)(L))BIN FLOAT CTL,00000018
  NOMER(L) BIN FIXED CTL,(I,0,0,N,L,IMEEM) BIN FIXED, 00000019
  (NOGR(L,N+1),ANOCR(L,N) BNOCR(L),LBDB(N),KSIMX(L,N),IND(L), 00000020
  C(ITRA,ITRA),D(ITRA,OD)) BIN FLOAT CTL,OPT CHAR(1), 00000021
  IPER(ITRA) BIN FIXED CTL,RASSM(L) BIN FLOAT CTL, 00000022
  CD(ITRA,ITRA) BIN FLOAT CTL, DET BIN FLOAT,CON BIN FLOAT, 00000023
  OD=1, 00000024
  ALLOCATE T,TT,QT,PRR,PRR1,RASST,DOPR,NOMER, 00000025
  ALLOCATE NOGR,ANOCR,BNOCR,LBDB,KSIMX,IND,RASSM, 00000026
  Q=0, OPT='5', T=0, CON=0, 00000027
/* НОРМИРОВАНИЕ СИСТЕМЫ ОГРАНИЧЕНИЙ */00000028
DO I=1 TO L, 00000029
  NOGR(I,*)=OCR(I,*)/SQRT(SUM(OCR(I,*)**2)-OCR(I,N+1)**2), 00000030
END, 00000031
DO I=1 TO L, 00000032
  DO J=1 TO N, 00000033
    ANOCR(I,J)=NOGR(I,J), 00000034
  END, 00000035
  BNOCR(I)=NOGR(I,N+1), 00000036
END, 00000037
DO I=1 TO L, 00000038
  NOMER(I)=I, IND(I)=1, 00000039
  DO J=1 TO N, 00000040
    PRR(J)=ANOCR(I,J)*X(J), 00000041
  END, 00000042
  RASST(I)=ABS(SUM(PRR)+BNOCR(I)), 00000043
END, 00000044
RASSM=RASST, 00000045
CALL SORT(RASSM,NOMER,L), 00000046
T(1,*)=ANOCR(NOMER(1),*), 00000047
IND(NOMER(1))=0, 00000048
Q(1,1)=Z/RASST(NOMER(1))**2, 00000049
IF L=1 THEN DO, 00000050
  IMEEM=1, 00000051
  CALL ORTGR1 (T,N,IMEEM), 00000052
  DO I1=IMEEM+1 TO N, 00000053

```

```

      Q(I1,I1)=Z/10**5;
      END;
      GO TO MT2;
      END;
      RASST(NOMER(1))=10**10;
      K=0;
ITR  K=K+1, ITRA=K+1,
      ALLOCATE C,D,IPER,CD,
      C=0, D=0, IR=0;
      DO J1=1 TO L,
        NOMER(J1)=J1;
        IF IND(J1)~=0 THEN DO,
          DO I2=1 TO K,
            C(I2,I2)=1,
            C(I2,ITRA)=SUM(T(I2,*)*
                          ANOGR(J1,*)),
            C(ITRA,I2)=C(I2,ITRA),
          END;
          C(ITRA,ITRA)=1,
          D(ITRA,1)=SUM(X(*)*ANOGR(J1,*))+
                     BNOGR(J1),
          CD=C,
          CALL MINV(CD,ITRA,DET,CON),
          IF ABS(DET)<=0.0001 THEN DO,
            IR=IR+1;
            RASST(J1)=10**10;
            GO TO MT1;
          END,
          CALL MFG(C,IPER,ITRA,IS);
          CALL MDLG(C,D,IPER,ITRA,OD,OPT),
          LBDB=0,
          DO I2=1 TO K,
            LBDB =LBDB+D(I2,1)*T(I2,*),
          END,
          KSIMX(J1,*)=D(ITRA,1)*ANOGR(J1,*)+LBDB,
          RASST(J1)=SQRT(SUM(KSIMX(J1,*)**2));
          C=0, D=0,
        END,
      END,
MT1  IF IR=L-ITRA+1 & IR~=0 THEN DO,
      IMEEM=ITRA-1,
      CALL ORTGR1(T,N,IMEEM),
      DO I1=ITRA TO N,
        Q(I1,I1)=Z/10**5,
      END,
      GO TO MT2,
      END,
      RASSM=RASST,
      CALL SORT(RASSM,NOMER,L),
      T(ITRA,*)=KSIMX(NOMER(1),*)/SQRT(SUM(KSIMX(NOMER(1),*)**2)),
      IND(NOMER(1))=0,
      Q(ITRA,ITRA)=Z/RASST(NOMER(1))**2,
      RASST(NOMER(1))=10**10,
      FREE C,D,IPER,CD,
      IF L<N & SUM(IND)=0 THEN DO,
        IMEEM=ITRA,
        CALL ORTGR1(T,N,IMEEM),
        DO I1=ITRA+1 TO N,
          Q(I1,I1)=Z/10**5,
        END,
        GO TO MT2,
      END,
      IF SUM(IND)~=L-N & SUM(IND)~=0 THEN GO TO ITR,
MT2  CALL MMGG(Q,T,N,N,N,QT),
      CALL TRANSP(T,TT,N,N),
      CALL MMGG(TT,QT,N,N,N,Q),
      FREE T,TT,QT,PRR,PRR1,RASST,DOPR,NOMER,
      NOGR,ANOGR,BNOGR,LBDB,KSIMX,IND,RASSM;
      END ELLIPS2,

```

# Текст программы АКМ

```

      АКМ PROC OPTIONS(MAIN),
/* ***** */
/* ПРОГРАММА АКМ ПРЕДНАЗНАЧЕНА ДЛЯ РЕШЕНИЯ ЗАДАЧИ ШТЕРНЕРА- */
/* НАХОЖДЕНИЯ ТОЧКИ НА МИНИМАЛЬНОМ РАССТОЯНИИ ОТ СОВОКУПНОСТИ */
/* ТЧЕК В ЕВКЛИДОВОЙ МЕТРИКЕ ПРИ НАЛИЧИИ ЛИНЕЙНЫХ ОГРАНИЧЕНИЯ */
/*
/* ВХОДНЫЕ ПАРАМЕТРЫ
/* N1 -ЧИСЛО ИСХОДНЫХ ТОЧЕК,
/* N -РАЗМЕРНОСТЬ ПРОСТРАНСТВА,
/* KRD(N1,N) -МАССИВ КООРДИНАТ ИСХОДНЫХ ТОЧЕК,
/* W(N1) -ВЕКТОР ВЕСОВЫХ КОЭФФИЦИЕНТОВ,
/* IT -ОГРАНИЧЕНИЕ ПО ЧИСЛУ ИТЕРАЦИЙ,
/* OGR(L,N+1) -КОЭФФИЦИЕНТЫ ОГРАНИЧЕНИЙ
/* L -ЧИСЛО ОГРАНИЧЕНИЙ
/* EPS -КОНСТАНТА ОСТАНОВКИ ПРОЦЕССА
/* ВЫХОДНЫЕ ПАРАМЕТРЫ
/* X(N) -КООРДИНАТЫ ИСКОМОЙ ТОЧКИ НА N- ЕЯ ИТЕРАЦИИ,
/* NEVF -СУММАРНОЕ РАССТОЯНИЕ НА N- ЕЯ ИТЕРАЦИИ,
/* N -НОМЕР ИТЕРАЦИИ
/* ИСПОЛЗУЮТСЯ ПОДПРОГРАММЫ SORT,VVPR,ELLIPS,RANDOM,ORTGR1,
/* POISK,TABL,TIMES,PTMATR,TPR2,UMV,
/* TRANSP
/* ИЗ ПАКЕТА ПНП ПС-1 MSLQ,MMCG,MINV,MFG,MDLG
/* ***** */
DCL (KRD(N1,N),(W,NEVI,NEVB)(N1),X(N),A(M,N),B(M,OD),P(M),
      TF(IT),TX(IT,N),OGR(L,N+1),XSTAR(N) ) BIN FLOAT CTL,
      (EPS,F,NEVF) BIN FLOAT,(N1,N,IT,M,OD,H,JJ,L,NC) BIN FIXED,
      (RASST(L),PRR(N) ) BIN FLOAT CTL,
RANDOM RETURNS(BIN FLOAT),
      (FUN,Z) BIN FLOAT;
/* ВВОД И РАСПЕЧАТКА ИСХОДНЫХ ДАННЫХ
CALL TIMES,
DAN ON ENDFILE(SYSIN) GO TO KON,
GET LIST(N1,N,IT,EPS,L) COPY,
NAP, FORMAT(SKIP(2),COL(10),A,F(3)),
PUT EDIT('ЧИСЛО ТОЧЕК=',N1)(R(NAP)),
PUT EDIT('ЧИСЛО ОГРАНИЧЕНИЙ=',L)(R(NAP)),
PUT EDIT('РАЗМЕРНОСТЬ ПРОСТРАНСТВА=',N)(R(NAP));
M=N1*N; OD=1, H=0, F=10**10,
ALLOCATE KRD,W,X,A,B,NEVI,NEVB,P,TF,TX,OGR,XSTAR,RASST,PRR;
GET LIST(KRD,W,OGR);
CALL PTMATR(KRD,'КООРДИНАТЫ ТОЧЕК'),
PUT SKIP LIST('ВЕСА ТОЧЕК',W);
CALL PTMATR(OGR,'КОЭФФИЦИЕНТЫ ОГРАНИЧЕНИЙ');
GET LIST(X);
PUT SKIP LIST('КООРДИНАТЫ ИСХОДНОЙ ТОЧКИ',X);
/* ФОРМИРОВАНИЕ МАТРИЦЫ А И ВЕКТОРА В ДЛЯ СИСТЕМЫ УРАВНЕНИЙ AX=B */
JJ=0, A=0, B=0,
DO I=1 TO M BY N,
DO J=1 TO N,
A(I+J-1,J)=1,
END;
END;
DO I=1 TO N1,
DO J=1 TO N,
JJ=JJ+1,
B(JJ,1)=KRD(I,J),
END;
END;
CALL POISK(L,N,OGR,X),
H=0,Z=N,
SIKL H=H+1; /* СЧЕТЧИК ЧИСЛА ИТЕРАЦИЙ
IF H=1 & Z=N THEN Z=1;
JJ=0, NEVI=0;
/* ПРОВЕРКА СОБЛЮДЕНИЯ ОГРАНИЧЕНИЙ И ПРАВИЛА ОСТАНОВА
/* В СЛУЧАЕ ПОПАДАНИЯ В ОКРЕСТНОСТЬ ОГРАНИЧЕНИЯ
IF Z =1 THEN DO;
DO I=1 TO L,

```

|                                                               |            |
|---------------------------------------------------------------|------------|
| DO J=1 TO N,                                                  | AKM 0069   |
| PRR(J)=OCR(I,J)*X(J),                                         | AKM 0070   |
| END,                                                          | AKM 0071   |
| RASST(I)=SUM(PRR)+OCR(I,N+1),                                 | AKM 0072   |
| IF RASST(I)>0 THEN DO,                                        | AKM 0073   |
| PUT SKIP EDIT('НАРУШЕНО ОГРАНИЧЕНИЕ ',I,' ОМЕТКА= ',RASST(I)) | AKM 0074   |
| (A,F(3),A,F(10,7)),                                           | AKM 0075   |
|                                                               | AKM 0076   |
| Z=N, H=H-1,                                                   | AKM 0077   |
| GO TO POVT,                                                   | AKM 0078   |
| END,                                                          | AKM 0079   |
| END,                                                          | AKM 0080   |
| XSTAR=X,                                                      | AKM 0081   |
| DO I=1 TO N1,                                                 | AKM 0082   |
| NEVB(I)=SQRT(SUM((KRD(I,*)-X(*))**2)),                        | AKM 0083   |
| NEVI(I)=NEVB(I)*W(I),                                         | AKM 0084   |
| DO J=1 TO N,                                                  | AKM 0085   |
| JJ=JJ+1,                                                      | AKM 0086   |
| P(JJ)=W(I)/(NEVB(I)+1E-5),                                    | AKM 0087   |
| END,                                                          | AKM 0088   |
| END,                                                          | AKM 0089   |
| NEVF=SUM(NEVI),                                               | AKM 0090   |
| TF(H)=NEVF, TX(H,*)=X(*),                                     | AKM 0091   |
| /* ПРОВЕРКА УСЛОВИЯ ЗАВЕРШЕНИЯ ИТЕРАЦИОННОГО ПРОЦЕССА         | */AKM 0092 |
| IF ABS(NEVF-F)<EPS   H=IT THEN GOTO OUT,                      | AKM 0093   |
| F=NEVF,                                                       | AKM 0094   |
| POVT CALL TPR2(X,XSTAR,N,M,OCR,P,A,B,L,Z),                    | AKM 0095   |
| GOTO CIKL,                                                    | AKM 0096   |
| /* ПЕЧАТЬ РЕЗУЛЬТАТОВ РАСЧЕТА                                 | */AKM 0097 |
| OUT CALL TABL(N,TX,TF,H),                                     | AKM 0098   |
| CALL TIMES,                                                   | AKM 0099   |
| FREE KRD,W,X,A,B,NEVI,NEVB,P,TF,TX,OCR,XSTAR,RASST,PRR,       | AKM 0100   |
| GO TO DAN,                                                    | AKM 0101   |
| KON                                                           | AKM 0102   |
| END AKM,                                                      | AKM 0103   |

## Текст вспомогательных программ

|                                                                       |            |
|-----------------------------------------------------------------------|------------|
| *****                                                                 | PRUC0001   |
| TPR2 PROC(X,XSTAR,N,M,OCR,P,B,C,L,Z),                                 | PROC0002   |
| *****                                                                 | PROC0003   |
| /* ПОДПРОГРАММА TPR СЛУЖИТ ДЛЯ НАХОЖДЕНИЯ ТОЧКИ ПЕРЕСЕЧЕНИЯ           | */PROC0004 |
| /* МАЖОРИРУЮЩЕГО ЭЛЛИПСОИДА С МАЖОРАНТОМ ЦЕЛЕВОЙ ФУНКЦИИ,             | */PROC0005 |
| /* В КОТОРОЙ ЗНАЧЕНИЕ ЦЕЛЕВОЙ ФУНКЦИИ МИНИМАЛЬНО                      | */PROC0006 |
| *****                                                                 | PROC0007   |
| DCL ((X,XSTAR,P)(*), (OCR,B,C)(*), (LAMBDA,D/Z) EIN FLOAT, PROC0008   |            |
| ((BX,BXMC)(M), (BXMC,VBVT,BVBT,PK)(OD,M), (BTPB,BTPM1)(OD,N) PROC0009 |            |
| , (BT,BTPK)(N,M), LAM(OD), (BTPKB,PERM,Q,QKM1)(N,N), PROC0010         |            |
| BTPKC(N,OD), (QX,VTOR)(N), PK(M,M) ) BIN FLOAT CTL, PROC0011          |            |
| (I,OD,N,M,L,J) ) BIN FIXED, PROC0012                                  |            |
| OD=1,                                                                 | PROC0013   |
| ALLOCATE BX,BXMC,BXMC,TP,BTPB,QKM1,BTPM1,BT,BVBT,BVBT, LAM,           | PROC0014   |
| BTPK,BTPKB,PERM,Q,BTPKC,QX,VTOR,PK, Q=0, PK=0,                        | PROC0015   |
| CALL ELLIP2 (XSTAR,N,OCR,L,D,Z),                                      | PROC0016   |
| CALL UMV(M,N,B,XSTAR,BX),                                             | PROC0017   |
| BXMC(*)=BX(*)-C(*,1),                                                 | PROC0018   |
| BXMC,TP(OD,*)=BXMC(*)*P(*),                                           | PROC0019   |
| CALL MMGC(BXMC,TP,B,OD,M,N,BTPB),                                     | PROC0020   |
| QKM1=Q,                                                               | PROC0021   |
| CON=0,                                                                | PROC0022   |
| CALL MINV(QKM1,N,D,CON),                                              | PROC0023   |
| CALL MMGC(BTPB,QKM1,OD,N,N,BTPM1),                                    | PROC0024   |
| CALL TRANSP(B,BT,M,N),                                                | PROC0025   |
| CALL MMGC(BTPM1,BT,OD,N,M,BVBT),                                      | PROC0026   |
| BVBT,PK(1,*)=BVBT(1,*)*P(*),                                          | PROC0027   |
| CALL UMV(OD,N,BVBT,PK,BXMC,LAM),                                      | PROC0028   |
| LAMBDA=SQRT(LAM(1)),                                                  | PROC0029   |
| ОПРЕДЕЛЕНИЕ НОВОГО ЗНАЧЕНИЯ X                                         | */PROC0030 |
| DO I=1 TO M, PK(I,I)=P(I), END,                                       | PROC0031   |
| CALL MMGC(BT,PK,N,M,M,BTPK),                                          | PROC0032   |

```

CALL MMGL(BTPK,B,N,M,N,BTPKB),
PERM=BTPKB+LAMBDA*Q,
CON=0,
CALL MINV(PERM,N,D,CON),
CALL MMGL(BTPK,C,N,M,QD,BTPKC),
CALL UMV(N,N,Q,XSTAR,QX),
QX=LAMBDA*QX,
VTOR(*)=BTPKC(*,1)+QX(*),
CALL UMV(N,N,PERM,VTOR,X),
FREE BX,BXMC,BXMCST,BTPB,QKM1,BTPQM1,BT,BBVT,BBVTBK,LAM,
BTPK,BTPKB,PERM,Q,BTPKC,QX,VTOR,PK,
END TPR2,
/*****
ORTGR1 PROC(A,N,IMEEM),
/*****
/* ПОДПРОГРАММА ORTGR1 ПРЕДНАЗНАЧЕНА ДЛЯ ПОСТРОЕНИЯ ОРТОГОНАЛЬ-
/* НОЙ СИСТЕМЫ ВЕКТОРОВ ПО МЕТОДУ ГРАММА-ШМИДТА
/*****
DCL (A(*,*)) BIN FLOAT,(V,VORN)(N,N) BIN FLOAT CTL,
SUMMA(N) BIN FLOAT CTL,
(I,J,N,IMEEM) BIN FIXED,
ALLOCATE V,VORN,SUMMA,
DO I=1 TO IMEEM,
VORN(I,*)=A(I,*),
END,
DO I=IMEEM+1 TO N,
DO J=1 TO N,
A(I,J)=RANDOM,
END,
END,
DO I=IMEEM+1 TO N,
SUMMA=0,
DO J=1 TO I-1,
SUMMA=SUMMA+SUM(A(I,*)*VORN(J,))*VORN(J,*),
END,
V(I,*)=A(I,*)-SUMMA,
VORN(I,*)=V(I,*)/SQRT(SUM(V(I,)**2)),
END,
A=VORN,
FREE VORN,V,SUMMA,
END ORTGR1,
/*****
RANDOM PROC RETURNS(BIN FLOAT),
/*****
/* ПРОЦЕДУРА RANDOM ГЕНЕРИРУЕТ СЛУЧАЙНОЕ ЧИСЛО
/* С ПЛАВАЮЩЕЙ ТОЧКОЙ В ИНТЕРВАЛЕ (0,1)
/*****
DCL IX FIXED BINARY (31,0) STATIC INITIAL(51771),
IY FIXED BINARY(31,0),YFL FLOAT DEC,
(NOFIXEDOVERFLOW)
IY=IX*65539,
IF IY<0 THEN IY=IY+2147483647+1,
YFL=IY,
YFL=YFL*0.4656613E-9,
IX=IY,
RETURN(YFL),
END RANDOM,
/*****
SORT PROC(A,C,N),
/*****
/* ПРОЦЕДУРА SORT СЛУЖИТ ДЛЯ УПОРЯДОЧЕНИЯ ЭЛЕМЕНТОВ МАССИВА
/* ОТ БОЛЬШЕГО К МЕНЬШЕМУ
/*****
DCL (I,J,K) BIN FIXED,
(U,A(*)) BIN FLOAT,
(C(*),CS,N) BIN FIXED,
DO I=1 TO N-1,
K=I,
DO J=I TO N,

```

16. Литвак Б. Г. Экспертная информация. Методы получения и анализа. — М.: Радио и связь, 1982. — 184 с.
17. Лоусон Ч. Х., Хенсон Р. Численное решение задач метода наименьших квадратов: Пер. с англ. — М.: Наука, 1968. — 232 с.
18. Маршал А., Олкин И. Неравенства: теория мажоризации и ее приложения: Пер. с англ./Под ред. В. М. Круглова. — М.: Мир, 1983. — 574 с.
19. Моисеев Н. Н., Иванилов Б. П., Столярова Е. М. Методы оптимизации. — М.: Наука, 1978. — 351 с.
20. Мудров В. И. Задача о коммивояжере. — М.: Знание, 1969. — 60 с. — (Сер. Математика и кибернетика. Вып. 10).
21. Мудров В. И., Кушко В. Л. Метод наименьших модулей. — М.: Знание. — 1971. — 64 с. — (Сер. Математика и кибернетика. Вып. 7).
22. Мудров В. И., Кушко В. Л. Методы обработки измерений. Квазиправдоподобные оценки. — Изд. 2-е переработанное и исправл. — М.: Радио и связь, 1983. — 303 с.
23. Ослон А. А. О задаче одномерного метрического шкалирования // Автоматика и телемеханика. — 1984. — № 6. — С. 115—121.
24. Плинер В. М. Об одном классе моделей метрического шкалирования // Автоматика и телемеханика. — 1984. — № 6. — С. 123—128.
25. Поляк Б. Т. Введение в оптимизацию. — М.: Наука, 1983. — 384 с.
26. Ремез Е. Я. Вычислительные методы чебышевских приближений. — Киев: Наукова думка, 1956. — 508 с.
27. Розен В. В. Цель — оптимальность — решение. — М.: Радио и связь, 1982. — 169 с.
28. Смоляк С. А., Титаренко Б. П. Устойчивые методы оценивания. — М.: Статистика, 1980. — 208 с.
29. Стахов А. П. Введение в алгоритмическую теорию измерения. — М.: Сов. радио, 1977. — 288 с.
30. Стренг Г. Линейная алгебра и ее применения: Пер. с англ. /Под ред. Г. И. Марчука. — М.: Мир, 1980. — 454 с.
31. Фиакко А., Мак-Кормик Г. Нелинейное программирование. — М.: Мир, 1972. — 240 с.
32. Химмельблау Д. Прикладное нелинейное программирование: Пер. с англ./Под ред. М.Л. Быховского. — М.: Мир, 1975. — 534 с.
33. Хованов Н. В. Математические основы теории шкал измерения качества. — Л.: ЛГУ, 1982. — 186 с.
34. Хьюбер П. Робастность в статистике: Пер. с англ./Под ред. И. Г. Журбенко. — М.: Мир, 1984. — 303 с.
35. Chen R. Solution of Location Problems with Radial Cost Functions. — Comp. and Math. with Appl. — 1984. — Vol. 10, N 1. — P. 87—94.
36. Francis R. L., McGinnis L. F. Location Analysis // European J. Operational Res. — 1983. — Vol. 12. — P. 220—252.
37. Ignizio J. P., Thomas L. C. An Enhanced Conversion Scheme for Lexicographic, Multiobjective Integer Programs // European J. Operational Res. — 1984. — Vol. 18. — P. 57—61.
38. Jech T. The Ranking of Incomplete Tournaments: A Mathematician's Guide to Popular Sports // Amer. Math. Monthly. — 1983. — April — P. 246 — 266.
39. Morris J. G., Norback J. P. Linear Facility Locations Solving Extensions of the Basic Problem // European J. Operational Res., — 1983. — Vol. 12. — P. 90—94.
40. Vob H., Echard U. Linear Convergence of the Generalized Weiszfeld's Method // Computing. — 1980. — Vol. 25. — P. 243—251.
41. Weiszfeld E. Sur le point par lequel la somme des distances de n points donnees est minimum // Tohoku Math. J. — 1937 — Vol. 37. — P. 355—386.

```

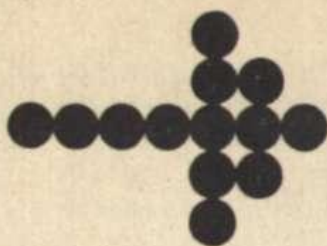
IF A(J) <= A(I) THEN I=J,
PROC000 END,
PROC000 U=A(I), A(I)=A(J), A(J)=U,
PROC000 CS=C(I), C(I)=C(J), C(J)=CS,
PROC000 END,
PROC000 END SORT,
PROC000 /*****
PROC000 UMV PROC(M,N,A,X,C),
PROC000 *****/PROC0110
PROC000 * ПРОЦЕДУРА УМНОЖЕНИЯ МАТРИЦЫ НА ВЕКТОР */PROC0111
PROC000 *****/PROC0112
PROC000 DCL (A(*,*),X(*),C(*)) BIN FLOAT,(I M,N) BIN FIXED
PROC000 DO I=1 TO M, C(I)=SUM(A(I,*)*X(*)), END,
PROC000 END UMV,
PROC000 /*****
**/PROC000 TRANSP PROC(A,AT M,N),
**/PROC000 *****/PROC0116
**/PROC000 * ТРАНСПОНИРОВАНИЕ МАТРИЦЫ */PROC0117
**/PROC000 *****/PROC0120
PROC000 DCL (A,AT)(* *) BIN FLOAT I,J,M,N
PROC000 DO I=1 TO M,
PROC000 DO J=1 TO N,
PROC000 AT(J,I)=A(I,J),
PROC000 END,
PROC000 END,
PROC000 END TRANSP,
PROC000
PROC000
PROC000
PROC000
PROC000
PROC000
PROC000
PROC000

```

#### СПИСОК ЛИТЕРАТУРЫ

- ПРОС000 Чоки М. Введение в методы оптимизации. Пер. с англ. Под ред. Б. Т. Поля-  
 ПРОС000 1.— М.: Наука, 1977. — 343 с.  
 ПРОС000 Чазара К. Нелинейное программирование: Пер. с англ./Под ред. Д. Б. Юди-  
 ПРОС000 1.— М.: Мир, 1982.—543 с.  
 ПРОС000 Береснев В. Л., Гимади Э. Х., Дементьев В. Т. Экстремальные задачи стан-  
 ПРОС000 дартизации.— Новосибирск: Наука, 1978.—335 с.  
 ПРОС000 Верховятницкий П. Д., Латинский В. С. Справочник по модульному проек-  
 ПРОС000 тированию радиоэлектронной аппаратуры.— Л.: Судостроение, 1983.—151с.  
 ПРОС000 Геминтерн В. И., Каган Б. М. Методы оптимального проектирования.  
 \*\*/ПРОС000 — М.: Энергия, 1980.— 159 с.  
 \*\*/ПРОС000 Гилл Ф., Мюррей У., Райт М. Практическая оптимизация: Пер. с англ./  
 \*\*/ПРОС000 Под ред. А. А. Петрова. — М.: Мир, 1985. — 509 с.  
 ПРОС000 Горелик А. Л., Скрипкин В. А. Методы распознавания.— М.: Высшая школа,  
 ПРОС000 1984.— 208 с.  
 ПРОС000 Гребенников Л. К., Лебедев В. Н. Решение задач на ПЛ-1 в ОС ЕС. — М.:  
 ПРОС000 Финансы и статистика, 1981.— 302 с.  
 ПРОС000 Гроссман К., Каплан А. А. Нелинейное программирование на основе безу-  
 ПРОС000 словной оптимизации.— Новосибирск: Наука, 1981.— 181 с.  
 ПРОС000 Гурин Л. С., Дымарский Я. С., Меркулов А. Д. Задачи и методы оптималь-  
 ПРОС000 ного распределения ресурсов.— М.: Сов. радио, 1968. — 463. с.  
 ПРОС000 Демьянов В. Ф. Задачи негладкой оптимизации и квазидифференциалы //  
 ПРОС000 Изв. АН СССР. Техническая кибернетика. — 1983. — № 1. — С. 9—19.  
 \*\*/ПРОС000 Дикин И. И., Зоркальцев В. И. Итеративное решение задач математического  
 \*\*/ПРОС000 программирования.— Новосибирск: Наука, 1980. — 144 с.  
 \*\*/ПРОС000 Дэвид Г. Метод парных сравнений: Пер. с англ./Под ред. Ю. Адлера. —  
 ПРОС000 М.: Статистика, 1978.— 137 с.  
 ПРОС000 Каменский В. С. Методы и модели неметрического многомерного шкалирова-  
 ПРОС000 ния: Обзор //Автоматика и телемеханика.— 1977. — № 8. — С. 118—155.  
 ПРОС000 Клигер С. А., Косолапов М. С., Толстова Ю. Н. Шкалирование при сборе  
 ПРОС000 и анализе социологической информации.— М.: Наука, 1978. — 142. с.

|                                                                                                               |  |
|---------------------------------------------------------------------------------------------------------------|--|
| 1.2. Численные методы отыскания абсолютного экстремума . . .                                                  |  |
| 1.2.1. Методы нулевого порядка (или прямые методы) . . .                                                      |  |
| 1.2.2. Методы первого порядка (или градиентные методы) . . .                                                  |  |
| 1.2.3. Методы второго порядка (или релаксационные методы). . .                                                |  |
| 1.3. Заключение . . .                                                                                         |  |
| <b>Г л а в а 2. Метод вариационно-взвешенных квадратических мажорант</b>                                      |  |
| 2.1. Основы метода ВВКМ . . .                                                                                 |  |
| 2.2. Приемы мажорирования . . .                                                                               |  |
| 2.3. Результаты Ремеза и дальнейшее расширение возможностей метода ВВКМ . . .                                 |  |
| 2.4. Линейная сходимость метода ВВКМ . . .                                                                    |  |
| 2.5. Заключение . . .                                                                                         |  |
| <b>Г л а в а 3. Решение некоторых задач размещения методом вариационно-взвешенных квадратических мажорант</b> |  |
| 3.1. Размещение точечных объектов. Минимизация суммарных расстояний                                           |  |
| 3.2. Размещение точечных объектов. Минимаксные задачи . . .                                                   |  |
| 3.3. Размещение линейного объекта . . .                                                                       |  |
| 3.4. Числовые примеры решения задач размещения методом ВВКМ . .                                               |  |
| 3.5. Одна задача из области статистики . . .                                                                  |  |
| 3.6. Заключение . . .                                                                                         |  |
| <b>Г л а в а 4. Применение мажорант в задачах обработки измерений и экспертных данных</b>                     |  |
| 4.1. Мажоранты в задачах обработки измерений . . .                                                            |  |
| 4.2. Виды экспертных данных . . .                                                                             |  |
| 4.3. Обработка экспертных данных, представленных в форме неравенств                                           |  |
| 4.4. Обработка «ненаправленных» соотношений . . .                                                             |  |
| 4.5. Пример количественной оценки качества изделий методом одномерного метрического шкалирования . . .        |  |
| 4.6. Заключение . . .                                                                                         |  |
| <b>Г л а в а 5. Мажоранты в задачах условной оптимизации . . .</b>                                            |  |
| 5.1. Мажорирование прямоугольных ограничений . . .                                                            |  |
| 5.2. Мажорирование произвольной системы линейных ограничений                                                  |  |
| 5.3. Алгоритм комплексного мажорирования . . .                                                                |  |
| 5.4. Использование штрафных функций . . .                                                                     |  |
| 5.5. Поискные методы с мажорированием ограничений . . .                                                       |  |
| 5.6. Заключение . . .                                                                                         |  |
| Приложение. Программная реализация метода ВВКМ . . .                                                          |  |
| Список литературы . . .                                                                                       |  |



45 коп.



Книги серии «Кибернетика»

Вы сможете приобрести в магазинах, распространяющих  
научно-техническую литературу



«РАДИО и СВЯЗЬ»