

**Современный
Гуманитарный
Университет**

Дистанционное образование

Рабочий учебник

Фамилия, имя, отчество _____

Факультет _____

Номер контракта _____

МЕТОДЫ ОПТИМИЗАЦИИ

ЮНИТА 2

**ПРЯМЫЕ МЕТОДЫ ОТЫСКАНИЯ ЭКСТРЕМУМА
И ОСНОВЫ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ**

МОСКВА 1999

Разработано В.Н. Кузубовым

Рекомендовано Министерством общего
и профессионального образования
Российской Федерации в качестве
учебного пособия для студентов
высших учебных заведений

КУРС: МЕТОДЫ ОПТИМИЗАЦИИ

Юнита 1. Вариационное исчисление и методы оптимального управления.
Юнита 2. Прямые методы отыскания экстремума и основы математического программирования.

ЮНИТА 2

Рассмотрены прямые методы отыскания экстремума: одномерный и многомерный поиск экстремума; последовательный поиск экстремума; метод рандомизации и другие. Значительное внимание уделено также основам линейного программирования - симплекс-методу в линейном программировании, прямой и двойственной задачам линейного программирования. Рассмотрены такие методы оптимизации, как нелинейное и целочисленное программирование, классические методы определения экстремума функции. Приведено много поясняющих примеров и иллюстраций.

Для студентов Современного Гуманитарного Университета

Юнита соответствует профессиональной образовательной программе № 2

ОГЛАВЛЕНИЕ

ДИДАКТИЧЕСКИЙ ПЛАН	4
ЛИТЕРАТУРА	5
ПЕРЕЧЕНЬ УМЕНИЙ	6
ТЕМАТИЧЕСКИЙ ОБЗОР	7
1. Прямые методы отыскания экстремума	7
1.1. Одномерный поиск экстремума	7
1.2. Последовательный поиск экстремума	17
1.3. Метод рандомизации	25
1.4. Методы многомерного поиска экстремума	28
1.4.1. Градиентный метод	28
1.4.2. Методы ньютона и секущих	31
1.4.3. Овражный метод	33
1.5. Методы отыскания экстремума в условиях помех	34
2. Основы линейного программирования	37
2.1. Задачи линейного программирования	37
2.1.1. Математическая формулировка задачи линейного программирования	37
2.1.2. Примеры прикладных задач линейного программирования	38
2.2. Геометрическая интерпретация задач линейного программирования	40
2.3. Симплекс-метод в линейном программировании	43
2.4. Формализованная симплекс-таблица	49
2.5. Прямая и двойственная задачи линейного программирования	52
2.5.1. Введение в проблему двойственности	52
2.5.2. Двойственный симплекс-метод	54
2.6. Общая теория симплекс-метода на базе линейной алгебры ..	56
3. Нелинейное и целочисленное программирование.	
Классические методы определения экстремума функций	60
3.1. Нелинейное программирование	60
3.2. Классические методы определения экстремумов функции ..	65
3.2.1. Задача на абсолютный экстремум	65
3.2.2. Задача на условный экстремум	67
3.3. Целочисленное программирование	68
3.3.1. Особенности задач целочисленного программирования	68
3.3.2. Нелинейное и целочисленное программирование	73
3.4. Методы отсечения	76
3.5. Комбинаторные методы	78
3.5.1. Решение задач целочисленного программирования с помощью динамического программирования	78
3.5.2. Метод ветвей и границ	81
3.6. Другие методы оптимизации	85
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ	87
ТРЕНИНГ УМЕНИЙ	89

* Глоссарий расположен в середине учебного пособия и предназначен для самостоятельного заучивания новых понятий.

ДИДАКТИЧЕСКИЙ ПЛАН

Прямые методы отыскания экстремума. Одномерный поиск экстремума. Последовательный поиск экстремума. Метод рандомизации. Многомерный поиск. Градиентный метод. Овражный метод. Метод отыскания экстремума в условиях помех.

Основы линейного программирования. Задачи линейного программирования. Геометрическая интерпретация задач линейного программирования. Симплекс-метод в линейном программировании. Прямая и двойственная задачи линейного программирования. Характеристика других методов оптимизации.

Нелинейное программирование. Классические методы определения экстремума функции. Целочисленное программирование. Оптимизация на графах.

ЛИТЕРАТУРА

Базовая

1. Банди Б. Методы оптимизации. Вводный курс. М.: Радио и связь, 1988.

Дополнительная

2. Боглаев Ю. П. Вычислительная математика и программирование. М.: Высшая школа, 1990.
3. Данциг Дж. Линейное программирование, его применение и обобщение. М.: Прогресс, 1966.
4. Никайдо Х. Выпуклые структуры и математическая экономика. М.: Мир, 1972.
5. Сухарев А.Г, Тимохов А.В., Федоров В.В. Курс методов оптимизации. М.: Наука, 1986.

ПЕРЕЧЕНЬ УМЕНИЙ

№ п/п	Умение	Алгоритмы
1.	Нахождение условных экстремумов методом Лагранжа.	<ol style="list-style-type: none"> 1. Построить функцию Лагранжа. 2. Найти частные производные первого порядка для $L(x_1, x_2, \lambda)$. 3. Составить систему уравнений для нахождения стационарных точек $L(x_1, x_2, \lambda)$. 4. Решить систему уравнений. 5. Найти экстремумы функции.
2.	Решение задачи линейного программирования с помощью симплекс-метода	<ol style="list-style-type: none"> 1. Ввести вспомогательные переменные. 2. Преобразовать задачу к стандартной форме. 3. Получить каноническую форму задачи, принимая в качестве базисных переменных x_1, x_2. 4. Провести анализ поведения целевой функции в зависимости от изменений x_4. 5. Определить, какую базисную переменную необходимо заменить. 6. Найти значения коэффициентов в новой канонической форме после замены базиса. 7. Получить новую каноническую форму после замены базиса на x_1 и x_4. 8. Провести анализ поведения целевой функции в зависимости от изменений свободных переменных x_3 и x_2. 9. Найти оптимальное решение.
3.	Исследование функций на безусловный экстремум	<ol style="list-style-type: none"> 1. Найти частные производные первого порядка. 2. Построить систему уравнений. 3. Найти стационарные точки. 4. Найти частные производные второго порядка. 5. Найти значения частных производных второго порядка в первой стационарной точке ($x_1 = 0, x_2 = 0$). 6. Определить наличие экстремума в точке ($x_1 = 0, x_2 = 0$). 7. Найти значения частных производных второго порядка во второй стационарной точке ($x_1 = 1, x_2 = 1$). 8. Определить наличие экстремума в точке ($x_1 = 1, x_2 = 1$). 9. Найти минимум функции в точке ($x_1 = 1, x_2 = 1$). 10. Найти значения частных производных второго порядка во второй стационарной точке ($x_1 = -1, x_2 = -1$). 11. Определить наличие экстремума в точке ($x_1 = -1, x_2 = -1$). 12. Найти минимум функции в точке ($x_1 = -1, x_2 = -1$).

1. ПРЯМЫЕ МЕТОДЫ ОТЫСКАНИЯ ЭКСТРЕМУМА

1.1. Одномерный поиск экстремума

В юните 1 в основном рассматривались методы нахождения экстремума функционала. В разделе дискретных методов оптимизации проводился поиск оптимального решения с помощью динамического программирования и принципа максимума и рассматривались методы нахождения экстремума функции. Прямые методы применялись на каждом этапе отыскания оптимальных значений и, в частности, при решении функционального уравнения Беллмана. Далее будут рассмотрены прямые методы отыскания экстремума функций, т.е. методы, в которых используются только значения функции.

Рассмотренные ранее методы решали задачу в соответствии с определенным дифференциальным или рекуррентным соотношением, которому удовлетворяло оптимальное решение, т.е. имели аналитический характер. В прямых методах, как правило, отсутствуют такие аналитические соотношения. Прямые вариационные методы, так же как и процедуры решения задач с помощью дискретных методов динамического программирования и принципа максимума, по существу близки к прямым методам отыскания экстремума и отличаются от них только своей ориентацией на определенные аналитические уравнения.

Особенностью постановки задач, решаемых прямыми методами, является *отсутствие ограничений на изменение переменных*, характерных для линейного и нелинейного программирования (за исключением простейших ограничений вида $a_i \leq x_i \leq b_i$).

Прежде чем перейти к методам, обратим внимание на близость задач отыскания экстремума и нуля функции. Допустим, имеется m совместных уравнений:

$$\varphi_i(x_1, x_2, \dots, x_n) = 0, \quad i = 1, 2, \dots, m;$$

требуется найти x_j ($j = 1, 2, \dots, n$), удовлетворяющие им. Задачу можно свести к нахождению экстремума функции Φ , представляющей собой сумму квадратов функций φ_i , то есть

$$\Phi = \sum_{i=1}^m (\varphi_i(x))^2.$$

И наоборот, если требуется определить экстремум дифференцируемой функции $\Phi(x_1, x_2, \dots, x_n)$, эту задачу в простейшем случае можно свести к определению корней уравнений

$$\partial\Phi/\partial x_j = 0, \quad j = 1, 2, \dots, n.$$

Поиск экстремума может быть детерминированным (при отсутствии шумов) и стохастическим (при наличии случайной ошибки замеров значений функции). Различают пассивный или параллельный поиск, когда стратегия известна до получения результатов эксперимента, и активный или последовательный поиск, когда будущие стратегии уточняются в зависимости от результатов предыдущих экспериментов.

* Жирным шрифтом выделены новые понятия, которые необходимо усвоить. Знание этих понятий будет проверяться при тестировании.

Вначале ограничимся **одномерным детерминированным** случаем, когда на заданном интервале или множестве имеется одно экстремальное значение.

Будем называть функцию одной переменной $f(x)$ на интервале $[0, 1]$ **строго унимодальной** (функция, которая строго возрастает или убывает на заданном интервале), если существует x^* — точка минимума из интервала $[0, 1]$, т.е. $0 \leq x^* \leq 1$ и

$$f(x^*) = \min_{0 \leq x \leq 1} f(x)$$

и для любых двух точек $x^1, x^2 \in [0, 1]$ имеет место, что из $x^1 < x^2 \leq x^*$ следует

$$f(x^1) > f(x^2).$$

Очевидно, что всякая выпуклая функция на интервале изменения $[0, 1]$ одновременно и унимодальна. Однако обратное утверждение может быть и

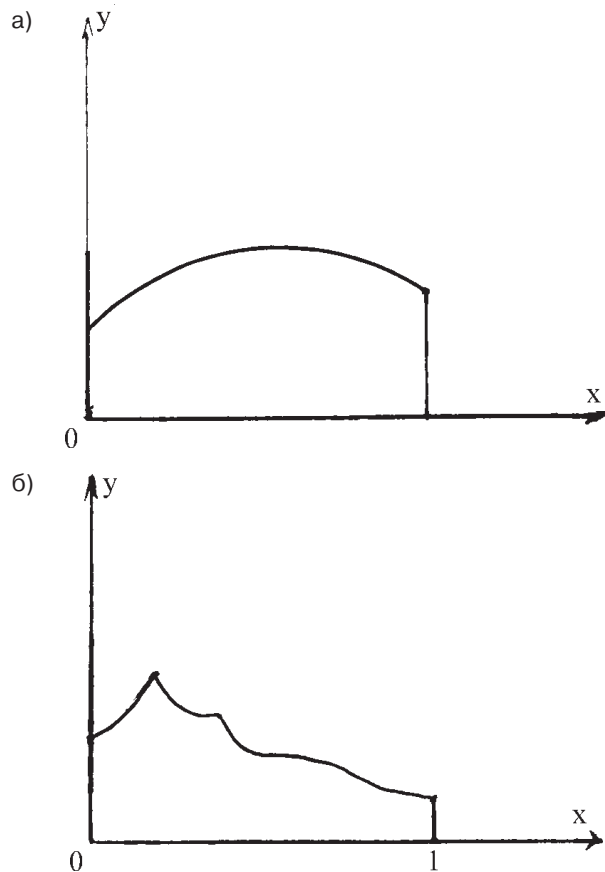


Рис. 1.1. Выпуклая (а) и невыпуклая (б) унимодальные функции

несправедливо. Если допустить разрывы непрерывности функции $f(x)$ и ее производной, то могут встретиться унимодальные на интервале $[0, 1]$ функции, не обладающие свойством выпуклости (рис. 1.1). Унимодальность обеспечивает выполнение следующего очень важного условия: если значения функции $y = f(x)$

$$y^1 = f(x^1), \quad y^2 = f(x^2)$$

взяты по одну сторону от максимума, то большему значению y соответствует более близкое к оптимуму значение x . Эта зависимость, позволяющая сформулировать критерий эффективности прямых методов поиска экстремума, показана на рис. 1.2, из которого видно, что с учетом условия унимодальности после двух экспериментов исключаются отрезки: $[x^2, 1]$ (рис. 1.2, а), $[0, x^1]$ (рис. 1.2, б), $[0, x^1]$ и $[x^2, 1]$ (рис. 1.2, в).

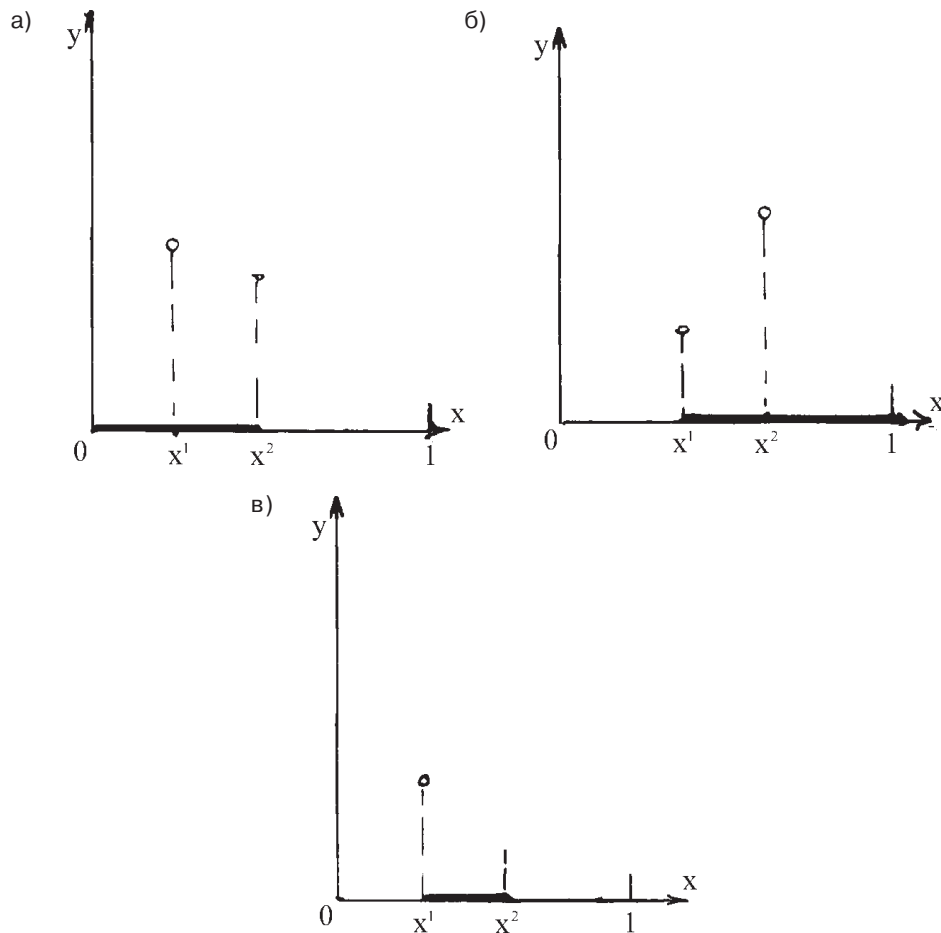


Рис. 1.2. Варианты уменьшения интервала неопределенности при двух экспериментах

Современный Гуманитарный Университет

Рассмотрим три случая с тремя экспериментами (рис. 1.3). Значения x в каждом эксперименте отстоят от левого конца исходного интервала неопределенности $[0,1]$ соответственно на величины $x^1 = 0,2$; $x^2 = 0,6$; $x^3 = 0,9$. Каждая из трех позиций рисунка содержит три возможных результата экспериментов. В каждом из трех случаев максимальное значение достигается соответственно в первом (рис. 1.3, а), втором (рис. 1.3, б) и третьем

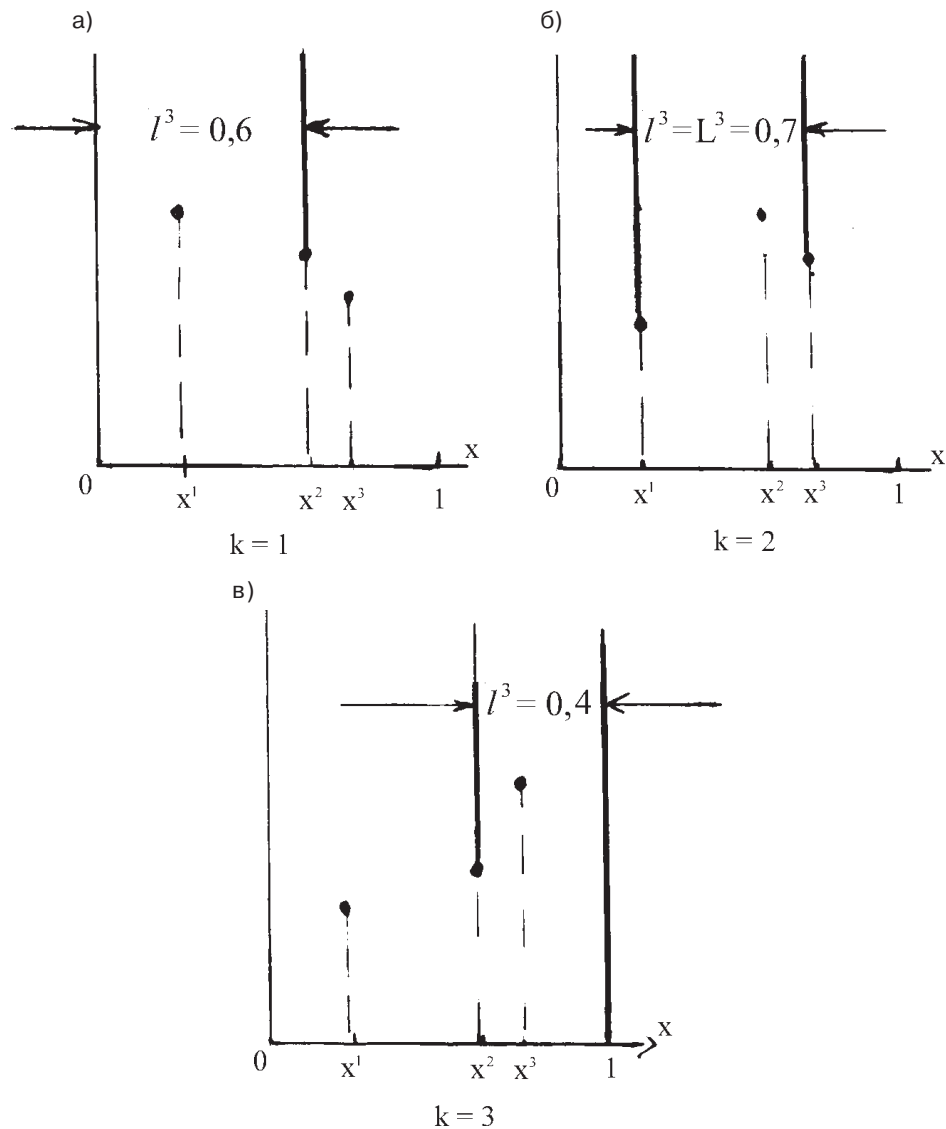


Рис. 1.3. Определение меры эффективности поиска при трех экспериментах

(рис. 1.3, в) экспериментах. Обозначим через K номер эксперимента, в котором получен наилучший в смысле максимума результат:

$$y^K = \max_{1 \leq k \leq N} \{y^k\};$$

$$y(x^K) = y^K; \quad k = 1, 2, \dots, N,$$

где N – число экспериментов. Для каждого из трех случаев можно указать новый, более узкий интервал неопределенности, в котором лежит оптимальное значение x . Действительно, при $K = 1$ $0 \leq x_{\text{опт}} < x^2$; при $K = 2$ $x^1 < x_{\text{опт}} < x^3$; при $K = 3$ $x^2 < x_{\text{опт}} \leq 1$. Можно получить общую формулу для экспериментов, если обозначить через x^0 начало исходного интервала неопределенности, т.е. $x^0 = 0$, и через $x^{N+1} = 1$ – его конец. Тогда для интервала неопределенности I^N после N экспериментов имеем

$$I^N = x^{K+1} - x^{K-1},$$

причем

$$x^{K-1} < x_{\text{опт}} < x^{K+1}.$$

В зависимости от стратегии поиска (выбора x^k) получаются разные значения интервалов $x^{k+1} - x^{k-1}$ ($k = 1, \dots, N$). Из результатов эксперимента (значений y^k) видно, при каком $k = K$ получается наилучший результат y^K и соответственно наилучшее значение интервала неопределенности I^N . Таким образом, I^N определяется, с одной стороны, распределением x^k , а с другой стороны, номером K :

$$I^N = I^N(x^k, K).$$

Эта величина является хорошей мерой эффективности поиска после завершения всей серии из N экспериментов. Но нам нужна априорная мера эффективности, а I^N до серии экспериментов неизвестна. Поэтому введем в рассмотрение наибольший из I^N (наихудший) интервал неопределенности

$$L^N(x^k) = \max_{1 \leq K \leq N} \{I^N(x^k, K)\}. \quad (1.1)$$

Нетрудно убедиться, что L^N зависит только от стратегии поиска (выбора x^k) и, конечно, от унимодальной зависимости $y(x)$ и не зависит от интервала, в котором окажется значение $x_{\text{опт}}$, т.е. величины K . Для нашего примера имеем

$$L^3(0,2; 0,6; 0,9) = \max_{1 \leq K \leq N} \{I^3(x^k, 1), I^3(x^k, 2), I^3(x^k, 3)\} =$$

$$= \max \{ (x^2 - x^0), (x^3 - x^1), (x^1 - x^2) \} = \max \{0,6; 0,7; 0,3\} = 0,7.$$

Величина L^N – единственная при заданной стратегии поиска (выборе x^k), так как среди возможных значений I^N только одно или, в крайнем случае, несколько равных между собой значений будут наибольшими. Выбор в качестве меры величины I^N для наихудшего случая избавляет нас от нежелательной зависимости от результатов испытаний и дает априорную, хотя и пессимистическую оценку эффективности поиска. При реальном поиске может получиться лучший результат.

После того как введена мера эффективности поиска, можно указать

критерий выбора оптимального поиска, в качестве которого берется минимум величин:

$$L_{\text{опт}}^N = \min_{x^k} \{ L^N(x^k) \}. \quad (1.2)$$

Оптимальная стратегия $x_{\text{опт}}^k$ определяется соотношением

$$L^N(x_{\text{опт}}^k) = L_{\text{опт}}^N. \quad (1.3)$$

Величина $L_{\text{опт}}^N$ – постоянная и единственная для данной унимодальной зависимости. Она определяет минимальный интервал неопределенности, хотя оптимальных стратегий $x_{\text{опт}}^k$ может быть несколько. Стратегия $x_{\text{опт}}^k$ может быть названа минимаксной, так как

$$L_{\text{опт}}^N = \min_{x^k} \max_{1 \leq K \leq N} \{ L^N(x^k, K) \}.$$

Здесь

$$\min_{x^k} \max_{1 \leq K \leq N} \{ L^N(x^k, K) \} = 1;$$

$$\min_{1 \leq K \leq N} \{ L^N(x^k, K) \} = 0.$$

Не во всех ситуациях можно реализовать последовательную процедуру поиска. Так, при управлении доменным или химическим процессом, когда нет времени на последовательный анализ, делается одновременный параллельный замер параметров. При этом возникает необходимость определения новых значений управляющих параметров, которые обеспечили бы оптимальную близость к экстремуму.

Или, например, группа из нескольких экономистов должна решить задачу оптимального распределения ресурсов в условиях ограниченного заказчикам времени (допустим, одна неделя). Такое условие заставляет построить работу по параллельной схеме, т.е. каждый экономист ведет поиск в отдельности по заданной ему программе распределения ресурсов. Если бы заказчик не торопил с решением вопроса и выделил вместо одной недели четыре, то десять экономистов могли бы совместно последовательным поиском рассчитать четыре варианта и получить ту же точность, что и при пассивном поиске при расчете десяти вариантов. Если бы в их распоряжении было десять недель, то совместная работа по последовательной схеме дала бы эффективность в 18 раз большую, чем при пассивном поиске. В этой модели условно считается, что при совместной работе десяти экономистов по последовательной схеме время расчета одного варианта остается равным неделе. Однако можно составить другую модель, предположив, что при совместной работе время расчета одного варианта сокращается в 10 раз. Тогда уже к концу второго дня получился бы тот же самый результат, что и при пассивном поиске к концу недели (считается, что рабочих дней в неделе пять).

Для случая двух экспериментов пассивная стратегия ничем не отличается от активной, так как информацию, полученную от одного эксперимента, нельзя использовать для уточнения второго эксперимента. Пусть имеются результаты двух экспериментов ($N = 2$): x^1, x^2 , причем $0 \leq x^1 < x^2 \leq 1$. С помощью соотношений (1.1) – (1.3) можно написать

$$L^2 = \max \{ (x^2 - x^0), (x^3 - x^1) \} = \max \{ x^2, (1 - x^1) \}.$$

На рис. 1.4 показаны кривые $L^2 = \text{const}$ в зависимости от $x^1 = x_1$ и $x^2 = x_2$. Из-за условия $x^2 > x^1$ оптимальное на первый взгляд значение $L^2 = 0,5$ при $x^1 = x^2 = 0,5$ отвергается. Поэтому вводится минимальная величина ε , которая определяется погрешностью измерительной аппаратуры, позволяющей при различии между x^1 и x^2 , равном ε , обнаружить разницу между соответствующими y^1 и y^2 . Тогда в результате экспериментов $x^1 = 0,5 - \varepsilon/2$ и $x^2 = 0,5 + \varepsilon/2$ можно получить не только требуемое разделение y^1 и y^2 , но и минимально возможное значение $L^2_{\text{опт}}$

$$L^2_{\text{опт}} = L^2 [(0,5 - \varepsilon/2), (0,5 + \varepsilon/2)] = 0,5 + \varepsilon/2.$$

Только что рассмотренная стратегия носит наименование ε -минимаксной.

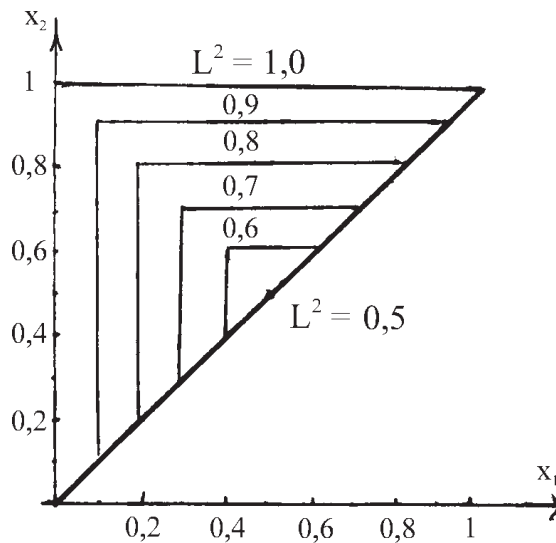


Рис. 1.4. Определение эффективности пассивного поиска при двух экспериментах

Теперь рассмотрим случай трех экспериментов и постараемся улучшить довольно плохой план, ранее рассмотренный и представленный на рис. 1.3 ($x^1 = 0,2$; $x^2 = 0,6$; $x^3 = 0,9$). Ранее было получено максимальное значение интервала неопределенности, равное 0,7. Из рис. 1.3, б видно, что можно уменьшить L^3 , сблизив x^1 и x^3 , в связи с чем, уменьшив x^3 и увеличив x^1 на 0,1, получим (рис. 1.5):

$$x^1 = 0,3; x^2 = 0,6; x^3 = 0,8; x^3 - x^1 = 0,5.$$

При этом

$$L^3 = \{0,6; 0,5; 0,4\} = 0,6 = x^2 - x^0 = x^2.$$

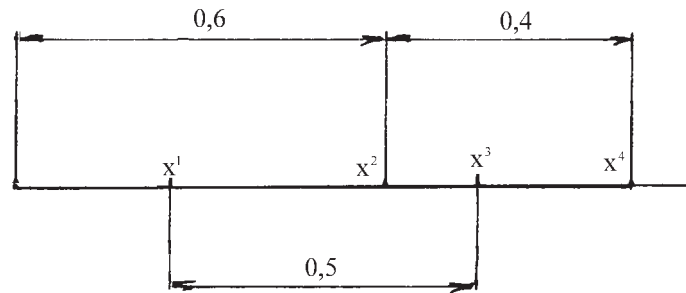


Рис. 1.5. Улучшенный вариант пассивного поиска при трех экспериментах

Теперь, перемещая x^2 влево, можно еще уменьшить L^3 , однако одновременно будет увеличиваться $x^4 - x^2 = 1 - x^2$. Поэтому для выбора x^2 следует рассмотреть график функции

$$\max \{x^2, 1 - x^2\} \quad (1.4)$$

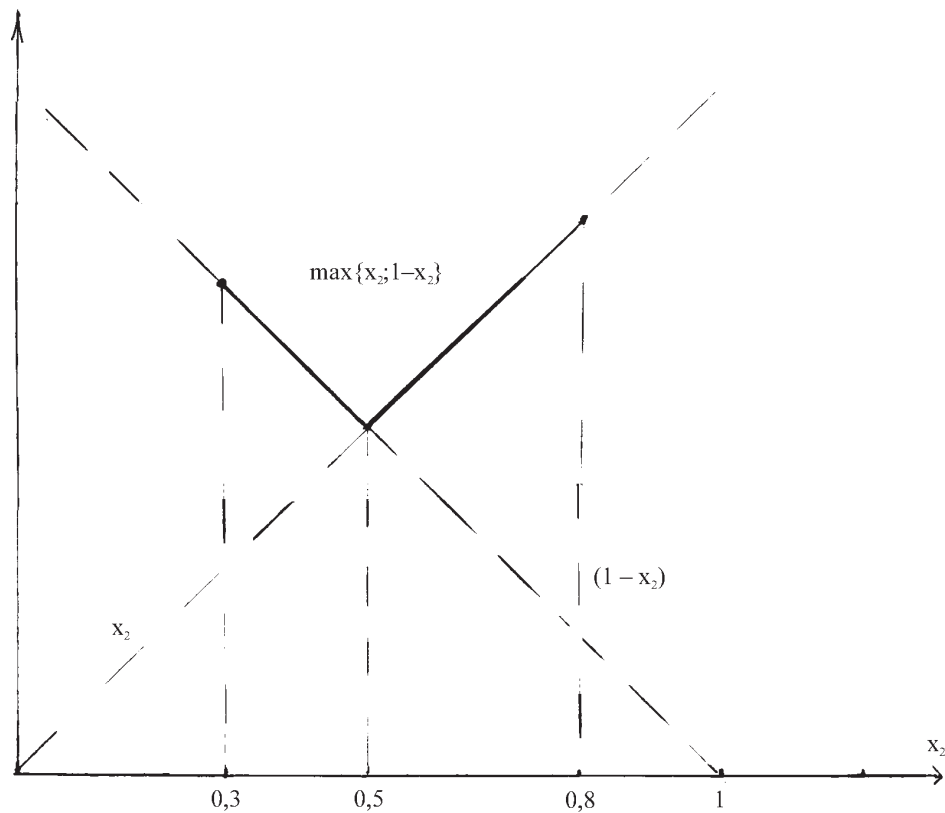


Рис. 1.6. Определение эффективности пассивного поиска при трех экспериментах

в интервале $0,3 = x^1 < x^2 < x^3 = 0,8$. В соответствии с рис. 1.6 минимальное значение функции (1.4) получается при $x_2 = x^2 = 0,5$. Это значение и выбирается в оптимальной минимаксной стратегии. Таким образом, оптимальной стратегией будет всякая пассивная стратегия (x^1, x^2, x^3) , для которой $x^2 = 0,5$ (рис. 1.7, а) и $x^3 - x^1 \leq 0,5$ (рис. 1.7, б) и которая дает оптимальное значение $L^3_{\text{опт}} = 0,5$.

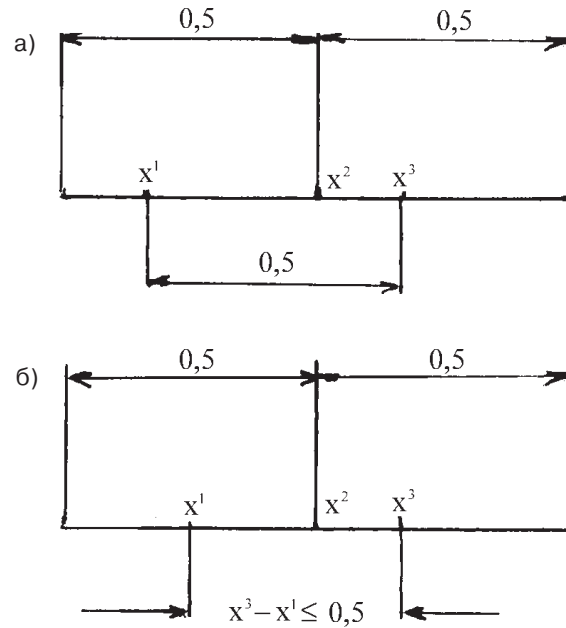


Рис. 1.7. Оптимальные пассивные стратегии при трех экспериментах

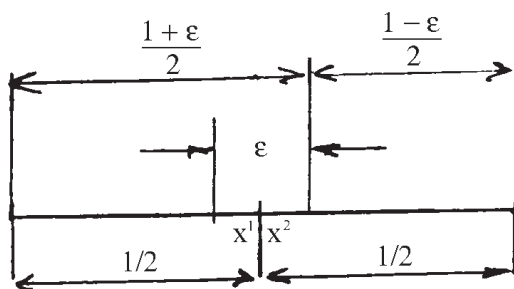
Определение минимаксной стратегии с помощью рис. 1.7 практически совпадает с геометрическим методом отыскания оптимальной стратегии в теории игр.

Из рассмотренных двух примеров следует, что добавление третьего эксперимента, требующего дополнительных средств и усилий, в случае пассивного поиска мало себя оправдывает, так как разность $L^3_{\text{опт}} - L^2_{\text{опт}}$ составляет малую величину $\epsilon/2$. Это положение остается справедливым для любого числа опытов N , т.е. использование нечетного числа опытов целесообразно только при большой погрешности измерений ϵ , поэтому часто на практике используют поиск парами экспериментов. Можно показать, что наилучший выбор получается при разделении экспериментальных точек на равноотстоящие пары. При таком распределении процедура носит название поиска однородными парами. Так, при $n = 2$ (рис. 1.8, а) эксперимент состоит из двух значений:

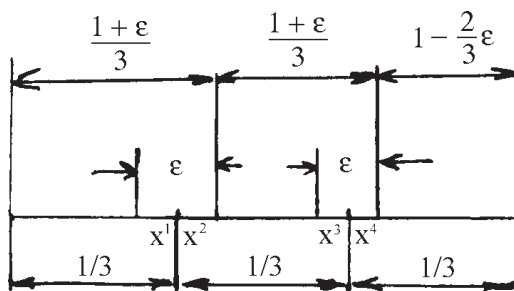
$$x^1 = (1 - \epsilon)/2, \quad x^2 = (1 + \epsilon)/2;$$

при $n = 4$ (рис. 1.8, б) – из четырех значений:

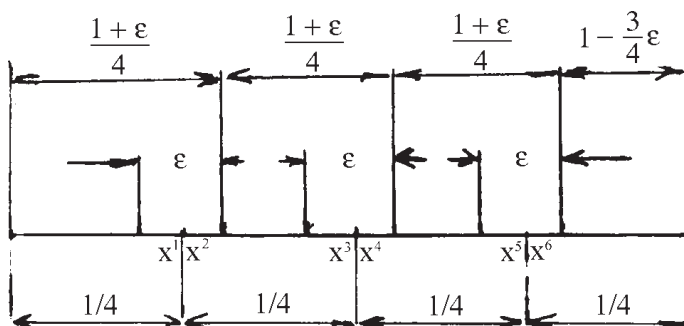
$$x^1 = (1 - \epsilon)/2, \quad x^2 = (1 + \epsilon)/2, \quad x^3 = (2 - 2\epsilon)/3 - \epsilon, \quad x^4 = (2 + 2\epsilon)/3 \text{ и т.д.}$$



а) $N = 2$ (одна пара)



б) $N = 4$ (две пары)



в) $N = 6$ (три пары)

Рис. 1.8. Поиск однородными парами

Аналитически значения при поиске однородными парами можно задать с помощью формулы

$$x^k = \{(1 + \varepsilon)[(k+1)/2]\} / \{(N/2) + 1\} - \{[(k+1)/2] - [k/2]\}\varepsilon,$$

где квадратные скобки $[a]$ обозначают наибольшее целое число, не превышающее a : $[\pi] = 3$, $[4] = 4$ и т. д.

Из вышеизложенного следует, что такая схема для четного числа опытов представляет ε -минимаксную стратегию, так как смещение любого из экспериментов только увеличивает L^N . Величина оптимального интервала неопределенности задается формулой

$$L_{\text{опт}}^N = (1 + \varepsilon) / [(N/2) + 1]. \quad (1.5)$$

Для числа опытов $N + 1$

$$L_{\text{опт}}^{N+1} = L_{\text{опт}}^N + \varepsilon / [(N/2) + 1],$$

где N – четное, т.е. с увеличением числа опытов разница между оптимальными интервалами неопределенности при четном и нечетном числе опытов уменьшается примерно обратно пропорционально числу опытов.

1.2. Последовательный поиск экстремума

При **последовательном поиске** анализируются результаты предыдущего эксперимента и в зависимости от них ставится следующий эксперимент. При этом получается большой выигрыш эффективности поиска. Поэтому на практике, если позволяют условия, следует отдавать предпочтение последовательному поиску перед параллельным (пассивным).

Организация последовательного поиска иногда требует разработки специальных технических устройств для последовательных замеров. Так, одна подвижная термopapa позволяет активно следить за высокотемпературной зоной химического реактора, т.е. осуществляет последовательный поиск, а несколько неподвижно установленных термopap позволяют осуществлять лишь пассивный поиск.

Самым естественным и наиболее распространенным на практике является метод поиска экстремума путем последовательного деления отрезка пополам. Этот метод, известный еще в Древней Греции, назван **методом дихотомии** (dicha – на две части + tome – сечение).

Ранее было показано, что при двух экспериментах (первая пара на рис. 1.9) лучше всего выбирать точки в середине интервала – возможно ближе друг к другу. Тогда длина интервала неопределенности будет равна $0,5 + \varepsilon/2$. В третьем и четвертом экспериментах (вторая пара) точки выбираются вблизи середины получившегося интервала. Новый интервал неопределенности при этом равен

$$1/2(1/2 + \varepsilon/2) + \varepsilon/2 = 1/4 + (3/4)\varepsilon.$$

После шести опытов (третья пара) интервал неопределенности составит

$$1/8 + (7/8)\varepsilon.$$

После N опытов, где N – четное и конечное число, интервал неопределенности запишется как

$$L_{\text{опт}}^N = 2^{-N/2} + (1 - 2^{-N/2})\varepsilon.$$

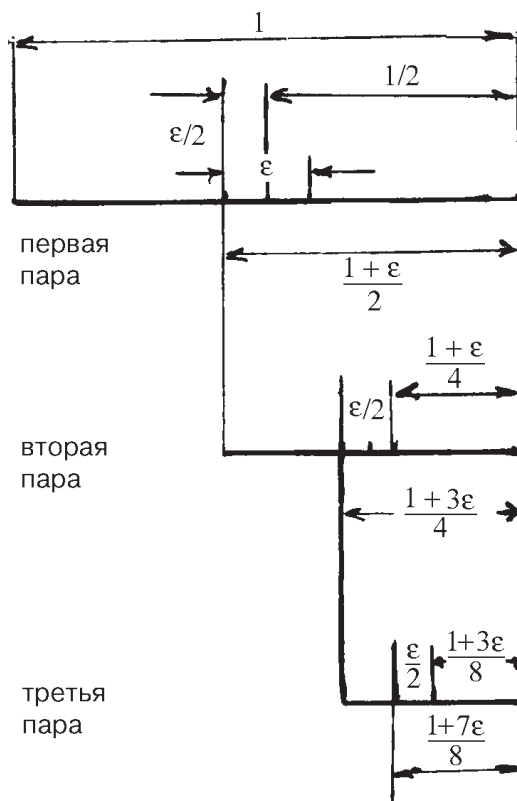


Рис. 1.9. Метод дихотомии

Отсюда следует, что **эффективность поиска при методе дихотомии** (оценка точности поиска по аргументу) растет с ростом N экспоненциально, тогда как при пассивном поиске однородными парами эффективность растет лишь прямо пропорционально – см. формулу (1.5).

Метод Фибоначчи своими корнями восходит к работам математика XIII века Фибоначчи и даже к более ранним исследованиям Евклида.

Будем считать, что для исследования исходного интервала неопределенности единичной длины было выделено N экспериментов, $N - 1$ из которых выполнены. В последнем эксперименте требуется исследовать интервал неопределенности $L_{\text{опт}}^{N-1}$, держащий точку-эксперимент x^{N-1}_+ , для которой получено самое большое значение y при первых $N - 1$ испытаниях (рис. 1.10). Положение точки x^{N-1}_+ целиком зависит от стратегии поиска. Последнюю точку x^N можно выбрать произвольно, но не ближе, чем на расстоянии ε от точки x^{N-1} . Ситуация на этом последнем шаге полностью совпадает с обычным поиском при наличии двух экспериментов. Минимальное значение последнего интервала $L_{\text{опт}}^N$ получается, если точки x^{N-1} и x^N расположить симметрично на расстоянии $\varepsilon/2$ относительно середины интервала $L_{\text{опт}}^{N-1}$. Поэтому значения $L_{\text{опт}}^{N-1}$ и $L_{\text{опт}}^N$ связаны соотношением

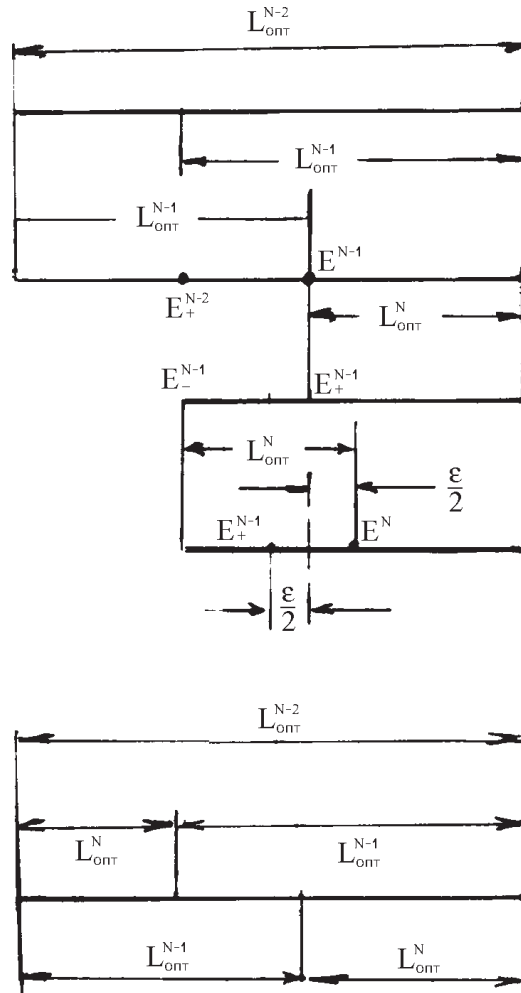


Рис. 1.10. Метод Фибоначчи

$$L^{N-1}_{\text{опт}} = 2 L^N_{\text{опт}} - \epsilon. \quad (1.6)$$

Продолжая процедуру движения с конца, аналогичную динамическому программированию, введем в рассмотрение интервал неопределенности $L^{N-2}_{\text{опт}}$, который получается в результате $N - 2$ экспериментов из общего числа N . Внутри этого интервала содержится эксперимент x^{N-2} , дающий наибольшее значение y при первых $N - 2$ экспериментах. В этом интервале следует произвести следующую пару экспериментов. Лучшее значение из двух обозначим через x^{N-1}_+ . Другая точка из этой пары, которую обозначим через x^{N-1}_- , будет граничной между частью $L^{N-2}_{\text{опт}}$ интервала $L^N_{\text{опт}}$ и частью $L^{N-1}_{\text{опт}}$.

оставляемой для рассмотрения. Но в начале эксперимента неизвестно, какое из двух значений x^{N-1} будет наибольшим и обозначится как x^{N-1}_{+} , а какое наименьшим и обозначится как x^{N-1}_{-} . Эти два значения могут меняться местами. Поэтому обе точки x^{N-1}_{+} и x^{N-1}_{-} располагаются на одинаковом расстоянии $L^{N-1}_{\text{опт}}$ от концов интервала. Так как эти точки расположены симметрично относительно середины интервала $L^{N-2}_{\text{опт}}$ и одна из них становится x^{N-1}_{+} (причем это может быть любая из двух точек x^{N-1}), то каждая из них должна находиться на расстоянии $L^N_{\text{опт}}$ от одного конца интервала и на расстоянии $L^{N-1}_{\text{опт}}$ от другого конца интервала. При этом, учитывая формулу (1.6), напомним:

$$L^{N-2}_{\text{опт}} = L^{N-1}_{\text{опт}} + L^N_{\text{опт}}$$

и

$$L^{N-2}_{\text{опт}} = (2L^N_{\text{опт}} - \varepsilon) + L^N_{\text{опт}} = 3L^N_{\text{опт}} - \varepsilon.$$

Аналогично для любого $1 < k < N$:

$$L^{k-1}_{\text{опт}} = L^k_{\text{опт}} + L^{k+1}_{\text{опт}}.$$

Отсюда при $k = N - 2$

$$L^{N-3}_{\text{опт}} = L^{N-2}_{\text{опт}} + L^{N-1}_{\text{опт}} = (3L^N_{\text{опт}} - \varepsilon) + (2L^N_{\text{опт}} - \varepsilon) = 5L^N_{\text{опт}} - 2\varepsilon;$$

$$L^{N-4}_{\text{опт}} = 8L^N_{\text{опт}} - 3\varepsilon;$$

$$L^{N-5}_{\text{опт}} = 13L^N_{\text{опт}} - 5\varepsilon.$$

Для удобства записи формул введем последовательность чисел Фибоначчи:

$$\begin{aligned} F_0 = F_1 &= 1 \\ F_k &= F_{k-1} + F_{k-2} \quad k = 2, 3, \dots, \end{aligned}$$

в соответствии с которой имеем:

$$F_2 = 2, F_3 = 3, F_4 = 5, F_5 = 8 \text{ и т.д.}$$

С помощью этих чисел можно записать следующую формулу для оптимальных интервалов неопределенности:

$$L^{N-k}_{\text{опт}} = F_{k+1} L^N_{\text{опт}} - F_k \varepsilon. \quad (1.7)$$

Полагая длину исходного интервала равной единице, получаем:

$$N - k = 1; \quad k = N - 1;$$

$$L^1_{\text{опт}} = 1 = F_N L^N_{\text{опт}} - F_{N-2} \varepsilon.$$

Отсюда находим оптимальный интервал после N опытов:

$$L^N_{\text{опт}} = 1/F_N - (F_{N-2}/F_N)\varepsilon. \quad (1.8)$$

Как видно из приведенной ниже табл. 1.1, для уменьшения интервала

неопределенности более чем в 100 раз по методу Фибоначчи необходимо провести 11 опытов, по методу дихотомии – 14, а при пассивном методе – 198 [см. формулу (1.5)]. При **методе Фибоначчи** каждая последующая точка выбирается симметрично по отношению к точке, которая осталась от предыдущего эксперимента и попала в оставшийся интервал. Первый эксперимент следует выбирать на расстоянии $L_{\text{опт}}^2$ от одного из концов начального интервала, безразлично от левого или правого, в силу симметрии. Чтобы получить точное выражение для $L_{\text{опт}}^2$, воспользуемся формулой (1.7), учитывая, что $N - k = 2$ и $k = N - 2$:

$$L_{\text{опт}}^2 = F_{N-1} L_{\text{опт}}^N - F_{N-3} \varepsilon.$$

Исключив из этого соотношения $L_{\text{опт}}^N$, с помощью формулы (1.8) получим:

$$L_{\text{опт}}^2 = F_{N-1} / F_N + [(F_{N-1} F_{N-3} - F_N F_{N-3}) / F_N] \varepsilon. \quad (1.9)$$

Используя тождества $F_{N-1} \equiv F_{N-2} + F_{N-3}$ и $F_N \equiv F_{N-1} + F_{N-2}$, которые следуют из самого определения чисел Фибоначчи, напомним:

$$F_{N-1} F_{N-2} - F_N F_{N-3} = (F_{N-2} + F_{N-3}) F_{N-3} - (F_{N-1} + F_{N-2}) F_{N-3} = F_{N-2}^2 - F_{N-1} F_{N-3}. \quad (1.10)$$

Доказано, что правая часть этого равенства равна $(-1)^N$, поэтому формулу (1.9) можно переписать в виде

$$L_{\text{опт}}^2 = F_{N-2} / F_N + (-1)^N \varepsilon / F_N. \quad (1.11)$$

Формула (1.11) позволяет начать процедуру поиска, если известны ε и N . В этом состоит недостаток метода: требуется заранее знать необходимое число опытов. Очень часто задается только величина интервала неопределенности, которую надо знать к концу N -го опыта. Эта особенность придает последовательному методу Фибоначчи свойства пассивного поиска. Иногда поэтому обращаются к методу дихотомии, который хотя и обладает меньшей эффективностью по сравнению с методом Фибоначчи, но не требует предварительного знания числа опытов.

Однако существует другой метод поиска, который почти так же эффективен, как метод Фибоначчи, но не требует априорного задания числа опытов N . Имеется в виду известный издавна метод золотого сечения (рис. 1.11).

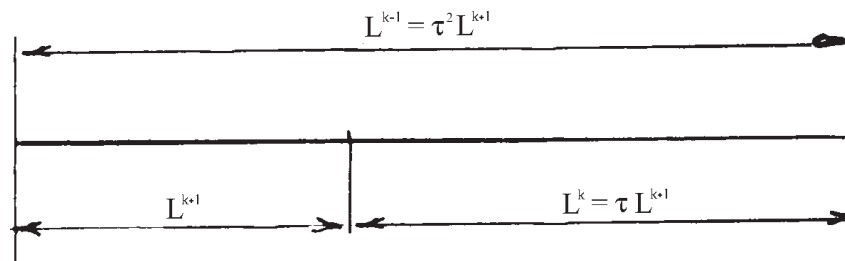


Рис. 1.11. Метод золотого сечения

Чтобы избавиться от зависимости первого опыта от числа опытов, поступают следующим образом. По-прежнему считаем, что сохраняется условие симметричности последовательных экспериментов, так же как в методе Фибоначчи, т.е. справедливо соотношение

$$L^{k-1} = L^k + L^{k+1}. \quad (1.12)$$

Однако вместо условия

$$L^{N-1} = 2L^N - \varepsilon,$$

при котором в методе Фибоначчи интервал $L^2_{\text{опт}}$ зависит от N, предлагается условие

$$L^{k-1} / L^k = L^k / L^{k+1} = \tau,$$

означающее постоянство отношения длин последовательных отрезков. Отсюда следует, что

$$L^{k-1} / L^{k+1} = \tau^2. \quad (1.13)$$

Учитывая соотношение (1.13) и поделив обе части (1.12) на L^{k+1} , получим

$$\tau^2 = \tau + 1.$$

Положительный корень этого квадратного уравнения

$$\tau = (1 + \sqrt{5})/2 = 1,6180.$$

По результатам двух экспериментов выбирают часть исходного отрезка, внутри которого расположен один из предыдущих опытов. В следующую пару опытов включают эту точку, а вторую выбирают внутри остающегося отрезка симметрично точке, оставшейся от предыдущего опыта. Первые точки найдутся на расстоянии $1/\tau$ от левого и правого концов единичного начального интервала. Процесс можно продолжать как угодно долго. После N опытов интервал неопределенности достигнет

$$L^N = 1/t^{N-1}. \quad (1.14)$$

Рассмотренная процедура носит название **метода золотого сечения**. Она состоит в делении отрезка на две неравные части так, чтобы отношение всего отрезка к большей части равнялось отношению большей части отрезка к меньшей (см. рис. 1.11). Существует уравнение, связывающее числа Фибоначчи F_N и величину τ :

$$F_N = [\tau^{N+1} - (-\tau)^{-(N+1)}] / \sqrt{5}.$$

При больших N вторым членом в этой формуле можно пренебречь, в результате чего получим:

$$F_N \approx \tau^{N+1} / \sqrt{5}.$$

Это соотношение дает возможность получить связь интервалов неопределенности L^N после N опытов при методе золотого сечения и при методе Фибоначчи:

$$L^N / L_{\text{опт}}^N = \tau^{N+1} / (\sqrt{5}) \tau^{N-1} = \tau^2 / \sqrt{5} \approx 1,17, \quad (1.15)$$

т.е. эффективность метода золотого сечения только на 17% ниже метода Фибоначчи. При больших N , как это следует из уравнения

$$F_N \approx \tau^{N+1} / \sqrt{5},$$

имеем

$$F_{N-1} / F_N \approx 1 / \tau.$$

Отсюда с учетом формул (1.10) и (1.14) для больших N

$$L_{\text{опт}}^2 \approx 1 / \tau = L^2,$$

т.е. при больших N обе процедуры начинают поиск почти в одних и тех же точках.

В табл. 1.1 приведены цифры, характеризующие уменьшение исходного единичного интервала неопределенности для различных методов поиска.

Все ранее рассмотренные методы исходили из того, что величина x^k

Таблица 1.1

Число опытов	Сокращение интервала неопределенности			
	Метод дихотомии	Метод Фибоначчи	Метод золотого сечения	Поиск по дискретным точкам
0	1	1	-	0
1	1	1	1	1
2	2	2	1,62	2
3	2	3	2,62	4
4	4	5	4,24	7
5	4	8	6,85	12
6	8	13	11,09	20
7	8	21	17,94	33
8	16	34	29,0	54
9	16	55	47,0	88
10	32	89	76,0	143

может принимать любое значение внутри интервала $[0,1]$ и интервал неопределенности может изменяться непрерывно. Есть задачи, когда требуется осуществлять выбор на конечном дискретном наборе значений x^k . Например, требуется поделить число квартир или составить сборную страны по футболу из игроков команд первой группы, – число комбинаций здесь всегда конечное. Иногда непрерывный поиск искусственно превращают в **поиск по дискретным точкам**, так как последний выделяет одну точку экстремума, а не интервал неопределенности, который неудобен для решения задач, имеющих дело с конкретными понятиями.

Допустим, задано s дискретных точек: x_1, x_2, \dots, x_s , значения которых необязательно должны быть целыми числами. Например, среднее число ящиков, которое загружается в единицу времени, может принимать значения: 3,0; 3,4; 3,6; 4,2; 4,4; 4,6; 5,0. Сопоставим каждому дискретному значению целые числа:

Число ящиков	3,0	3,4	3,6	4,2	4,4	4,6	5,0
Номер варианта	1	2	3	4	5	6	7

В интервале от нуля до единицы расположим равномерно k точек (рис. 1.12). Для данного примера число $k + 1$ равно числу Фибоначчи $F_5 = 8$, исходный интервал неопределенности отличается от единицы и равен $L^1 = F_N$ ($N = 5$). Поэтому два первых эксперимента отстоят от концов исходного интервала неопределенности на величины $F_N L^2_{\text{опт}}$ где

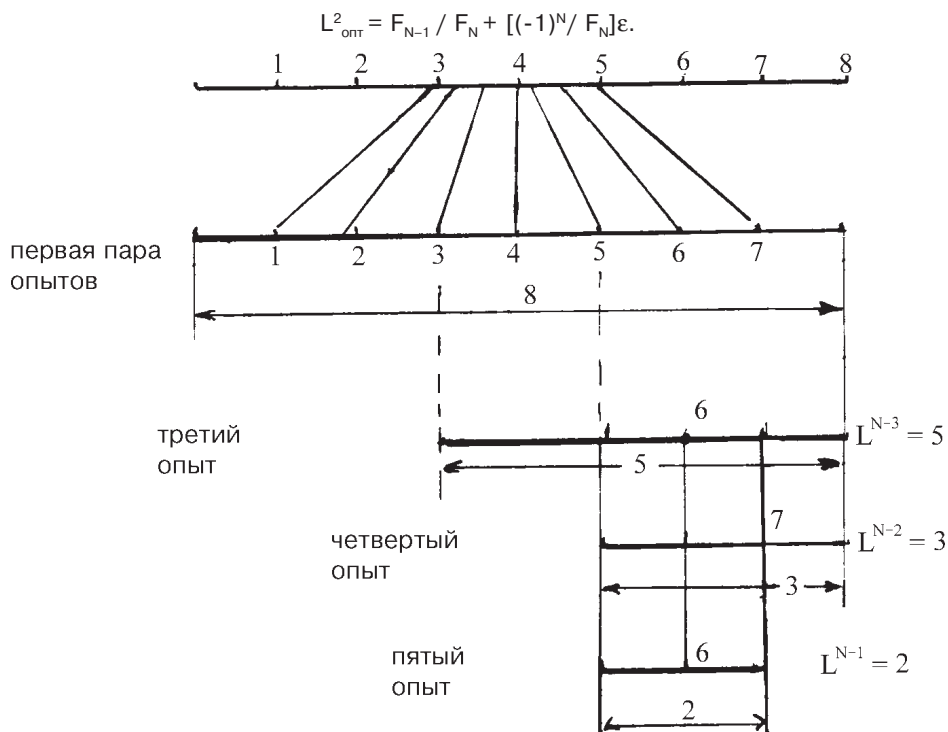


Рис. 1.12. Метод поиска по дискретным точкам

Полагая $\varepsilon = 0$, получаем

$$F_N L_{\text{опт}}^2 = F_{N-1}.$$

В силу этого соотношения первые два эксперимента совпадут с дискретными точками. В данном примере $F_{N-1} = F_4 = 5$ и точками первых двух экспериментов будут точки 3 и 5, отстоящие от концов интервала на величину $F_4 = 5$. Третий эксперимент попадает в дискретную точку 6, четвертый – в точку 7, пятый и последний – в точку 6. Процесс закончится, когда длина интервала неопределенности станет равной единице. Отсюда в соответствии с формулой (1.8) при $\varepsilon = 0$ имеем

$$F_N L_{\text{опт}}^N = 1.$$

Следует специально остановиться на условии $\varepsilon = 0$, благодаря которому последний эксперимент попадает в точку, где уже производился один из предыдущих экспериментов. Это позволяет не производить последнего эксперимента. Действительно, интервал неопределенности $F_N L_{\text{опт}}^{N-1}$, остающийся после $N-1$ шагов, в соответствии с формулой (1.15) равен двум. Причем, как видно из рис. 1.12, концы этого интервала уже испытаны и отвергнуты, поэтому оптимумом должна быть середина интервала.

Напомним, что в методе Фибоначчи ε появилось для выполнения последней пары экспериментов: $(N-1)$ -го и N -го, которые пришлось расположить в силу непрерывности x на расстоянии $\varepsilon/2$ от середины интервала $L_{\text{опт}}^{N-1}$. Именно поэтому и из требования различения двух последних экспериментов по x возникла необходимость введения ε .

Если обозначить через s_{N-1} максимальное число точек, которые после $N-1$ экспериментов можно свести к единственному экстремуму, то очевидно, что $s_{N-1} = F_N - 1$, так как в интервале длиной F_N , содержащем F_{N-1} точек, можно по методу Фибоначчи найти оптимальную точку после $N - 1$ экспериментов. Иначе эту формулу можно переписать как $s_N = F_{N+1} - 1$. Сравнительные значения этой величины приведены в табл. 1.1.

1.3. Метод рандомизации

В предыдущем разделе было показано, что метод Фибоначчи может применяться при поиске на дискретном множестве точек. Если число исходных точек сильно отличается от ближайшего, строго большего числа Фибоначчи, то приходится добавлять фиктивные точки и тем самым увеличивать количество экспериментов. Так, при 8 опытах добавляется еще 5, т.е. общее число точек доводится до ближайшего, строго большего числа Фибоначчи, 13. Но можно ввести элемент случайности и выбирать экспериментальные точки в соответствии с определенным законом распределения (например, с помощью бросания монеты или кости). При этом в среднем можно получить выигрыш в числе точек эксперимента по сравнению с детерминированным поиском по методу Фибоначчи. В этом и заключается **метод рандомизации** (выбор экспериментальных точек в соответствии с определенным законом распределения случайности). Процедура поиска экстремума основывается на определенной вероятностной модели расположения экстремума. Считается, что природа, как партнер в игре, предлагает для поиска определенный вероятностный закон распределения экстремума. Стратегия строится вероятностным образом, так как экстремум с определенной вероятностью может появиться в любом месте. Такой подход по существу совпадает с

игровым методом и процедурами проверки статистических гипотез.

Для простоты рассмотрим случай унимодальной функции, заданной в трех точках: 1, 2, 3. Все варианты проведения эксперимента можно разбить на три: (1, 2), (1, 3) и (2, 3). Если экстремум окажется в точке 1, то при выборе стратегии (1,2) (т.е. первые два эксперимента производятся в точках 1 и 2), третьего эксперимента можно не делать. Двух экспериментов окажется достаточно, если экстремум находится в точке 3 и было принято решение провести эксперименты в точках (2, 3). Во всех остальных случаях требуется провести три эксперимента. Окончательные данные о числе необходимых опытов по всем вариантам сведены в табл. 1.2.

Таблица 1.2

Место проведения первых двух опытов	Положение экстремума		
	1	2	3
(1,2)	2	3	3
(1,3)	3	3	3
(2,3)	3	3	2

Эта таблица по существу представляет собой игровую матрицу. Платежом в такой игре является число необходимых экспериментов N . Матрица получилась не вполне удачной, так как в ней отсутствует седловая точка, поэтому для решения данной игровой задачи необходимы специальные приемы. Можно сразу исключить стратегию, состоящую в выборе второй пары точек (1, 3), так как в этом случае при любом положении экстремума заведомо получаются три эксперимента.

Для упрощения задачи вначале предположим, что во второй точке максимума быть не может. Остается установить вероятности выбора стратегий (1, 2) и (2, 3). Так как нет никаких оснований для предпочтения одной стратегии перед другой, положим вероятность выбора каждой из них, равную половине. Докажем это положение следующим образом. Будем считать, что из общего числа опытов n экстремум в точке 1 встречается n_1 раз. С помощью датчика случайных чисел выберем стратегию (1, 2) $n_1 p$ раз и ограничимся двумя опытами. В остальных $n_1(1-p)$ случаях выберем стратегию (2, 3) и ограничимся тремя опытами. Величину p найдем исходя из минимаксного критерия. Основываясь на приведенных рассуждениях, считаем, что требуется произвести $2 n_1 p + 3 n_1(1-p)$ опытов. Разделив эту величину на n_1 , получим среднее число опытов e_1 , которые необходимо проделать при условии, что экстремум действительно находится в точке 1:

$$e_1 = 2p + 3(1 - p) = 3 - p.$$

Аналогично, если обозначим через n_3 число случаев нахождения оптимума в точке 3, то для необходимого количества опытов получим выражение $3 n_3 p + 2 n_3(1 - p)$, а для среднего числа опытов формулу

$$e_3 = 3p + 2(1 - p) = 2 + p.$$

Для удовлетворения минимаксного критерия требуется найти такое значение $p = p_{\text{опт}}$, при котором минимизируется функция

$$\max \{(3-p), (2+p)\}.$$

Минимальное значение, равное 2,5, для этого выражения достигается при $p_{\text{опт}} = 0,5$.

Теперь учтем возможность нахождения экстремума в точке 2. Для этого введем в рассмотрение относительную частоту f_i , с которой экстремум появляется в точке i :

$$f_i = n_i / (n_1 + n_2 + n_3), \quad i = 1, 2, 3.$$

Среднее число экспериментов при минимаксной рандомизированной стратегии определится как

$$e_{\text{опт}} = 2,5(1 - f_2) + 3f_2 = 2,5 + 0,5 f_2. \quad (1.16)$$

Эта величина не зависит явно от f_1 и f_3 , т.е. если природа или противник выбирает наихудшую для нас стратегию (частоты f_1 и f_3), среднее число опытов остается неизменным при неизменности f_2 .

Теперь отойдем от рандомизированной стратегии и будем считать, что относительные частоты f_i известны заранее. Тогда, учитывая, что

$$f_2 = 1 - f_1 - f_3, \quad (1.17)$$

для среднего числа экспериментов напомним:

$$e = (3 - p)f_1 + 3(1 - f_1 - f_3) + (2 + p)f_3 = 3 - f_3 + p(f_3 - f_1).$$

Отсюда, если $f_1 > f_3$, минимум e достигается при $p = 1$; если $f_1 = f_3$, значение p не оказывает влияния на e ; если $f_1 < f_3$, минимум e достигается при $p = 0$. Поэтому оптимум будет иметь место для

$$\begin{aligned} p &= 0 \text{ при } f_1 < f_3, \\ p &= 1 \text{ при } f_1 > f_3, \end{aligned} \quad (1.18)$$

чему соответствует

$$\begin{aligned} e &= 3 - f_3 \text{ при } f_1 < f_3, \\ e &= 3 - f_1 \text{ при } f_1 > f_3. \end{aligned} \quad (1.19)$$

Если подставить выражение (1.17) в (1.16), то

$$e_{\text{опт}} = 3 - 0,5 (f_1 + f_3). \quad (1.20)$$

С помощью формул (1.19) и (1.20) можно получить выражение для выигрыша при детерминированном поиске, когда по сравнению с рандомизированным поиском заранее известны частоты f_i :

$$e_{\text{опт}} - e = 0,5 |f_1 - f_3| \geq 0.$$

Из формулы (1.18) видно, что при известных f_1 и f_3 стратегия становится детерминированной (чистой). Если эти частоты неизвестны заранее, то применяется смешанная стратегия.

1.4. Методы многомерного поиска экстремума

1.4.1. Градиентный метод

Многомерный поиск экстремума проводится, когда функция-критерий зависит от нескольких переменных $y = y(x_1, x_2, \dots, x_n)$. Для двух переменных функция $y = y(x_1, x_2)$ геометрически изображается поверхностью в трехмерном пространстве (рис. 1.13). В общем случае имеет место поверхность в $(n+1)$ -мерном пространстве.

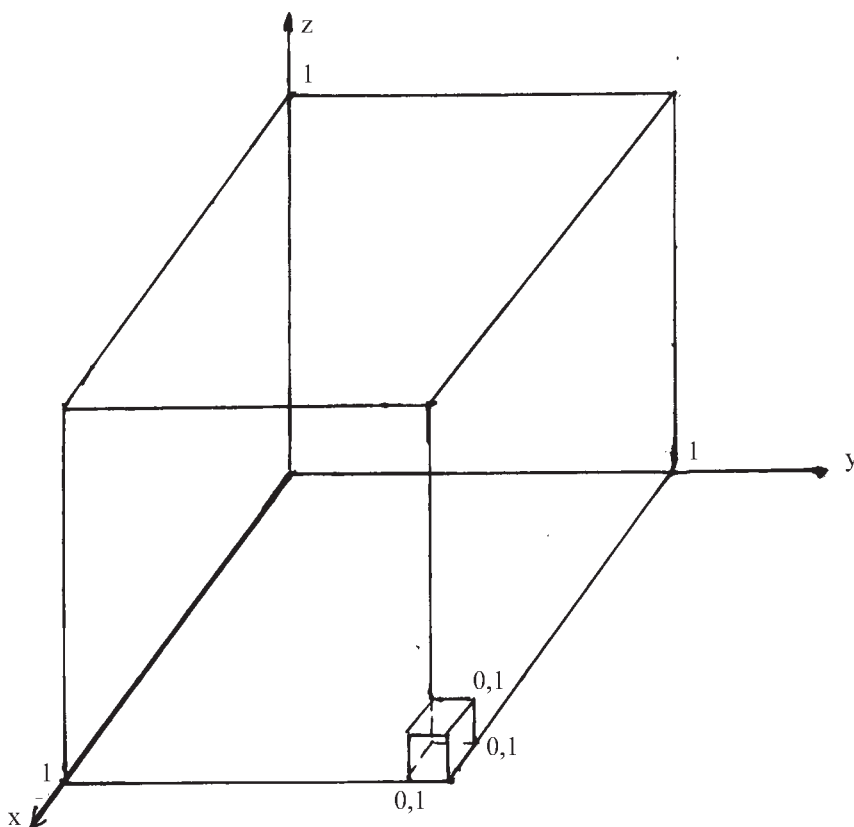


Рис. 1.13. К проблеме размерности при многомерном поиске

Как и в случае одномерного поиска экстремума, имеется большое количество методов оптимизации в многомерном случае. Находят применение такие методы как:

многомерный случайный поиск экстремума, который проводится вероятностным способом;

метод исключения касательными, при котором исключается поверхность критерия, лежащая по одну сторону от вертикальной плоскости, проведенной через касательную к линии уровня, и другие.

Метод градиента предполагает движение по нормали к линиям уровней (рис. 1.14) и представляет собой модификацию метода, часто называемого покоординатным спуском (подъемом).

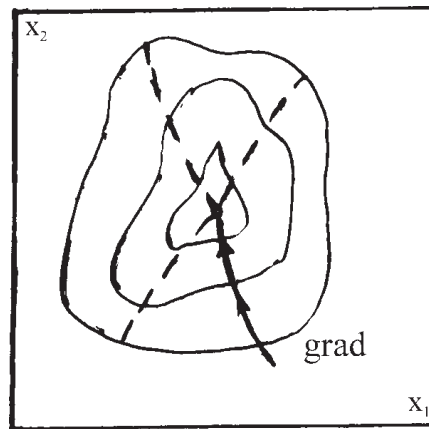


Рис. 1.14. Метод градиента

Идея всех методов спуска состоит в следующем: из начального приближения – точки $x^0 = (x_1^0, x_2^0, \dots, x_n^0)$ необходимо перейти в следующую точку $x^1 = (x_1^1, x_2^1, \dots, x_n^1)$ так, чтобы значение $y(x_1^0, x_2^0, \dots, x_n^0)$ уменьшилось:

$$y(x_1^1, x_2^1, \dots, x_n^1) < y(x_1^0, x_2^0, \dots, x_n^0).$$

С помощью **метода покоординатного спуска** осуществляется поиск из заданной точки в направлении, параллельном одной из осей координат, до точки минимума в данном направлении. Затем поиск производится в направлении, параллельном другой оси, и т. д. по x_1, x_2, \dots, x_n (направления фиксированы, см. рис. 1.15, ломанная a).

Движение осуществляется по одной координате x_i до тех пор, пока не станет равной нулю соответствующая производная, т.е.

$$\partial F / \partial x_i = 0, \quad y = F(x_1, x_2, \dots, x_n)$$

(все остальные координаты сохраняют постоянное значение $x_i = \text{const}$). После этого спуск начинается по другой координате. Обычно начинают с x_1 , потом x_2 и т.д. После того как произведен спуск по всем координатам, начинают повторно с x_1 и т. д. Процесс поиска заканчивается, когда все производные будут равны нулю, т.е.

$$\partial F / \partial x_i = 0, \quad i = 1, 2, \dots, n,$$

точнее, все производные будут меньше порога чувствительности вычислительного метода.

Кажется разумным попытаться модифицировать этот метод таким образом, чтобы на каждом этапе поиск точки минимума производился вдоль “наилучшего” направления. Не ясно, какое направление является “наилучшим”,

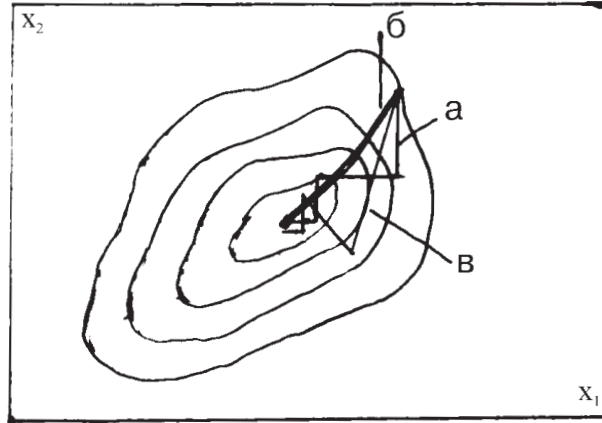


Рис. 1.15. Методы покоординатного (а), градиентного (б) и наискорейшего (в) спусков

но известно, что направление градиента является направлением наискорейшего возрастания функции. Следовательно, противоположное направление является направлением наискорейшего убывания функции. Напомним, что градиент функции $F(x)$ определяется формулой

$$\text{grad } F(x) = [\partial F(x)/\partial x_1, \partial F(x)/\partial x_2, \dots, \partial F(x)/\partial x_n].$$

Вектор $\text{grad } F(x)$ ортогонален линиям уровня (см. рис 1.14) и его направление совпадает с направлением максимального роста $F(x)$ в заданной точке. В точке минимума функции $\text{grad } F(x) = 0$.

Основная проблема заключается в выборе величины каждого дискретного шага. Шаги могут быть постоянными и переменными. Этот метод вполне естествен. Здесь характерен пример с восхождением на гору: метод градиента обеспечивает наибо́льшее достижение вершины. В этом случае линиями уровня будут проекции линий пересечения горизонтальной плоскости с поверхностью горы, т.е. $F(x) = \text{const}$.

Аналитически метод градиента записывается в виде

$$x_i^{k+1} = x_i^k - \lambda (\partial F / \partial x_i)_x^k$$

или

$$x^{k+1} = x^k - \lambda \text{grad } F(x^k),$$

где λ – некоторый постоянный или переменный ($\lambda = \lambda^k$) параметр, определяющий величину шага; $k = 0, 1, 2, \dots$ – номер шага.

Если аналитически производные определить невозможно, их вычисляют приближенно:

$$\partial F / \partial x_i \approx \Delta F / \Delta x_i,$$

где ΔF – приращение функции F при изменении переменной x_i на величину Δx_i . В точке экстремума градиент равен нулю.

Модификаций метода градиента много, но нет единообразия в терминологии. Здесь будут рассмотрены различные модификации, которые не всегда относят к классу градиентных методов и которым в литературе иногда даются иные названия.

Следующая модификация – **метод наискорейшего спуска** (см. рис. 1.15, ломанная линия *в*). Направление градиента определяют в начальной точке $(\text{grad } F)_{x_0}$ и далее в этом направлении осуществляют спуск до тех пор, пока производная $\partial F / \partial x_i$, взятая вдоль этого направления, не обратится в нуль. После этого снова определяют направление градиента и осуществляют по нему спуск до нулевого значения производной и т. д.

Классический метод градиента предполагает движение по кривой градиента, нормальной к линиям уровней (линия *б* на рис. 1.15), и при малом шаге $\Delta x_i = x_i^{k+1} - x_i^k$ может быть описан следующим дифференциальным уравнением:

$$dx/dt = -\lambda \text{ grad } F(x).$$

1.4.2. Методы Ньютона и секущих

Близким к методу градиента является **метод Ньютона** (последовательный поиск методом пересечения касательных с осью абсцисс), который широко используется для отыскания нулей функции (корней уравнения). Поиск нулей функции можно свести к задаче минимизации и наоборот.

Для простоты рассмотрим функцию одной переменной $F(x)$ (рис. 1.16). Итерационный процесс отыскания нуля функции определяется формулой

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

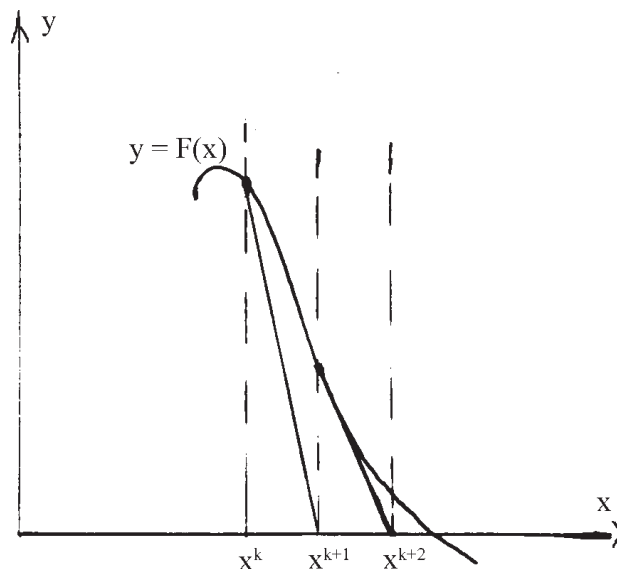


Рис. 1.16. Метод Ньютона

Современный Гуманитарный Университет

Геометрически процесс отыскания нуля функции по методу Ньютона заключается в проведении касательной к кривой $y = F(x)$ в точке x^k , уравнение которой задается соотношением

$$y(x) = F(x^k) - F'(x^k)(x - x^k).$$

Точку пересечения касательной с осью абсцисс принимают за новую точку, в которой проводят следующую касательную и т. д.

Этот метод в отличие от метода градиента не гарантирует сходимости. Для улучшения сходимости используют модифицированный метод Ньютона, который вводит шаг $0 < \lambda^k < 1$. При этом процесс отыскания нуля задается формулой

$$x^{k+1} = x^k - \lambda^k [F(x^k)/F'(x^k)].$$

Более простым для вычислений по сравнению с методом Ньютона является **метод секущих**, хотя он обладает более медленной сходимостью.

Через x^0 и x^1 обозначим две точки, в которых функция одной переменной $y = F(x)$ имеет разные знаки (рис. 1.17). Уравнение секущей, проведенной через две точки $[x^0, F(x^0)]$, $[x^1, F(x^1)]$, будет иметь вид:

$$y = [F(x^1) - F(x^0)]x / (x^1 - x^0) + [F(x^0)x^1 - F(x^1)x^0] / (x^1 - x^0).$$

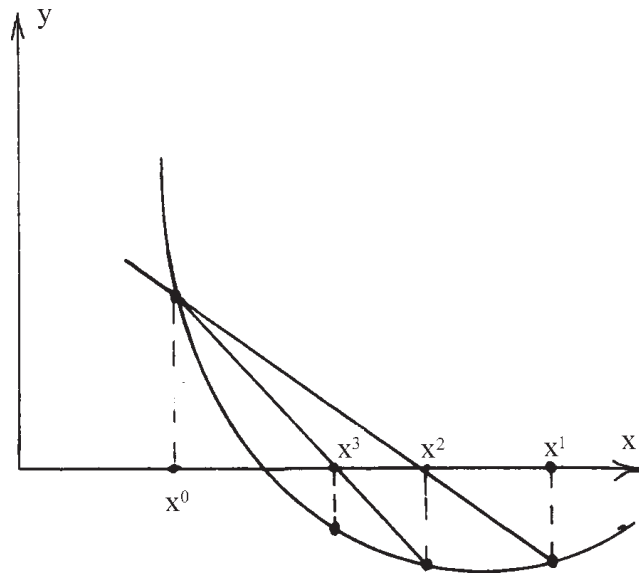


Рис. 1.17. Метод секущих

Точку x^2 пересечения этой прямой с осью абсцисс определим из формулы

$$x^2 = [F(x^1)x^0 - F(x^0)x^1] / [F(x^1) - F(x^0)]$$

и примем за следующее приближение. Очередную секущую проведем через точки $[x^0, F(x^0)]$ и $[x^1, F(x^1)]$ и т. д. В общем случае итерационный процесс задается формулой

$$x^{k+1} = [F(x^k) x^0 - F(x^0) x^k] / [F(x^k) - F(x^0)].$$

В отличие от метода Ньютона здесь не требуется на каждом шаге вычислять производные, достаточно определить только значение самой функции.

1.4.3. Овражный метод

Если топография поверхности имеет “овражный” характер (рис. 1.18), то градиентный метод может оказаться малоэффективным, так как движение может происходить последовательно с одного склона на другой с очень медленным продвижением к точке минимума $x_{\text{опт}}$.

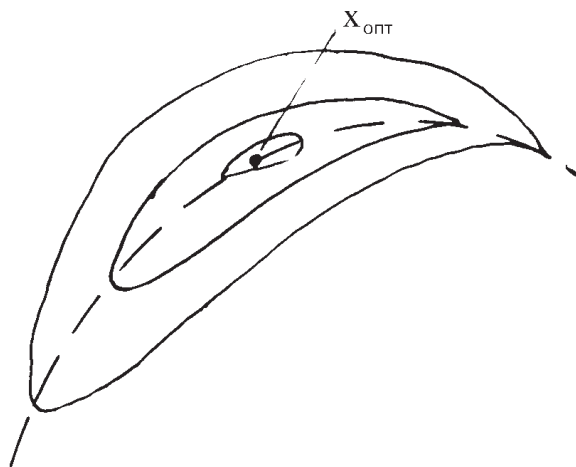


Рис. 1.18. Линии уровней при наличии “оврагов”

В связи с этим разработано много эвристических методов ускоренного продвижения вдоль оврага или гребня, которые так или иначе основаны на релаксационном методе, являющемся обобщением рассмотренных методов. В общем случае **релаксационный итерационный метод** основан на следующей формуле:

$$x^{k+1} = x^k + B^k (dF/dx)_x^k. \quad (1.21)$$

Один метод отличается от другого выбором матрицы B^k . Так, градиентный метод получается, если $B^k = \lambda E$, где E – единичная матрица. Метод покоординатного спуска может быть получен из формулы (1.21), если в матрице B^k все элементы, кроме одного, расположенного на диагонали, равны нулю. На каждой итерации положение ненулевого элемента выбирается заново.

При первом овражном методе задается малое положительное число ε_1 . В очередной точке x^k вычислим все производные $\partial F/\partial x_i$. Далее осуществим градиентный спуск, полагая равными нулю производные, абсолютное значение которых меньше ε_1 , т.е.

$$|\partial F / \partial x_i| \leq \varepsilon_1.$$

Этот метод может обеспечить быстрый спуск на дно “оврага”. Для ускоренного движения по дну “оврага” вводят другое большое положительное число $1 \leq \varepsilon_2$. Опять используют метод градиентов, но полагая равными нулю все те производные, для которых

$$\varepsilon_2 \leq |\partial F / \partial x_i|.$$

Другой метод ускоренного движения по оврагу заключается в следующем. Вначале выбирают две близкие точки \mathbf{x}^* и $\bar{\mathbf{x}}^*$ и из них градиентным методом спускаются с малым шагом в точки \mathbf{x}^1 и $\bar{\mathbf{x}}^1$ соответственно. Далее через них проводят прямую и вдоль нее делают большой овражный шаг λ в точку \mathbf{x}^2 , где процесс повторяется и определяется новое направление овражного шага (рис. 1.19).

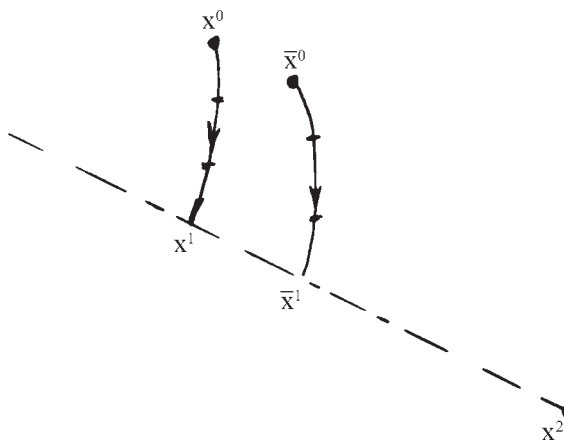


Рис. 1.19. Метод “оврагов”

1.5. Методы отыскания экстремума в условиях помех

Быстрая сходимость при наличии помех (например, когда замеры характеристик оптимизируемого процесса содержат случайные ошибки) может привести к асимптотической ошибке, т.е. поиск будет сходиться к неправильному значению. Основным ограничением в скорости уменьшения интервала поиска является уровень шумов.

Допустим, требуется найти нуль $x_{\text{опт}}$ функции $y(x)$, измерение значений которой $z(x)$ производится с ошибками. Сделаем ограничения относительно помехи. Будем считать, что $y(x)$ – несмещенная оценка функции $z(x)$, т.е.

$$M\{z(x)\} = y(x),$$

где $M\{z(x)\}$ – математическое ожидание.

Очевидно, что $y(x)$ является функцией регрессии для случайного процесса $z(x)$. Напомним, что функция регрессии приближенно представляет статистическую зависимость одной случайной величины от другой.

Далее предположим, что дисперсия ошибки есть величина ограниченная, т.е.

$$\sigma_z^2(x) = M\{z(x) - y(x)\}^2 < \infty. \quad (1.22)$$

При этом x изменяется по закону

$$x^{k+1} = x^k - \gamma^k z(x^k),$$

где x^k и x^{k+1} – значения x в k -м и $(k+1)$ -м экспериментах; $z(x^k)$ – результат измерения на k -м шаге. Смысл **стохастической аппроксимации** очень прост: на каждом шаге коррекцию в значение коэффициента γ^k вносят пропорционально значению функции, измеренному на предыдущем шаге. Допустим, что помеха отсутствует или мала. Тогда в зависимости от вида функции и значения коэффициента γ^k возможны два случая: в результате коррекции точка x^{k+1} или не дойдет до нуля функции $y(x)$, или перейдет за него. В первом случае процесс поиска монотонно сходится к нулевой точке, во втором случае он носит колебательный характер. При наличии помех процесс носит тот же характер. Коэффициенты γ^k определяют тангенс угла, образованного гипотенузой треугольников с вертикальной прямой:

$$-\gamma^k = (x^{k+1} - x^k) / z(x^k).$$

В методе стохастической аппроксимации γ^k задаются заранее и должны удовлетворять определенным условиям. Для сходимости процесса поиска необходимо, чтобы

$$\lim_{k \rightarrow \infty} \gamma^k = 0 \quad (1.23)$$

и последовательность величин γ^k не должна сходиться, т.е.

$$\sum_{k=1}^{\infty} \gamma^k = \infty; \quad \sum_{k=1}^{\infty} (\gamma^k)^2 < \infty. \quad (1.24)$$

В этом смысле в качестве **коэффициентов коррекции** γ^k можно выбрать гармоническую последовательность

$$\gamma^k = 1/k.$$

Условие (1.24) можно объяснить следующим образом. Если длина шага уменьшается в соответствии со сходящимся рядом, то процесс может не дойти до нуля, так как величина общей коррекции ограничена. Очевидно, что ряд

$$\sum_{k=1}^{\infty} 1/k^p$$

при $p > 1$ сходится, т.е.

$$\infty > \sum_{k=1}^{\infty} 1/k^p,$$

а при $p \leq 1$ расходится, т.е.

$$\sum_{k=1}^{\infty} 1/k^p = \infty. \quad (1.25)$$

Однако ряд, составленный из квадратов коэффициентов, должен сходиться:

$$\sum_{k=1}^{\infty} (\gamma^k)^2 < \infty. \quad (1.26)$$

Этому условию опять удовлетворяет гармоническая последовательность

$$\sum_{k=1}^{\infty} 1/k^2 = 1 + 1/4 + 1/9 + \dots < \infty.$$

Смысл этого условия заключается в том, что случайная составляющая σ должна уменьшаться с ростом k .

Чаще всего в качестве последовательности коэффициентов коррекции выбирается гармоническая последовательность.

Наконец, имеется еще одно условие, которому должно удовлетворять множество значений x и $z(x)$. Так как коррекция на каждом шаге пропорциональна значению $z(x)$ на предыдущем шаге, то надо быть уверенным, что величины $z(x^k)$ ограничены. Выше было наложено ограничение на величину дисперсии ошибки $y(x) - z(x)$, поэтому достаточно потребовать, чтобы была ограничена функция регрессии :

$$|y| < A |x - x_{\text{опт}}| + B < \infty. \quad (1.27)$$

Величины A и B должны быть конечны, но не обязательно известны. Если величина A известна хотя бы приближенно, то можно ускорить поиск. При выполнении условий (1.22) – (1.26) эта процедура сходится в среднеквадратичном смысле:

$$\lim_{k \rightarrow \infty} M\{(x^k - x_{\text{опт}})\} = 0.$$

При отыскании максимума функции для одномерного случая в условиях помех следует записать формулу применительно к производной $y'(x)$, обозначив ее измерение через $z'(x)$. Тогда получим

$$x^{k+1} = x^k + \gamma^k z'(x^k),$$

т.е. процедура стохастической аппроксимации в этом случае изменяет шаг пропорционально производной от функции на k -м шаге или пропорционально градиенту в многомерном случае. Таким образом, это просто градиентный метод. Но в результате дифференцирования функция $z'(x^k)$ гораздо сильнее загрязнена шумами [больше отличается от $y'(x^k)$, чем $z(x)$ от $y(x)$]. Можно не дифференцировать функцию, а изменять шаг пропорционально величине

$$[z(x^k + b^k) - z(x^k - b^k)] / 2b^k,$$

где b^k – некоторая постоянная. При малом b^k эта величина пропорциональна производной z' . Поэтому процедура задается формулой

$$x^{k+1} = x^k + \gamma^k [z(x^k + b^k) - z(x^k - b^k)] / 2b^k,$$

где γ^k – положительные числа. Коэффициенты должны удовлетворять условиям:

$$\begin{aligned} \lim_{k \rightarrow \infty} \gamma^k &= 0 & \text{при } k \rightarrow \infty \\ \lim_{k \rightarrow \infty} b^k &= 0 & \text{при } k \rightarrow \infty \end{aligned}$$

$$\sum_{k=1}^{\infty} \gamma^k = \infty;$$

$$\sum_{k=1}^{\infty} (\gamma^k / b^k)^2 < \infty.$$

Для сходимости необходимо выполнение не очень жестких ограничений. Имеются также условия, аналогичные несмещенности оценки и ограниченности дисперсии помехи. При этом доказано, что процедура сходится (т.е. процесс сходится к точке максимума) как в среднеквадратичном, так и в вероятностном смысле при условии одного максимума (униmodalной функции).

Напоминаем, что вышеописанные методы по существу являются градиентными методами отыскания экстремума.

Если производная (или градиент) велика в данной точке (например, крутая гора), необходим большой шаг, так как k -я точка находится далеко от экстремума (вершины). При малой производной шаг становится меньше, чтобы не “проскочить” экстремум (вершину).

2. ОСНОВЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

2.1. Задачи линейного программирования

2.1.1. Математическая формулировка задачи линейного программирования

Большинство задач оптимизации относится к нелинейным, но решение нелинейных задач – это сложная вычислительная проблема. Поэтому практически во всех реальных приложениях для **решения нелинейных задач** используются **приближенные методы** решения. Линейное программирование выделяется среди других методов программирования как основа для многих процедур решения.

Математически задача линейного программирования ставится следующим образом: ищется максимум **линейной формы (функции цели)**

$$L = c_1 x_1 + c_2 x_2 + \dots + c_n x_n = \sum_{i=1}^n c_i x_i \quad (2.1)$$

при условиях

$$a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n \leq b_1, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n;$$

или

$$-a_{i1}x_1 - a_{i2}x_2 - \dots - a_{in}x_n + b_i \geq 0.$$

Словесно **задачу линейного программирования** можно сформулировать так: требуется найти максимум линейной формы от n переменных при m ограничениях в виде неравенств или равенств. Нетрудно убедиться, что всегда можно говорить только о равенствах, так как введением дополнительных переменных x_{n+v} ($v = 1, \dots, p \leq m$) неравенства всегда можно свести к равенствам. Так, ограничение

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \quad (2.2)$$

можно свести к равенству, добавив переменную x_{n+1}

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + x_{n+1} = b_i. \quad (2.3)$$

Тогда условие (2.2) сведется к (2.3) и условию неотрицательности переменной x_{n+1} . Поэтому можно сказать, что при **решении задачи линейного программирования** определяются такие значения n переменных x_j , которые бы обращали в максимум линейную форму

$$L = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

при условии выполнения m равенств

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i, \quad i = 1, 2, \dots, m$$

и n неравенств $x_j, \quad j = 1, 2, \dots, n$.

2.1.2. Примеры прикладных задач линейного программирования

Рассмотрим несколько практических задач, которые сводятся к линейному программированию.

Транспортная задача

В трех месторождениях добывается определенное количество угля. Имеются три пункта потребления угля. Известны расстояния между пунктами добычи и потребления и стоимость перевозок c_{ij} ($i = 1, 2, 3; j = 1, 2, 3$). Необходимо так определить девять чисел x_{ij} , означающих количество грузов, перевозимых из пункта добычи в пункт потребления, чтобы суммарная стоимость перевозок была минимальна:

$$L = \sum c_{ij}x_{ij} \rightarrow \min$$

при условиях

$$x_{1j} + x_{2j} + x_{3j} = b_j, \quad j = 1, 2, 3,$$

требующих, чтобы спрос удовлетворялся во всех пунктах, и при условиях

$$x_{i1} + x_{i2} + x_{i3} = a_i, \quad i = 1, 2, 3,$$

требующих, чтобы из каждого пункта добычи вывозилось угля не больше количества a_i , которое на нем производится. Считается, что количество добытого угля равно сумме потребляемого, т.е.

$$\sum_{i=1}^3 a_i = \sum_{j=1}^3 b_j,$$

хотя это ограничение непринципиально.

Задача о рациональном питании

При правильном питании калорийность пищевых продуктов должна полностью возмещать расход энергии человека, причем потребление отдельных растительных и животных жиров, белков, углеводов и витаминов не должно превышать определенную норму.

Предположим, имеется n различных продуктов калорийностью a_{ij} ($j = 1, 2, \dots, n$), равной числу калорий в единице массы. В единице массы j -го продукта содержится a_{2j} жиров, a_{3j} белков, a_{4j} углеводов. Обозначим через b_1, b_2, b_3, b_4 потребность организма в энергии, жирах, белках и углеводах соответственно. Тогда при правильном питании должны выполняться следующие соотношения:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1;$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2;$$

$$a_{31}x_1 + a_{32}x_2 + \dots + a_{3n}x_n \leq b_3;$$

$$a_{41}x_1 + a_{42}x_2 + \dots + a_{4n}x_n \leq b_4,$$

где $x_j \geq 0$ – количество потребления j -го продукта.

Если ввести требование экономного расходования средств, то это эквивалентно критерию

$$L = \sum_{j=1}^n c_j x_j \rightarrow \min.$$

Если поменять знаки b_1, a_{1j} ($j = 1, 2, \dots, n$), то первое неравенство запишется в виде

$$-a_{11}x_1 - a_{12}x_2 - \dots - a_{1n}x_n \leq -b_1.$$

После этого задача о рациональном питании приобретает стандартный вид задачи линейного программирования.

Задача об использовании ресурсов

Предприятие имеет определенное количество ресурсов: рабочую силу, сырье, оборудование и т. д. Для простоты будем считать, что число ресурсов равно трем, и каждого ресурса имеется b_1, b_2, b_3 условных единиц. Предприятие выпускает два вида товаров. Для производства единицы каждого товара затрачивается a_j ресурсов. Известна стоимость c_i единицы каждого товара. Требуется при данных ресурсах выпустить такую комбинацию товаров x_1 и x_2 , чтобы доход предприятия L был максимален. При линейной зависимости стоимости продукции от количества продукции задача записывается в виде

$$a_{i1}x_1 + a_{i2}x_2 \leq b_i; \quad i = 1, 2, 3; \quad x_j \geq 0$$

выпуклую линейную комбинацию

$$\mathbf{x}^3 = \lambda \mathbf{x}^1 + (1 - \lambda) \mathbf{x}^2, \quad (2.5)$$

где λ – произвольное действительное число, для которого $0 \leq \lambda \leq 1$, то вектор \mathbf{x}^3 также будет принадлежать этой области: $\mathbf{x}^3 \in G$.

На рис. 2.1 изображены выпуклая и невыпуклая области значений \mathbf{x}^1 и \mathbf{x}^2 . Действительно, если на рис. 2.1, б выбрать точку \mathbf{x}^3 , равную линейной комбинации точек \mathbf{x}^1 и \mathbf{x}^2 , то она не будет принадлежать заштрихованной области, и поэтому эта область не будет выпуклой. С другой стороны, нетрудно убедиться, что для любых двух точек, принадлежащих заштрихованной области на рис. 2.1, а, выполняется условие (2.5), и поэтому эта область является выпуклой.

Линейная форма (2.1) в случае двух переменных принимает вид

$$L = c_1 x_1 + c_2 x_2. \quad (2.6)$$

Это уравнение прямой в плоскости (x_1, x_2) , пересекающей оси x_1 и x_2 в точках L/c_1 и L/c_2 соответственно (рис. 2.2). Величины c_1 и c_2 определяют

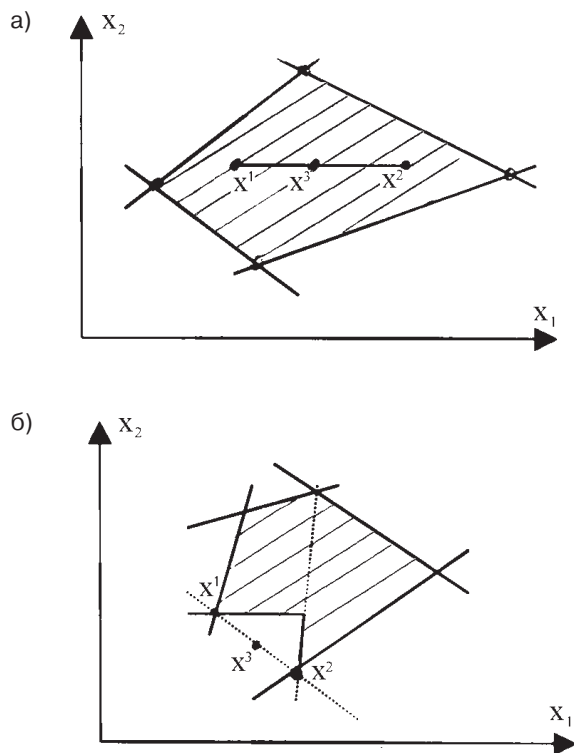


Рис. 2.1. Выпуклая (а) и невыпуклая (б) области значений переменных

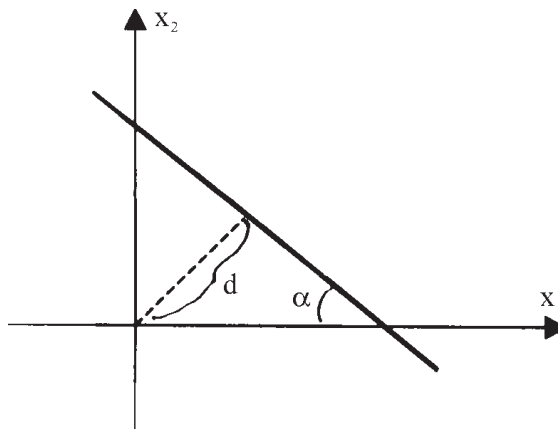


Рис. 2.2. Геометрическая интерпретация линейной функции

наклон прямой [угол наклона к оси x_1 задается формулой $\cos \alpha = c_1 / (c_1^2 + c_2^2)^{1/2}$ определяет расстояние от начала координат до прямой, которое находится из формулы $d = L / (c_1^2 + c_2^2)^{1/2}$]. Теперь можно дать **геометрическую интерпретацию задачи линейного программирования**. Если требуется определить такие x_1 и x_2 , которые придали бы линейной форме минимальное значение, то геометрически это означает, что необходимо провести прямую L [формула (2.6)], проходящую хотя бы через одну точку области и имеющую минимальное расстояние d от начала координат (рис. 2.3, а). В случае максимизации это расстояние должно быть максимальным (рис. 2.3, б). Могут представиться два варианта: прямая имеет с допустимой областью G одну общую точку в вершине области или совпадает с одним из ее ребер. Во втором варианте (рис. 2.4) имеет место вырожденный случай, т.е. линейная функция цели совпадает с левой частью одного из ограничений.

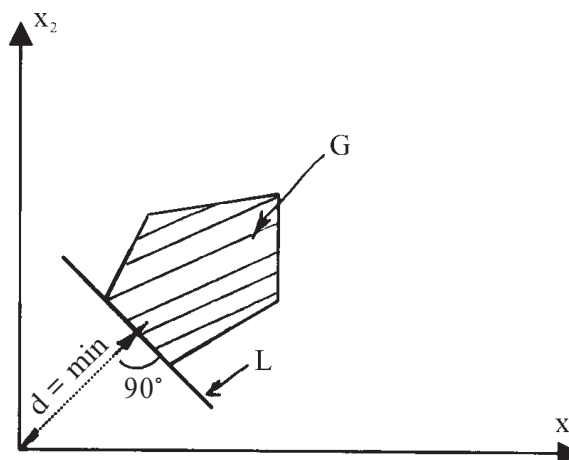


Рис. 2.4. Геометрическая интерпретация линейного программирования, вырожденный случай

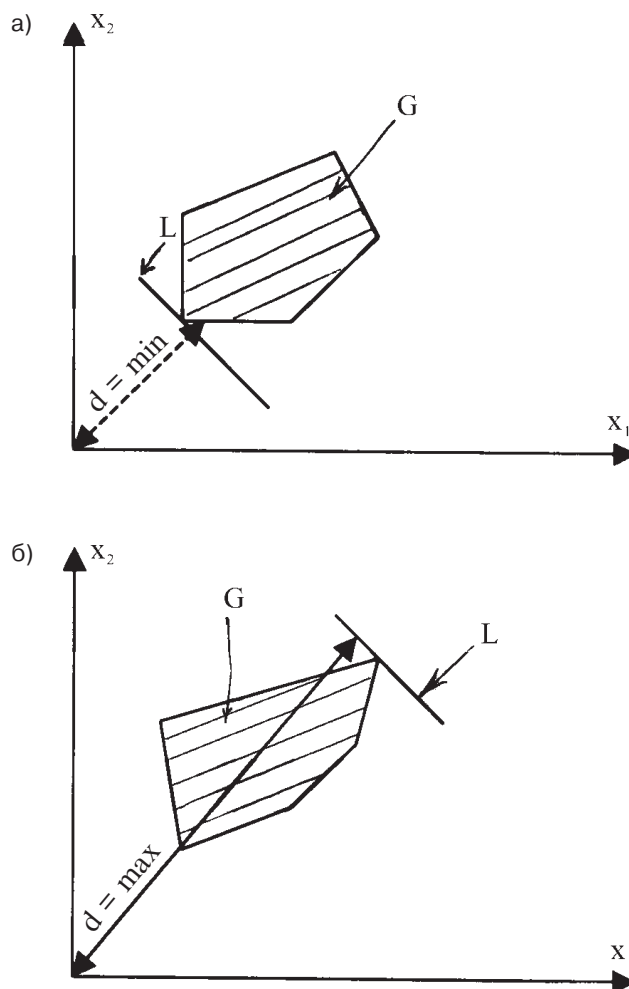


Рис. 2.3. Геометрическая интерпретация линейного программирования для случаев минимума (а) и максимума (б)

2.3. Симплекс-метод в линейном программировании

Задачи линейного программирования удобно решать **симплекс-методом** – специальным методом оптимального (направленного) перехода. Он заключается в последовательном переходе от одной вершины области допустимых значений к другой, соседней, в которой значение функции цели лучше, чем в исходной точке. Движение происходит по периметру контура двумерной области или в случае более двух переменных по ребрам многомерного многогранника. Более оптимальным, строго говоря, был бы путь

не по периметру (или ребрам), а поперек области или вдоль грани, или внутри объема многогранника к оптимальной вершине.

Аналитически доказывается, что:

– **экстремум в задачах линейного программирования** – единственный, т.е. локальный экстремум одновременно является и глобальным и достигается на границе области допустимых значений, как правило, в вершине (при этом исключаются из рассмотрения особые случаи, в том числе случаи достижения экстремума на множестве точек, расположенных на ребре выпуклой области допустимых значений);

– симплекс-метод обеспечивает сходимость к экстремальной точке за конечное число шагов (**конечность симплекс-метода**), так как он предусматривает последовательный просмотр вершин многогранника, число которых конечно, и (при определенных оговорках) является оптимальной процедурой отыскания экстремума.

Пусть исходная формулировка задачи содержит только положительные переменные и равенства, а не неравенства-равенства. Если этого нет, то изменением начала координат можно добиться положительности всех координат в допустимой области, а введением дополнительных положительных переменных свести неравенства к равенствам.

Геометрически (для случая $n = 2$) этот метод означает следующее. На первом шаге выбирают любую вершину многогранника и проводят через нее прямую – функцию цели. Решения (конкретные значения x_1, x_2), соответствующие вершине, будут опорными (базисными). По найденным значениям x_1, x_2 и функции цели находят направление к другой вершине (второй шаг), в которой функция цели возрастает (или убывает, если ищется минимум). В результате получают второе опорное решение. Симплекс-метод дает оптимальную последовательность шагов, приводящую к оптимальному решению, если оно существует.

Наглядная геометрическая интерпретация процесса нахождения оптимального решения получится, если от исходной прямоугольной системы координат (x_1, x_2) перейти к двумерной косоугольной системе введением дополнительных неосновных свободных переменных

$$y_v = x_{n+v}, \quad v = 1, \dots, m$$

в соответствии с равенством (2.3). Тогда для случая двух переменных имеем

$$y_v = a_{v1}x_1 + a_{v2}x_2 + b_v.$$

В качестве преобразующих формул можно выбрать любые из соотношений (1.4). При этом в области допустимых значений y_v положительны. Из всего многообразия набора новых переменных следует выбрать такую систему координат, в которой бы новое начало координат $y = 0$ совпадало с одной из вершин многогранника. В противном случае придется приближать новое начало координат к одной из вершин многогранника.

Пример. Максимизируем функцию

$$L = -4x_1 + x_2 = \max \quad (2.7)$$

при ограничениях:

$$\begin{aligned} x_1 + 4x_2 - 8 &\geq 0; \\ 2x_1 - x_2 + 2 &\geq 0; \end{aligned}$$

$$\begin{aligned} -x_1 - x_2 + 10 &\geq 0; \\ -x_1 + 2x_2 + 2 &\geq 0. \end{aligned} \quad (2.8)$$

Для этого перейдем от прямоугольной системы координат (x_1, x_2) к косоугольной (y_1, y_2) (рис. 2.5), для которой (система 1)

$$\begin{aligned} y_1 &= -x_1 + 2x_2 + 2 = x_3; \\ y_2 &= x_1 + 4x_2 - 8 = x_4. \end{aligned}$$

В новой системе 1 осями координат будут прямые $y_1 = 0$, $y_2 = 0$, совпадающие с ребрами AD и AB, начало координат совпадает с вершиной A, координаты которой $x_1 = 4$, $x_2 = 1$. Функция цели примет вид:

$$L = -4x_1 + x_2 = 17/16 y_1 - 7/6 y_2 - 15,$$

так как

$$\begin{aligned} x_1 &= -2/3y_1 + 1/3y_2 + 4; \\ x_2 &= 1/6y_1 + 1/6y_2 + 1. \end{aligned}$$

В начале координат (точка A) $y_1 = x_3 = 0$; $y_2 = x_4 = 0$; $L_A = -15$. Функция цели возрастает при увеличении $y_1 = x_3$, поэтому следует двигаться в положительном направлении оси $y_1 = x_3$; $y_2 = x_4 = 0$ и в вершине B перейти к новой системе координат (система 2):

$$\begin{aligned} y_2 &= x_4 = x_1 + 4x_2 - 8; \\ y_3 &= x_5 = 2x_1 - x_2 + 2 \end{aligned}$$

(точка пересечения прямых $y_2 = 0$, $y_3 = 0$ определяет вершину B). Выражая из формул (2.8) x_1 и x_2 через y_2 , y_3 , получаем:

$$\begin{aligned} x_1 &= 1/9y_2 + 4/9y_3; \\ x_2 &= 2/9y_2 - 1/9y_3 + 2. \end{aligned}$$

Подставляя эти выражения в линейную форму (2.7), имеем

$$L = -4x_1 + x_2 = -2/9y_2 - 2y_3 + 2.$$

Отсюда значение функции цели в вершине B $L_B = +2$.

Поскольку оба коэффициента при линейной форме отрицательны, увеличивать ее значение, оставаясь в пределах ограничений (2.8) (т.е. внутри многоугольника ABCD), больше невозможно – максимум достигнут.

Заметим, что на рис. 2.5 граница области соответствует прямым $y_i = 0$ ($i = 1, 2, 3, 4$). Нумерация всей косоугольной системы координат меняется на каждом шаге. Так, на первом шаге прямая AD является осью y_2 , а прямая AB – осью y_1 ; на втором шаге прямая AB является осью y_3 , а прямая BC – осью y_2 . Косоугольная система координат будет выбрана неудачно, если:

$$\begin{aligned} y_1 &= -x_1 + 2x_2 + 2 = x_3; \\ y_3 &= 2x_1 - x_2 + 2 = x_5. \end{aligned}$$

В этом случае начало координат ($y_1 = x_3 = 0$, $y_3 = x_5 = 0$) придется на точку E с координатами $x_1 = -1,5$, $x_2 = -2$, не принадлежащую области допустимых

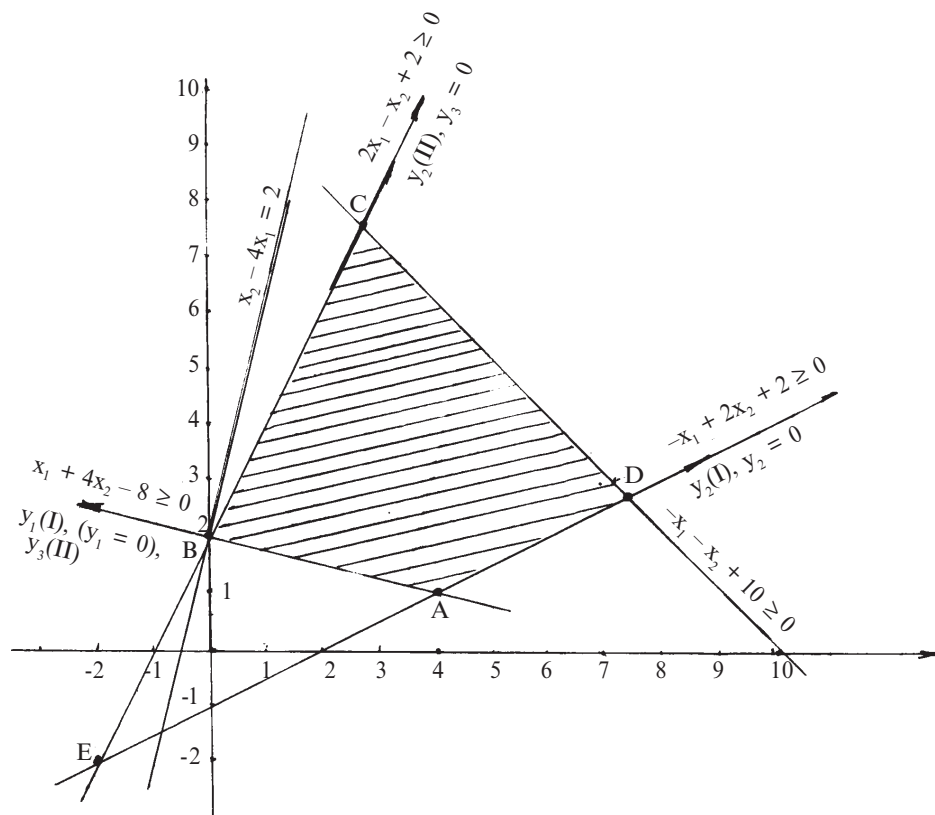


Рис. 2.5. Геометрическая интерпретация симплекс-метода для двумерного случая

значений. Общее правило выбора начальной вершины заключается в том, что она должна лежать на пересечении пары осей косоугольных координат $y_i = 0$ ($i = 1, 2, \dots, m$) и при подстановке ее координат в другие ограничения остальные координаты y_j должны быть положительны, т.е. точка должна принадлежать допустимой области.

При большом числе переменных геометрическая интерпретация не так удобна и целесообразно найти аналитический алгоритм определения оптимального решения.

В симплекс-методе, как это показано на примере, признаком движения вдоль грани является положительность знака коэффициента линейной функции цели, выраженной через косоугольные координаты. Двигаться к очередной вершине следует в положительном направлении той косоугольной координаты y_i , при которой коэффициент положителен, причем в случае многих переменных

$$L = c_1 y_1 + c_2 y_2 + \dots + c_m y_m$$

этот коэффициент должен быть наибольшим. При многих переменных геометрическая интерпретация симплекс-метода с помощью косоугольной системы координат сохраняет свою силу, только число координат должно быть равно числу ребер, исходящих из данной вершины (рис. 2.6). Далее заметим, что косоугольные координаты y_i совпадают с дополнительными переменными x_{n+i} , которые вводятся для того, чтобы ограничения-неравенства перевести в строгие равенства.

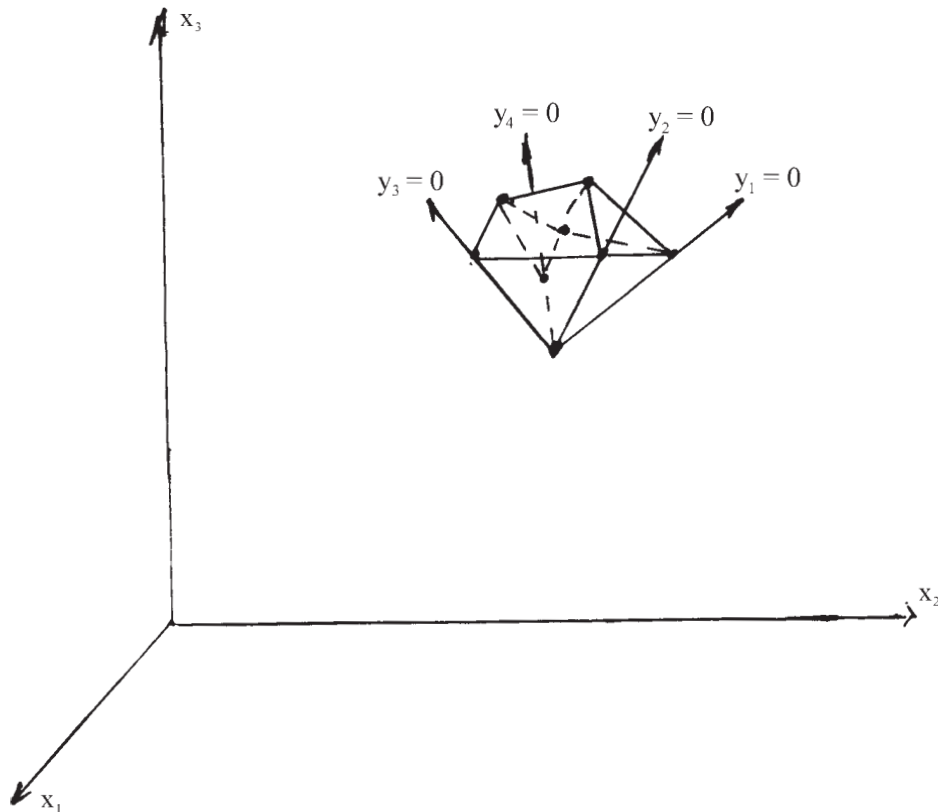


Рис. 2.6. Геометрическая интерпретация симплекс-метода для трехмерного случая

Рассмотрим рис. 2.7, на котором представлена область допустимых значений на плоскости (x_1, x_2) и поверхность функции цели $L(x_1, x_2)$. Так как ограничения линейные, то допустимая область представляет собой выпуклый многоугольник. Из-за линейности функции цели

$$L(x_1, x_2) = c_1 x_1 + c_2 x_2 + b$$

поверхность ее является плоскостью в трехмерном пространстве, наклонной к горизонтальной плоскости (x_1, x_2) под определенным углом. Граница области

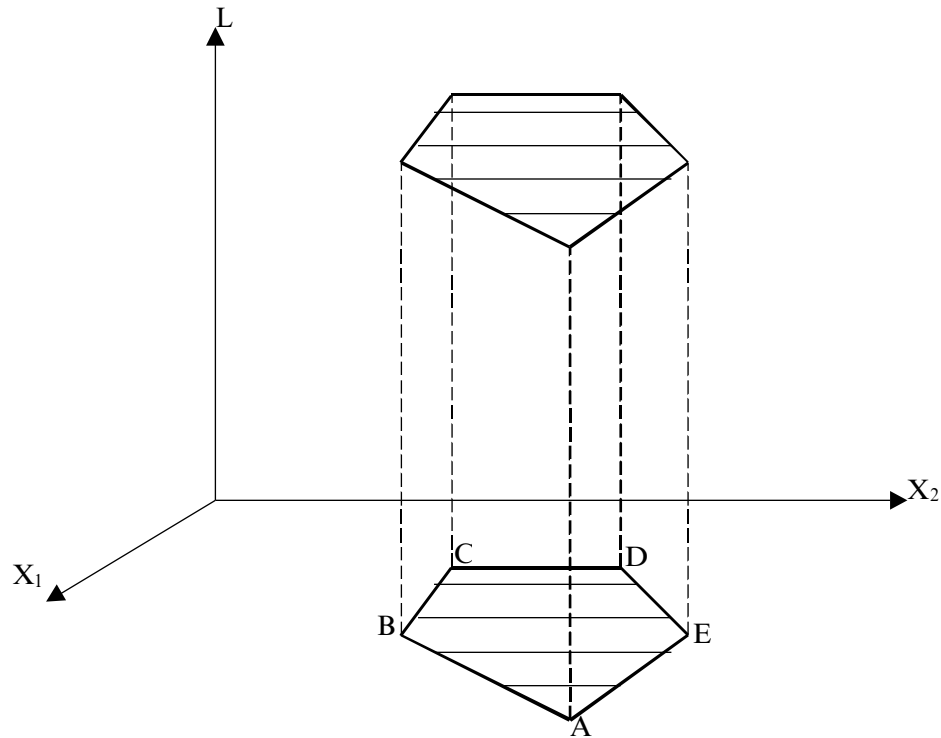


Рис. 2.7. Локальный (он же глобальный) экстремум в линейном программировании

допустимых значений через функцию цели отобразится в некоторый многоугольник в плоскости $L(x_1, x_2)$. Нетрудно с помощью геометрического воображения убедиться, что экстремум совпадает с одной из вершин этого многоугольника и он – единственный. В исключительном случае экстремум достигается на ребре многоугольника, лежащего в плоскости (x_1, x_2) , когда линейная форма совпадает с одним из ограничений и сторона многоугольника в плоскости $L(x_1, x_2)$ параллельна плоскости (x_1, x_2) .

Если функция цели нелинейная, то экстремум может достигаться в середине участка поверхности $L(x_1, x_2)$ и экстремумов может быть несколько. Аналогичная ситуация может появиться при нелинейных ограничениях. Область допустимых значений тогда может быть многосвязной, и в каждой односвязной области достигается свой экстремум.

Таким образом, применительно к симплекс-методу геометрически задачу линейного программирования для случая двух переменных можно представить в следующем виде. Над допустимой областью значений (x_1, x_2) строится поверхность функции цели $L(x_1, x_2)$. Так как функция цели линейная, то ее поверхность будет плоскостью в трехмерном пространстве. Расстояние любой точки поверхности $L(x_1, x_2)$ до горизонтальной плоскости (x_1, x_2) равно значению целевой функции при соответствующих x_1, x_2 . При симплекс-методе ищется

вершина, наиболее или наименее удаленная от плоскости (x_1, x_2) . Если взять три переменных, то необходимо провести четырехмерную плоскость. Область допустимых значений ограничена многогранником в четырехмерном пространстве. Очевидно, что экстремум в этом случае будет достигаться или в вершине, или на ребре, или на грани. Аналогичная ситуация имеет место и в случае любого числа переменных, только тогда будут многомерные многогранники и плоскости.

При нелинейности функции цели в случае двух переменных целевая поверхность может иметь несколько минимумов или максимумов и возникает так называемая **многоэкстремальная задача**, где требуется определить глобальный экстремум.

Обычно считается, что линейная форма зависит от всех переменных

$$L = \sum_{j=1}^n c_j x_j ,$$

но в частных случаях некоторые из c_j равны нулю. На первом шаге выбирают m базисных переменных. Решение системы уравнений

$$\sum_{i,j} a_{ij} x_j + b_i = 0$$

при нулевых значениях небазисных координат называется **базисным решением**. В качестве базисных переменных можно выбрать любые m переменных, для которых векторы $\mathbf{a}_i = ||\mathbf{a}_{ij}||$ линейно независимы, т.е. детерминант матрицы $||\mathbf{a}_{ij}||$ отличен от нуля. Однако на первом шаге их необходимо выбрать так, чтобы базисное решение, соответствующее началу координат косоугольной системы, принадлежало допустимой области (было бы одной из ее вершин). Вопрос о выборе начального базиса (базисного решения) требует специального рассмотрения, которое будет выполнено ниже.

2.4. Формализованная симплекс-таблица

Процедура отыскания экстремума с помощью симплекс-метода может оформляться в виде специальной таблицы (**формализованной симплекс-таблицы**). Разберем эту таблицу на примере. Для этого преобразуем рассмотренную выше задачу к такому виду, чтобы можно было применить симплекс-метод. С этой целью введем дополнительные переменные x_{n+i} :

$$\begin{aligned} x_3 &= -x_1 + 2x_2 + 2 \geq 0; \\ x_4 &= x_1 + 4x_2 - 8 \geq 0; \\ x_5 &= -x_1 - x_2 + 10 \geq 0; \\ x_6 &= 2x_1 - x_2 + 2 \geq 0; \\ L &= -4x_1 + x_2. \end{aligned} \tag{2.9}$$

Пример. Пусть система ограничений с введением переменных задана в виде равенств (2.9), т.е.

$$\begin{aligned} x_1 - 2x_2 + x_3 &= 2; \\ -x_1 - 4x_2 + x_4 &= -8; \\ x_1 + x_2 + x_5 &= 10; \\ -2x_1 + x_2 + x_6 &= 2. \end{aligned}$$

Минимизируем $L = -4x_1 + x_2$ при $x_i \geq 0, i = 1, 2, \dots, 6$. Очевидно, что решение $x_1 = 0, x_2 = 0, x_3 = 2, x_4 = -8, x_5 = 10, x_6 = 2$ удовлетворяет уравнениям (2.9), но не удовлетворяет условию $x_i \geq 0$ и поэтому не является базисным. Выбираемые небазисные переменные образуют косоугольную систему координат в ранее рассмотренной геометрической интерпретации.

Будем считать, что x_3, x_4 – небазисные переменные, и выразим через них базисные переменные и величину L :

$$\begin{aligned} x_1 &= -2/3x_3 + 1/3x_4 + 4; \\ x_2 &= 1/6x_3 + 1/6x_4 + 1; \\ x_5 &= 1/2x_3 - 1/2x_4 + 5; \\ x_6 &= -3/2x_3 + 1/2x_4 + 9; \\ L &= 17/6x_3 - 7/6x_4 - 15. \end{aligned} \quad (2.10)$$

Посмотрим, не достигла ли L своего минимального значения. Коэффициент при x_4 отрицательный, следовательно, возрастание приведет к дальнейшему уменьшению L . Однако при этом необходимо следить, чтобы x_1, x_2, x_5, x_6 , которые зависят от x_4 , не стали отрицательными, т.е. не вышли из допустимой области. Так как увеличение x_4 приводит к увеличению x_1, x_2, x_6 , то для этих переменных такой опасности не существует. Рассматривая переменную x_5 , убеждаемся, что максимально допустимое значение x_4 может быть $x_4 = 10$. При этом:

$$\begin{aligned} x_1 &= 22/3; \quad x_2 = 8/3; \\ x_3 &= 0; \quad x_5 = 0; \quad x_6 = 14. \end{aligned}$$

Поэтому за новый базис могут быть приняты переменные x_1, x_2, x_4, x_6 , т.е. мы перешли к вершине $x_3 = 0, x_5 = 0$. Для того чтобы приступить к следующему шагу, необходимо выразить эти базисные переменные и L через небазисные (через координаты новой косоугольной системы координат). В итоге получим:

$$\begin{aligned} x_1 &= -1/3x_3 - 2/3x_5 + 22/3; \\ x_2 &= 1/3x_3 - 1/3x_5 + 8/3; \\ x_4 &= x_3 - 2x_5 + 10; \\ x_6 &= -x_3 + x_5 + 14; \\ L &= 10/3x_3 + 7/3x_5 - 80/3. \end{aligned}$$

Совершенно очевидно, что как бы мы ни увеличивали x_3, x_5 , уменьшить L не удастся. Следовательно, достигнуто окончательное решение, при котором

$$\begin{aligned} x_1 &= 22/3; \quad x_2 = 8/3; \\ x_4 &= 10; \quad x_6 = 14; \\ L_{\min} &= -80/3. \end{aligned}$$

Вычисление удобно оформлять в виде симплекс-таблицы для каждого шага. Вначале уравнения и линейную форму L на каждом шаге записывают так, что свободные члены располагаются в правой части:

Первый шаг:

x_3
 x_4
 x_5
 x_6

$$\begin{aligned} + x_1 - 2x_2 &= 2; \\ - x_1 - 4x_2 &= -8; \\ + x_1 + x_2 &= 10; \\ - 2x_1 + x_2 &= 2; \\ L + 4x_1 - x_2 &= 0. \end{aligned}$$

Второй шаг:

x_1
 x_2
 x_5
 x_6

$$\begin{aligned} + 2/3x_3 - 1/3x_4 &= 4; \\ - 1/6x_3 - 1/6x_4 &= 1; \\ -1/2x_3 + 1/4x_4 &= 5; \\ +3/2x_3 - 1/2x_4 &= 9; \\ L - 17/6x_3 + 7/6x_4 &= -15. \end{aligned}$$

Третий шаг:

x_1
 x_2
 x_4
 x_6

$$\begin{aligned} + 1/3x_3 + 2/3x_5 &= 22/3; \\ -1/3x_3 + 1/3x_5 &= 8/3; \\ -x_3 + 2x_5 &= 10; \\ x_3 + x_5 &= 14; \\ L - 10/3x_3 - 7/3x_5 &= -80/3. \end{aligned}$$

Записи при вычислениях можно сократить, используя одну из форм симплекс-таблицы для каждого шага (см. табл. 2.1 – 2.3).

В строках, за исключением последней, записываются коэффициенты при соответствующих небазисных переменных, взятые со знаком минус, через которые выражена базисная переменная, соответствующая номеру строки.

Симплекс-таблица 2.1
(первый шаг)

Базисные переменные	Свободные члены в ограничениях	Небазисные переменные	
		x_1	x_2
x_3	2	1	-2
x_4	-8	-1	-4
x_5	10	1	1
x_6	2	-2	1
L	0	4	-1

Симплекс-таблица 2.2
(второй шаг)

Базисные переменные	Свободные члены в ограничениях	Небазисные переменные	
		x_3	x_4
x_1	4	2/3	-1/3
x_2	1	-1/6	-1/6
x_5	5	-1/2	1/4
x_6	9	3/2	-1/2
L	-15	-17/6	7/6

Симплекс-таблица 2.3
(третий шаг)

Базисные переменные	Свободные члены в ограничениях	Небазисные переменные	
		x_3	x_4
x_1	22/3	1/3	2/3
x_2	8/3	-1/3	1/3
x_4	10	-1	2
x_6	14	1	1
L	-80/3	-10/3	-7/3

На каждом шаге в базис включается одна из небазисных переменных, для которой положителен (или отрицателен в случае максимизации линейной формы) элемент, находящийся в самой нижней строке таблицы. Благодаря этому выбирается та переменная, увеличение которой приводит к уменьшению линейной формы (например, на втором шаге переменная x_4). При этом надо помнить, что числа в самой нижней строке симплекс-таблицы равны коэффициентам при соответствующих небазисных переменных в линейной форме, взятых с обратным знаком. Одновременно вычеркиваем из базисных переменных ту, которая дает наименьшее отношение свободного члена к коэффициенту в столбце, при соответствующей выбранной небазисной переменной в ограничениях, причем отрицательные отношения не учитываются, так как соответствующие переменные не могут стать отрицательными. Например, для табл. 2.2 отрицательное отношение свободного члена во второй строке к коэффициенту при переменной x_4 , которую рассматриваем на предмет возможности включения в базис, равно -6 . Это указывает на то, что за счет увеличения переменной x_4 переменная x_2 , которую предполагаем исключить из базиса, не обращается в нуль, что и соответствует уравнению

$$x_2 = 1 + (1/6) x_3 + (1/6) x_4$$

(см. формулу 2.10). Поэтому ее исключать не следует. Геометрически это означает, что с включением переменной x_2 и исключением переменной x_4 вершина многогранника не будет достигнута. Для второго шага остается одна возможность – исключить x_5 , при этом следующая вершина многогранника будет достигнута.

На пересечении строки, вводимой в базис переменной и столбца удаляемой переменной, находится элемент, называемый центральным или опорным, который отмечается звездочкой.

Напомним, что симплекс-таблица строилась для случая минимизации линейной формы.

2.5. Прямая и двойственная задачи линейного программирования

2.5.1. Введение в проблему двойственности

Прямая задача линейного программирования ставится следующим образом. Находят значения переменных (x_1, x_2, \dots, x_n) , удовлетворяющие условиям:

[illegible]

обращающим в максимум линейную форму

При этом допускается, что часть переменных x_1, \dots, x_l имеет любые знаки, а переменные $x_{l+1}, \dots, x_n \geq 0$.

[illegible]

обращающим в минимум линейную форму

Здесь переменные y_1, \dots, y_k также могут иметь произвольные знаки, а $y_{k+1}, \dots, y_m \geq 0$. Заметим, что l равенствам (2.12) соответствуют свободные переменные x_1, \dots, x_p , а $n - l$ неравенствам – неотрицательные переменные прямой задачи x_{p+1}, \dots, x_n . И наоборот, k равенствам прямой задачи (2.11) соответствуют свободные (неограниченные по знаку) переменные y_1, \dots, y_k , а $n - k$ неравенствам – неотрицательные переменные y_{k+1}, \dots, y_m двойственной задачи.

$$\begin{aligned} \mathbf{A} &= \|\mathbf{a}_{ij}\|; \\ \mathbf{A}^T &= \|\mathbf{a}_{ji}\|. \end{aligned} \quad (2.13)$$

Если нет специальных оговорок, далее используется матрица прямой задачи с m строками и n столбцами и i меняется от 1 до m , а j – от 1 до n . Поэтому переменные и коэффициенты имеют соответственно индексы x_j , a_{ij} , y_i , b_i , c_j . Соответственно число ограничений двойственной задачи равно числу неизвестных n прямой, а число неизвестных двойственной задачи равно числу ограничений m прямой задачи.

$$\min L' = \max L,$$

т.е. оптимальные значения функционалов для решений прямой и двойственной задач совпадают.

Все высказанные положения о взаимоотношении прямой и двойственной задач основаны на свойстве замкнутости пары прямой и двойственной задач, которое заключается в том, что задача, двойственная к двойственной, совпадает с основной, иногда они называются взаимно двойственными. По существу, мы специально так подобрали формализм прямой и двойственной задачи, чтобы удовлетворить сформулированному выше свойству замкнутости. Если попробовать изменить, к примеру, что-нибудь в формулировке двойственной задачи (минимум заменить на максимум или число свободных переменных изменить с k на $k + 1$), то это приведет к тому, что задача, двойственная к двойственной, не совпадет с прямой. На определенном уровне строгости, принятом в методе конструирования, никаких других доказательств не требуется. Свойство замкнутости (двойственности) широко используется и по существу является укрупненным свойством, с помощью которого можно экономным способом доказывать различные положения и получать методы оптимизации.

2.5.2. Двойственный симплекс-метод

Решение прямой задачи оптимизации можно определить с помощью симплекс-метода. Его можно получить такой заменой базиса, при которой все свободные члены в соотношениях прямой системы станут неотрицательными, так как только при неотрицательных свободных членах для нулевых значений небазисных переменных базисные переменные будут неотрицательны. Кроме того, экстремальное значение функции цели достигается при такой замене базисных переменных, при которой все коэффициенты при небазисных переменных в выражении для L отрицательны.

Двойственный симплекс-метод называют **методом последовательного улучшения оценок**. Смысл его заключается в том, что вместо прямой задачи решают двойственную и затем по оптимальным значениям переменных двойственной задачи определяют оптимальные значения прямой задачи, причем оптимальное базисное решение одной задачи получается приравниванием ее новых базисных переменных коэффициентам при соответствующих небазисных переменных в линейной форме двойственной задачи, взятым со знаком “-”.

Двойственный симплекс-метод целесообразно применять, когда в исходной задаче число ограничений значительно больше числа неизвестных. Если в этом случае перейти к двойственной задаче, то симплекс-процедура будет проще, чем для прямой задачи. Кроме того, двойственный симплекс-метод широко применяется в различных методах решения задач целочисленного программирования.

Часто решение двойственной задачи называют **псевдопланом**, чтобы отличать его от решения прямой задачи, называемого просто **планом**.

Проще всего проиллюстрировать двойственный метод на примере.

Пример. Рассмотрим следующую задачу линейного программирования:

$$\begin{aligned} -x_1 + 2x_2 - 3x_3 &\leq 1; \\ 2x_1 - x_2 - x_3 &\leq -1; \\ L = -x_1 - 2x_2 - 3x_3 &= \max; \\ x_1 \geq 0, x_2 \geq 0, x_3 &\geq 0. \end{aligned} \quad (2.14)$$

Составим задачу, двойственную (2.14):

$$\begin{aligned} -y_1 + 2y_2 &\geq -1; \\ 2y_1 - y_2 &\geq -2; \\ -3y_1 - y_2 &\geq -3; \\ L' = y_1 - y_2 &= \min; \\ y_1 \geq 0, \quad y_2 &\geq 0. \end{aligned} \quad (2.15)$$

Вводя добавочные переменные y_3, y_4, y_5 , сведем ограничения в виде неравенств к ограничениям в виде равенств. Тогда задача (2.15) примет вид:

$$\begin{aligned} -y_1 + 2y_2 - y_3 &= -1; \\ 2y_1 - y_2 - y_4 &= -2; \\ -3y_1 - y_2 - y_5 &= -3; \\ L' = y_1 - y_2 + 0y_3 + 0y_4 + 0y_5 &= \min; \\ y_1 \geq 0, \quad y_2 \geq 0, \quad y_3 \geq 0, \quad y_4 \geq 0, \quad y_5 \geq 0. \end{aligned} \quad (2.16)$$

Решая задачу (2.16) симплекс-методом, получим симплекс-таблицу 2.4.

Таблица 2.4

Базисные переменные	Свободные члены и ограничения	Небазисные переменные				
		v_1	v_2	v_3	v_4	v_5
y_1	1/5	1	0	0	-1/5	1/5
y_2	12/5	0	1	0	3/5	2/5
y_3	28/5	0	0	1	7/5	3/5
L'	-11/5	0	0	0	-4/5	-1/5

Из последней строки табл. 2.4 получим выражение линейной формы через небазисные переменные

$$L' = (4/5) y_4 + (1/5) y_5.$$

Отсюда, если учесть соответствие между переменными

$$\begin{aligned} y_1 &\leftrightarrow x_4; \quad y_2 \leftrightarrow x_5; \\ y_3 &\leftrightarrow x_1; \quad y_4 \leftrightarrow x_2; \\ y_5 &\leftrightarrow x_3; \end{aligned}$$

получим решение исходной задачи в виде

$$\begin{aligned} x_1 &= 0; \quad x_2 = 4/5; \quad x_3 = 1/5; \\ \max L &= -11/5. \end{aligned}$$

В этом примере использован по сравнению с предыдущим примером другой вариант симплекс-таблицы, в котором помимо столбцов, соответствующих небазисным переменным, добавлены столбцы, соответствующие базисным переменным с нулевыми элементами, за исключением тех, которые

стоят на пересечении строк и столбцов с одинаковыми номерами базисных элементов, равных единице.

Симплекс-таблица такого типа называется полной или расширенной в отличие от сокращенной симплекс-таблицы предыдущего примера.

2.6. Общая теория симплекс-метода на базе линейной алгебры

Рассмотрим задачу линейного программирования

$$\begin{aligned} a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n &= b_i, & i &= 1, \dots, m; \\ L = c_0 + c_1x_1 + c_2x_2 + \dots + c_nx_n &\rightarrow \min, & 0 &\leq x_i. \end{aligned} \quad (2.17)$$

Будем считать, что система уравнений (2.17) содержит r линейно независимых уравнений, где $r \leq m$. Эти r переменных на первом шаге являются базисными переменными, т.е. определяют базис. Строго говоря, с практической точки зрения можно считать, что $r = m$, так как в противном случае равенства (2.17) или несовместны, или хотя бы одно из них представляет линейную комбинацию остальных и поэтому является лишним. Разрешив уравнения (2.17) относительно этих переменных, получим:

$$x_i = b_i - \sum_{j=1}^{n-r} a_{ij} x_{m+j}, \quad i = 1, 2, \dots, r.$$

Без потери общности можно считать, что независимы первые r уравнений (2.17). Предположим, что свободные коэффициенты b_i неотрицательны. Каждое из этих уравнений можно рассматривать как проекцию векторного уравнения

$$\sum_{i=1}^r x_i \mathbf{p}_i = \mathbf{p}_0 - \sum_{j=r+1}^n x_j \mathbf{p}_j \quad (2.18)$$

на единичные векторы, направленные по координатным осям (ортам) $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_r$, где

$$\begin{aligned} \mathbf{p}_1 &= (1, 0, \dots, 0); \\ \mathbf{p}_2 &= (0, 1, 0, \dots, 0); \\ \mathbf{p}_r &= (0, 0, \dots, 1). \end{aligned}$$

Векторы $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_r$ образуют базис в m -мерном пространстве (рис. 2.8). При этом матрица разложений векторов $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$ в базисе $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_r$ представляется в виде

$$\left\| \begin{array}{cccccc} \mathbf{p}_1 & \mathbf{p}_2 & \dots & \mathbf{p}_r & \mathbf{p}_0 & \mathbf{p}_{r+1} & \dots & \mathbf{p}_n \\ 1 & 0 & \dots & 0 & b_1 & a'_{1,r+1} & \dots & a'_{1n} \\ 0 & 1 & \dots & 0 & b_2 & a'_{2,r+1} & \dots & a'_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & b_r & a'_{m,r+1} & \dots & a'_{mn} \end{array} \right\| \quad (2.19)$$

В этой матрице первые r столбцов представляют собой орты системы координат. Столбец $r+1$ состоит из свободных членов ограничений. Последние $n-r$ столбцов состоят из компонент проекций вектор-столбцов $\mathbf{p}_{r+1}, \dots, \mathbf{p}_n$ на орты $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_r$. В целом матрица состоит из $n+1$ вектор-столбцов с m компонентами каждый.

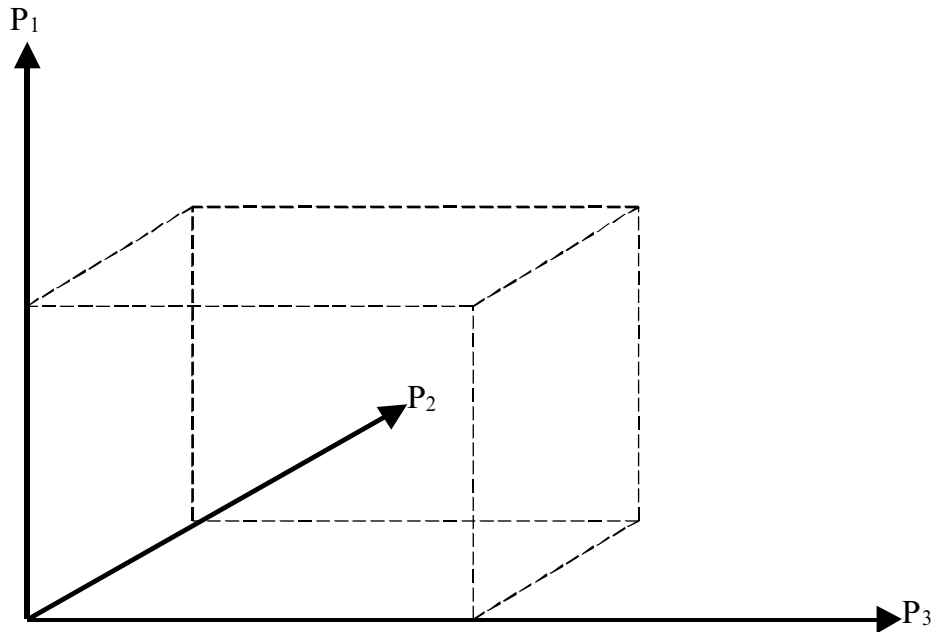


Рис. 2.8. Геометрическая интерпретация базисных переменных для трехмерного случая

Симплекс-таблица теперь примет вид:

Таблица 2.5

	1	x_{r+1}	...	x_n
x_1	b_1	$a_{1, r+1}$...	a_{1n}
x_2	b_2	$a_{2, r+1}$...	a_{2n}
...		...		
x_r	b_m	$a_{r, r+1}$...	a_{rn}
	c_0	c_1	...	c_n

Причем соотношение для L запишется как

$$L = c_0 - \sum_{j=r+1}^n c'_j x_j, \quad i = 1, 2, \dots, r.$$

После процедуры включения в базис небазисных переменных и исключения базисных, отвечающих ряду требований, получим группу векторов $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{r+1}, \dots, \mathbf{p}_r$ образующих новый базис. Их определитель, записанный в старой системе координат, отличен от нуля.

Следовательно, эти векторы независимы и могут быть выбраны за новый базис. Повторяя этот процесс до тех пор, пока все числа в последней строке симплекс-таблицы (2.5) станут неположительны, получим оптимальное решение. Еще раз напомним, что в практически важных случаях $r = m$.

Обсудим методы отыскания начальной вершины многоугольника.

Допустим, задача линейного программирования записана в виде

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b}; \\ \mathbf{Cx} &\rightarrow \min; \\ \mathbf{x} &\geq 0, \end{aligned} \quad (2.20)$$

где $\mathbf{b} \geq 0$; $\mathbf{C} = \|c_1, c_2, \dots, c_n\|$. Стартовый базис считается найденным, если ограничения, содержащиеся в задаче (2.20), записаны как

$$x_v = b_v - (a_{v,m+1} x_{m+1} + \dots + a_{v,n} x_n), \quad v = 1, 2, \dots, m,$$

где $b_v \geq 0$, так как, положив небазисные переменные $x_{m+k} = 0$ ($k = 1, 2, \dots, n-m$), получим значения $x_v = b_v \geq 0$, которые являются допустимыми и соответствуют одной из вершин.

Иногда сразу удастся выделить первый базис. Однако в общем случае это непростая задача.

Для каждого ограничения вводим вспомогательную переменную z_i ($i = 1, \dots, m$). Тогда ограничения системы (2.20) будут выглядеть как

$$\mathbf{Ax} + \mathbf{z} = \mathbf{b}; \quad \mathbf{x} \geq 0, \mathbf{z} \geq 0. \quad (2.21)$$

Без нарушения общности считаем, что \mathbf{b} неотрицательна, так как этого всегда можно достигнуть, поменяв соответствующие знаки в строках матрицы \mathbf{A} . Поэтому допустимым базисным решением будет $\mathbf{x} = 0, \mathbf{z} = \mathbf{b}$. Соответствующая симплекс-таблица на первом шаге запишется в виде

$$\mathbf{z} = \mathbf{b} - \mathbf{Ax},$$

т.е. за базис взят вектор \mathbf{z} .

На первом шаге минимизируется функция цели $\sum z_j$ при ограничениях (2.21). Непосредственно из этих ограничений следует, что если минимум функции $\sum z_j$ является положительной величиной, то система (2.21) не имеет неотрицательных решений, для которых $z_1, \dots, z_m = 0$, так как в противном случае $\min \sum z_j = 0$. Это означает, что исходная система (2.21) не имеет неотрицательных решений. Если $\min \sum z_j = 0$, т.е. $z_j = 0$ для всех j , то вектор \mathbf{z} является базисным. Отправляясь от этого базиса, мы стремимся прийти к базису, не содержащему ни одной вспомогательной переменной

$$x_v = b_v - (a_{v,m+1} x_{m+1} + \dots + a_{v,n} x_n + a_{v1} z_1 + \dots + a_{vm} z_m), \quad v = 1, 2, \dots, m,$$

где $b_v \geq 0$. Полагая в системе (2.21) $z_j = 0$, придем к системе

$$x_v = b_v - (a_{v,m+1} x_{m+1} + \dots + a_{v,n} x_n), \quad v = 1, 2, \dots, m,$$

которая равносильна исходной (2.20). Но здесь x_1, x_2, \dots, x_m – базисные переменные. Следовательно, задача нахождения базиса решена. Может получиться, что при переходе от базиса z_1, z_2, \dots, z_m к следующему эти переменные будут оставаться в следующем базисе. Тогда необходимо

постепенно их перевести в небазисные.

Если среди неизвестных x_1, x_2, \dots, x_m имеется такая переменная, например x_1 , которая входит только в одно уравнение, например в первое, причем коэффициент при этой переменной a_{11} имеет тот же знак, что и свободный член b_1 , тогда не имеет смысла вводить для первого уравнения искусственную переменную, так как x_1 сразу войдет в базис. Действительно, первое уравнение можно представить в виде

$$x_1 = b'_1 - (a'_{12} x_2 + \dots + a'_{1n} x_n),$$

где $b'_1 = b_1/a_{11} \geq 0$. Если таких неизвестных несколько, то все их можно сразу включить в базис.

Пример. Дана система

$$\begin{aligned} x_1 - 2x_2 + x_3 &= 2; \\ -x_1 - 4x_2 + x_4 &= -8; \\ x_1 + x_2 + x_5 &= 10; \\ -2x_1 + x_2 + x_6 &= 2, \end{aligned}$$

рассмотренная в примере ранее. Так как x_3, x_5, x_6 входят каждая только в одно ограничение и коэффициенты при них имеют тот же знак, что и соответствующие свободные члены, включаем их сразу в базис. Для оставшегося ограничения вводим искусственную переменную z_1 . В итоге получаем:

$$\begin{aligned} x_3 &= 2 - (x_1 - 2x_2); \\ x_5 &= 10 - (x_1 + x_2); \\ x_6 &= 2 - (-2x_1 + x_2); \\ z_1 &= 8 - (x_1 + 4x_2 - x_4); \\ x_1 &\geq 0; \quad z_1 &\geq 0. \end{aligned} \tag{2.22}$$

С помощью симплекс-метода требуется минимизировать форму

$$L_1 = z_1 = 8 - (x_1 + 4x_2 - x_4)$$

при ограничениях (2.22). Составляем симплекс-таблицу на первом шаге.

Таблица 2.6

		x_1	x_2	x_4
x_3	2	1	4	-1
x_5	10	1	-2	
x_6	2	1	1	
z_1	8	-2	1	
L_1	8	1	4	-1
L	0	1	4	-1

Необходимо минимизировать L . Опорный элемент выбираем на пересечении строки x_3 и столбца x_1 . После первого шага получаем базис (x_1, x_5, x_6, z_1) . На втором шаге опорный элемент выбирается на пересечении строки

z_1 и столбца x_4 , поэтому искусственная переменная z_1 выводится из базиса. При этом $L_1 = 0$. Таким образом, базис найден и состоит из x_1, x_2, x_5, x_6 :

$$\begin{aligned}x_1 &= 4 - 2/3 x_3 + x_4; \\x_2 &= 1 + 1/6 x_3 + 1/6 x_4; \\x_5 &= 5 + 1/2 x_3 - 1/2 x_4; \\x_6 &= 9 - 3/2 x_3 + 1/2 x_4; \\L &= -15 + 17/6 x_3 - 7/6 x_4.\end{aligned}$$

3. НЕЛИНЕЙНОЕ И ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ. КЛАССИЧЕСКИЕ МЕТОДЫ ОПРЕДЕЛЕНИЯ ЭКСТРЕМУМА ФУНКЦИЙ

3.1. Нелинейное программирование

Нелинейное программирование включает в себя методы определения минимума функции n переменных $F(\mathbf{x})$, где $\mathbf{x} = ||x_1, x_2, \dots, x_n||$ при $m+n$ ограничивающих условиях:

$$\begin{aligned}\varphi_i(\mathbf{x}) &\leq 0, & i &= 1, \dots, m; \\x_j &\geq 0, & j &= 1, \dots, n, \\F(\mathbf{x}) &\rightarrow \min;\end{aligned}\tag{3.1}$$

$$\varphi \leq 0, \quad \mathbf{x} \geq 0.\tag{3.2}$$

Соотношения (3.2) следует понимать таким образом, что каждая компонента векторов φ и \mathbf{x} в них не менее нуля. Иногда сокращенно соотношения (3.1) и (3.2) записывают в виде

$$\min \{ F(\mathbf{x}) \mid x_j \geq 0, j = 1, \dots, n; \quad \varphi_i(\mathbf{x}) \leq 0, i = 1, \dots, m \}$$

или

$$\min \{ F(\mathbf{x}) \mid \mathbf{x} \in G \},$$

где область G задается условиями

$$G = \{ \mathbf{x} \mid \mathbf{x} \geq 0, \varphi_i(\mathbf{x}) \leq 0 \text{ для всех } i \}.$$

В нелинейном программировании допускаются в общем случае любые соотношения между n и m , т.е. $n > m$, $n = m$, $m < n$.

В задачах нелинейного программирования, так же как и в задачах линейного программирования, могут встречаться другие формы написания условий. Но все возможные формы могут быть сведены к виду (3.1), который в дальнейшем будем называть нормальным.

В общем случае функции $F(\mathbf{x})$ и $\varphi_i(\mathbf{x})$ бывают произвольными и, в частности, линейными. Задачи, решаемые прямыми методами поиска, являются частными случаями задач вида (3.1), (3.2).

Задачи поиска экстремума функции при наличии ограничений можно решать с помощью классических методов, но они рассматривают только случаи, когда неравенства имеют вид строгих равенств:

$$\begin{aligned}F(\mathbf{x}) &\rightarrow \min; \\ \varphi_i(\mathbf{x}) &= 0.\end{aligned}$$

При этом не требуется неотрицательности переменных x_j , $m < n$, а функции $F(\mathbf{x})$ и $\varphi_i(\mathbf{x})$ непрерывны и имеют частные производные. Для нелинейного программирования классические методы имеют большое теоретическое значение, так как основополагающая теорема Куна–Таккера в выпуклом программировании обобщает теорему Лагранжа для классических задач. В начале будут рассмотрены классические методы отыскания экстремума функции с ограничениями.

Основной недостаток методов нелинейного программирования заключается в том, что с их помощью не всегда удастся найти глобальный экстремум при наличии нескольких локальных экстремумов. Определить глобальный экстремум можно лишь методом динамического программирования. Однако его применение зависит от определенных условий, обеспечивающих выполнение принципа оптимальности Беллмана. Решение задач нелинейного программирования методом динамического программирования имеет свою специфику, благодаря которой динамическое программирование часто рассматривают в разделе нелинейного программирования. Применение дискретного принципа максимума Понтрягина для решения задач нелинейного программирования менее разработано, чем применение динамического программирования, так как этот метод более “чувствителен” к введению ограничений на переменные.

Теоретически нелинейное программирование разработано только для одного частного случая выпуклых функций $F(\mathbf{x})$ и $\varphi_i(\mathbf{x})$, и соответственно этот раздел назван **выпуклым программированием** (рис. 3.1).

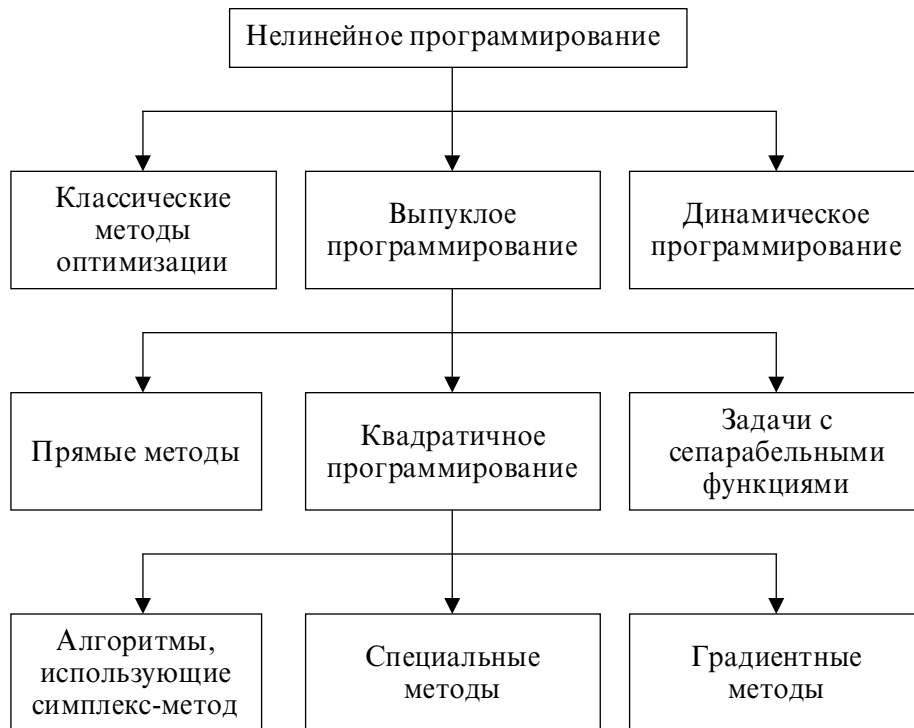


Рис. 3.1. Классификация методов нелинейного программирования

Функция $f(x)$ n переменных $||x_1, \dots, x_n|| = x \in G$ называется **выпуклой функцией в выпуклой области** G , если для любых двух точек из G выполняется соотношение

$$f\{\lambda x^1 + (1 - \lambda)x^2\} \leq \lambda f(x^1) + (1 - \lambda) f(x^2), \quad (3.3)$$

где $0 \leq \lambda \leq 1$. Функция будет **строго выпуклой**, если здесь знак “ \leq ” можно заменить на “ $<$ ”. Соотношение (3.3) означает, что выпуклая функция не может принимать больших значений, чем линейная функция, интерполирующая значения $f(x^1)$ и $f(x^2)$. На рис. 3.2 приведен пример выпуклой функции одной переменной.

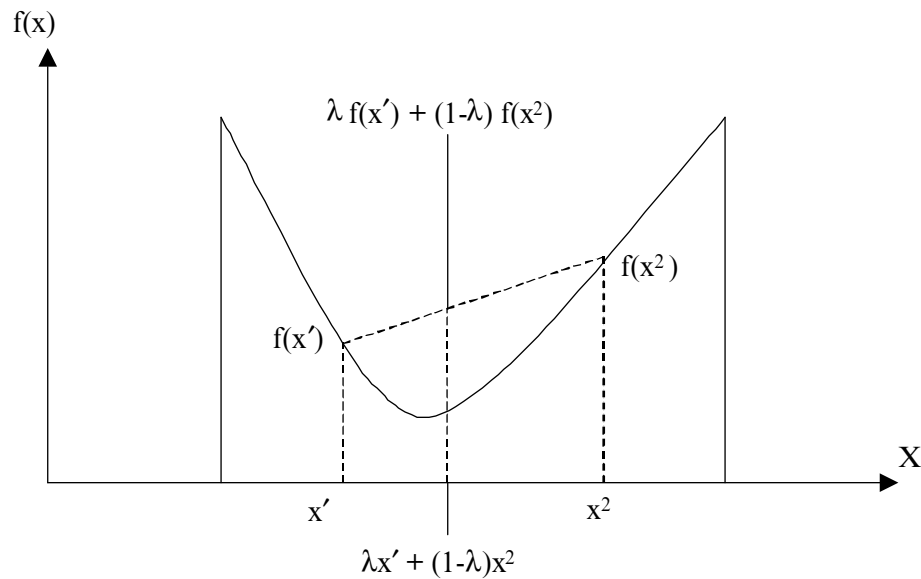


Рис. 3.2. К определению выпуклой функции

Соответственно функция $f(x)$ называется **вогнутой** (или строго вогнутой), если функция $[-f(x)]$ выпукла (строго выпукла).

Следует заметить, что определение вогнутости и выпуклости может показаться на первый взгляд неправильным. Например, тарелка, стоящая на столе, считается вогнутой, но если рассматривать ее с точки зрения нашего определения, т.е. при направлении третьей оси вверх, она выпукла. Дело в том, что обычное понятие выпуклости всегда совпадает с математическим, если смотреть по положительному, а не по отрицательному направлению оси, относительно которой определяется выпуклость. В примере с тарелкой на столе на нее следует смотреть снизу стола, тогда она будет выпуклой. Заметим, что для выпуклых функций (см. рис. 3.2)

$$d^2f(x)/dx^2 \geq 0.$$

Метод считается теоретически разработанным, если найдены соотношения, являющиеся необходимыми и достаточными условиями оптимума, и

алгоритмы поиска экстремума с доказательством их сходимости. Этим требованиям, строго говоря, удовлетворяют только методы, рассматриваемые в разделе квадратичного программирования, частично методы решения задач с сепарабельными функциями и в значительно меньшей степени прямые методы. Функция $f(\mathbf{x}) = f(x_1, \dots, x_n)$ называется **сепарабельной**, если она представлена в виде

$$f(x_1, \dots, x_n) = \sum_{j=1}^n c_j f_j(x_j) = c_1 f_1(x_1) + c_2 f_2(x_2) + \dots + c_n f_n(x_n).$$

В общем случае эти функции не являются выпуклыми. Однако если каждая из функций $f_j(x_j)$ выпуклая и коэффициенты c_j неотрицательны, то функция $f(\mathbf{x})$ тоже выпуклая. Методы решения задач с сепарабельными функциями, основанные на замене нелинейных функций ломаными кривыми, составленными из отрезков прямых, ищут локальный экстремум и не гарантируют отыскание глобального экстремума.

Теоретически наиболее широко и детально в нелинейном программировании разработан раздел выпуклого программирования, называемый **квадратичным**, в котором функции представляются в виде суммы линейной и квадратичной форм и имеют вид:

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{p}\mathbf{x} + \mathbf{x}\mathbf{C}\mathbf{x} = \sum_{j=1}^n p_j x_j + \sum_{j=1}^n \sum_{k=1}^n c_{jk} x_j x_k = \\ &= p_1 x_1 + \dots + p_n x_n + c_{11} x_1^2 + c_{12} x_1 x_2 + \dots + c_{1n} x_1 x_n + c_{nn} x_n^2. \end{aligned} \quad (3.4)$$

Для выпуклости необходимо, чтобы матрица $\mathbf{C} = || c_{jk} ||$ представляла собой симметричную матрицу, т.е. чтобы для любых \mathbf{x} выполнялись условия симметрии $c_{jk} = c_{kj}$, и неотрицательно определенную $\mathbf{x}\mathbf{C}\mathbf{x} \geq 0$.

Методы квадратичного программирования можно разделить на три группы:

- алгоритмы, использующие симплекс-метод;
- градиентные методы;
- специальные методы.

Отличие градиентных методов, рассматриваемых в квадратичном программировании, от рассмотренных в разделе прямых методов заключается в том, что в первом случае благодаря заданию функций в виде (3.4) удается получить значительно больше результатов, характеризующих конкретный метод. В прямых методах поиска, в градиентных и других алгоритмах, как правило, функция цели считается выпуклой (в большинстве случаев она задана аналитически, в виде таблиц или другим способом). Прямые методы по традиции много внимания уделяют аспектам поиска в условиях неопределенности, выбору, в каждом случае, оптимальной стратегии поиска.

Задачи нелинейного программирования по сравнению с задачами линейного программирования обладают большим многообразием. На рис. 3.3 представлены возможные варианты расположения точки экстремума для случая двух переменных. Так, в случае линейных ограничений и нелинейной функции цели экстремума можно достигнуть в крайней точке (вершине) допустимой области значений (рис. 3.3, а), в одной из точек, лежащих на ограничивающих прямых (рис. 3.3, б), и, наконец, в точке, расположенной внутри области (рис. 3.3, в). Пунктирные концентрические окружности изображают линии постоянных значений функции цели, сплошные линии – границу области допустимых значений. На рис. 3.3, б экстремум определяется как точка касания прямой, ограничивающей допустимую область значений, и линии равных значений функции цели.

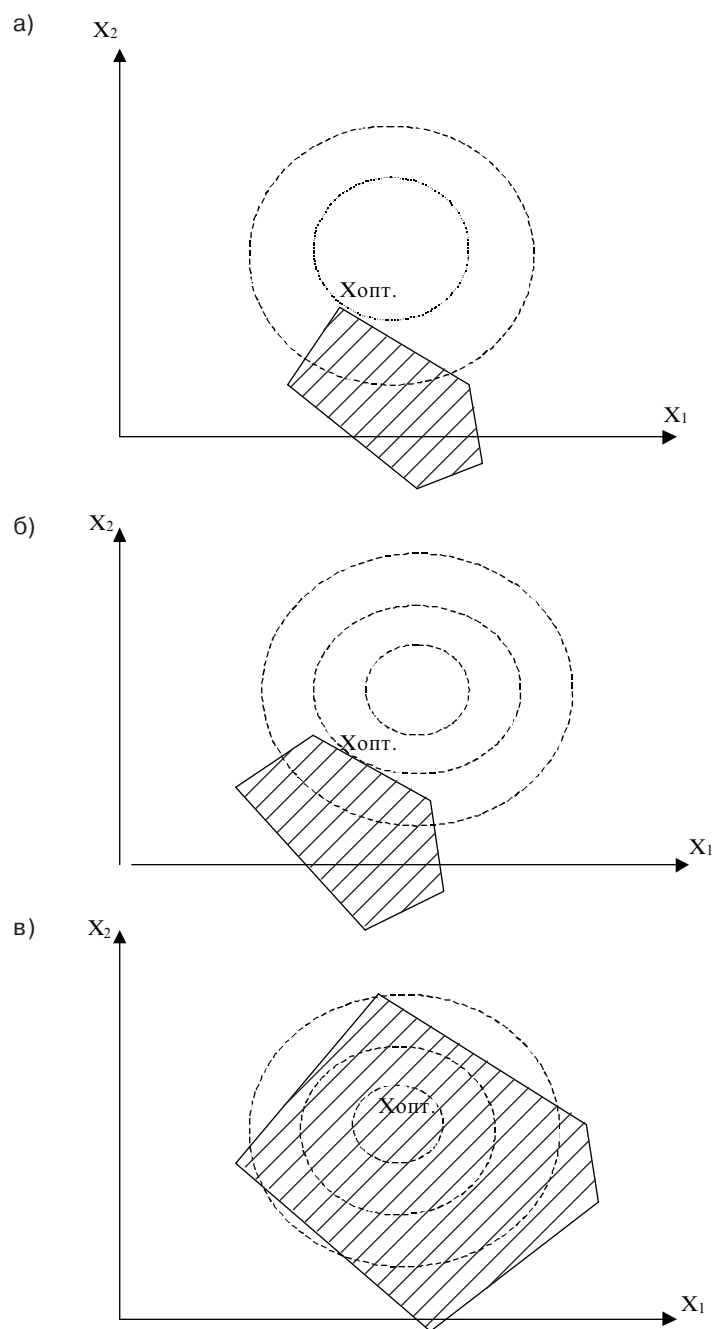


Рис. 3.3. Различные случаи оптимума в задачах нелинейного программирования

Современный Гуманитарный Университет

Решение задач нелинейного программирования может давать два или более экстремума, тогда как решение задач линейного программирования дает один экстремум. На рис. 3.4 показан случай, соответствующий линейным ограничениям и нелинейной (квадратичной) функции цели, где она достигает максимального значения в двух точках: А (локальный максимум) и В (глобальный максимум). На этом рисунке пунктиром обозначены постоянные значения функции цели $F = \text{const} = C_i$, сплошной линией ограничена область допустимых значений. При нелинейных ограничениях может иметь место случай многосвязной области допустимых значений, и в каждой изолированной подобласти функция цели может достигать своего одного или нескольких локальных экстремумов. На рис. 3.5 представлен случай двусвязной области, в которой функция цели достигает локальных экстремумов. Максимум в точке В – глобальный для всей области допустимых значений, в точке А – локальный.

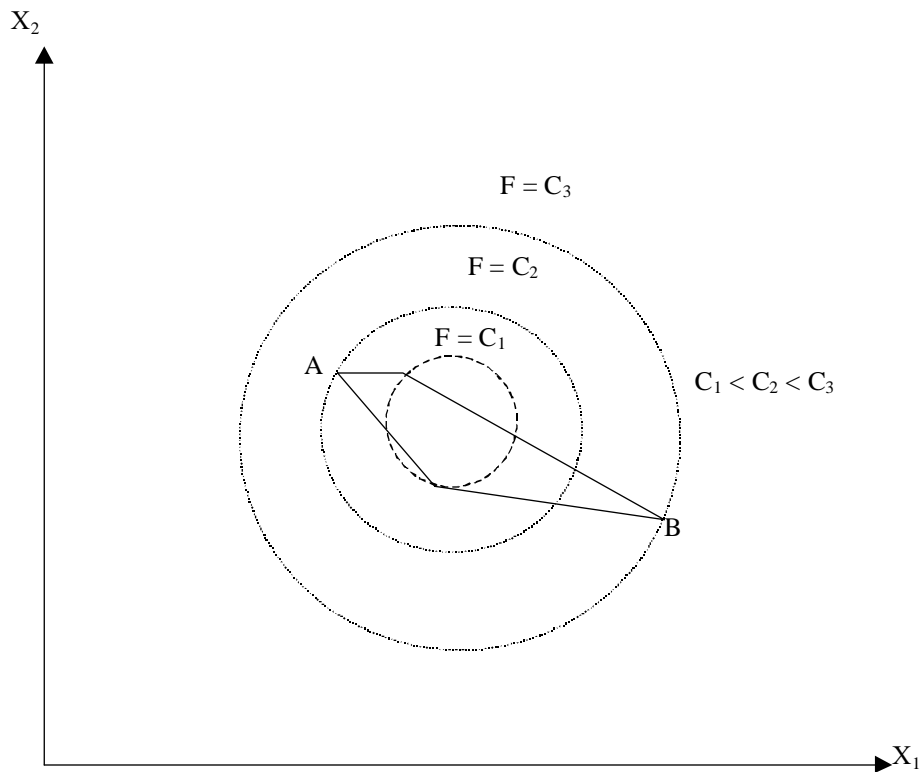


Рис. 3.4. Два экстремума при односвязной области допустимых значений

3.2. Классические методы определения экстремумов функции

3.2.1. Задача на абсолютный экстремум

Напомним, что если непрерывная функция n переменных $\mathbf{x} = (x_1, \dots, x_n)$ $F(\mathbf{x})$ имеет в точке $\mathbf{x}_{\text{опт}}$ максимум, то существует $\varepsilon > 0$ такое, что для всех \mathbf{x}

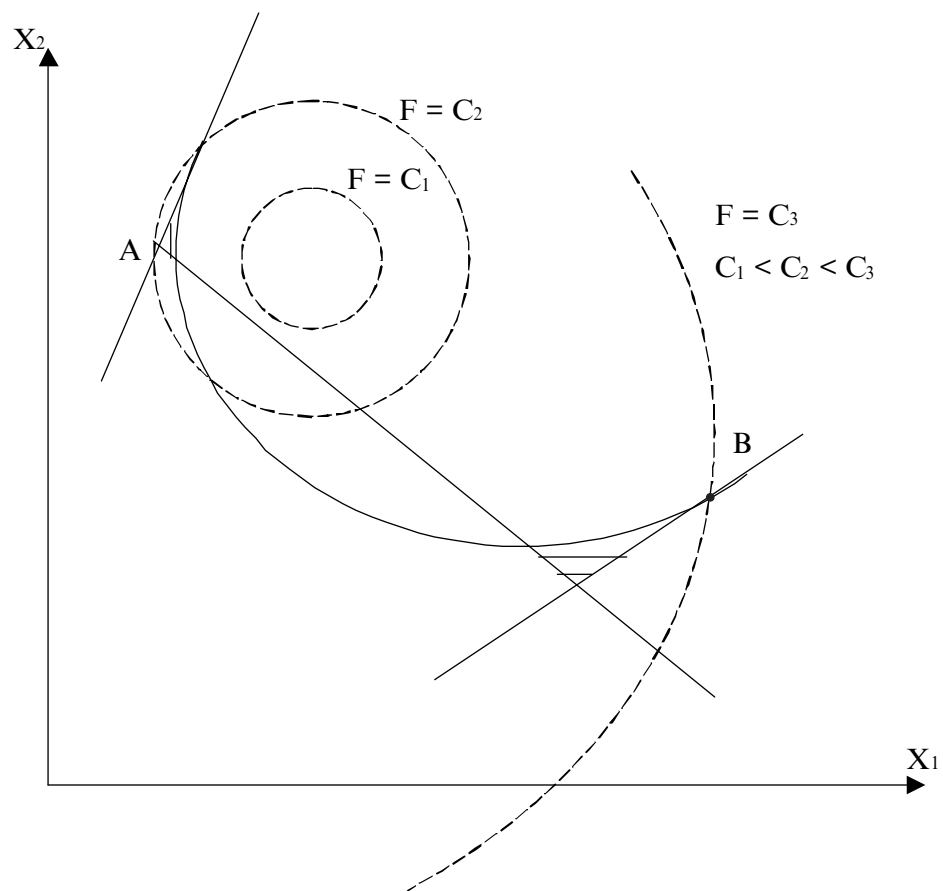


Рис. 3.5. Два экстремума при двусвязной области допустимых значений

из ε -окрестности точки $\mathbf{x}_{\text{опт}}$

$$F(\mathbf{x}) \leq F(\mathbf{x}_{\text{опт}}).$$

Выберем два вида приращения x_j вдоль j -й координаты:

$$\begin{aligned} \Delta x_j &= x_j - x_{j \text{ опт}} > 0; \\ \Delta x_j &= x_j - x_{j \text{ опт}} < 0. \end{aligned}$$

Тогда:

$$[F(x_j) - F(x_{j \text{ опт}})] / \Delta x_j \leq 0 \quad \text{при } \Delta x_j > 0;$$

$$[F(x_j) - F(x_{j \text{ опт}})] / \Delta x_j \geq 0 \quad \text{при } \Delta x_j < 0.$$

Переходя в этих соотношениях к пределу при $x_j \rightarrow 0$, получаем:

$$\partial F(x_{j \text{ опт}}) / \partial x_j \leq 0; \quad \partial F(x_{j \text{ опт}}) / \partial x_j \geq 0. \quad (3.5)$$

Из этих соотношений следует (условия Ферма), что

$$\partial F(x_{j \text{ опт}}) / \partial x_j = \partial F(\mathbf{x}_{\text{опт}}) / \partial x_j = 0; \quad j = 1, 2, \dots, n. \quad (3.6)$$

Аналогичное соотношение можно получить для случая минимума функции. Таким образом, доказана необходимость условий (3.6) для достижения в точке $\mathbf{x}_{\text{опт}}$ максимума или минимума функции $F(\mathbf{x})$, т.е. если имеется экстремум, то условия (3.6) удовлетворяются. Но равенство нулю всех производных в точке $\mathbf{x}_{\text{опт}}$ еще не обеспечивает существования в ней экстремума, т.е. условия (3.6) не являются достаточными. Геометрически это означает, что в случае нулевой производной от функции одной переменной может иметь место точка перегиба, а не максимум (или минимум), а в случае функции двух переменных – седловая точка, а не экстремум и т. д. Поэтому точки $\mathbf{x}_{\text{опт}}$, в которых выполняются соотношения (3.6), называются **стационарными**.

3.2.2. Задача на условный экстремум

Заметим, что условие (3.6) удалось получить благодаря возможности придавать переменной x приращения двух знаков, откуда и возникли два неравенства (3.5). Если допустимая область значений \mathbf{x} ограничена неотрицательными значениями $\mathbf{x} \geq 0$, то внутри области, где $\mathbf{x} > 0$, справедливость условия (3.6) сохраняется, так как там допустимы приращения обоих знаков. На границе области $\mathbf{x} \geq 0$, где $\mathbf{x} = 0$, допускается только положительное приращение $\Delta \mathbf{x} > 0$, можно говорить только об односторонней производной, и из (3.6) следует следующее необходимое условие максимума:

$$\partial F(\mathbf{x}_{\text{опт}}) / \partial x_j \leq 0.$$

Необходимое условие минимума на границе области $x_j = 0$ запишется в виде

$$\partial F(\mathbf{x}_{\text{опт}}) / \partial x_j \geq 0.$$

При определении условного экстремума функции, когда требуется определить максимум (или минимум) функции $F(\mathbf{x})$ при ограничивающих условиях:

$$\varphi_i(\mathbf{x}) = b_i, \quad i = 1, \dots, m,$$

т.е.

$$\begin{aligned} F(\mathbf{x}) &= \max; \\ \varphi_i(\mathbf{x}) &= b_i, \end{aligned}$$

используется также метод множителей Лагранжа, который, так же как в случае классического вариационного исчисления (см. юниту 1), заключается во введении функции Лагранжа

$$\Phi = F(\mathbf{x}) + \sum_{i=1}^m \lambda_i [b_i - \varphi_i(\mathbf{x})], \quad (3.7)$$

где λ_i – неопределенные множители Лагранжа.

Полагая, что функция является частным случаем функционала, получаем, что необходимые условия экстремума находятся прямым дифференцированием соотношения (3.7) и записываются в виде

$$\partial\Phi/\partial x_j = \partial F/\partial x_j + \sum \lambda_i (\partial\varphi_i / \partial x_j) = 0, \quad j = 1, \dots, n; \quad (3.8)$$

$$\partial\Phi/\partial \lambda_i = b_i - \varphi_i(\mathbf{x}) = 0, \quad i = 1, \dots, m. \quad (3.9)$$

Если ввести в рассмотрение векторы

$$\begin{aligned} \lambda &= (\lambda_1, \dots, \lambda_m); \\ \varphi &= (\varphi_1, \dots, \varphi_m); \\ \mathbf{b} &= (b_1, \dots, b_m); \end{aligned}$$

$$\text{grad } f(\mathbf{x}) = \{\partial f/\partial x_1, \partial f/\partial x_2, \dots, \partial f/\partial x_n\},$$

соотношения (3.8) и (3.9) переписуются как условия Лагранжа:

$$\begin{aligned} \text{grad } \Phi &= \text{grad } F - \lambda \text{ grad } \varphi = 0; \\ \mathbf{b} - \varphi &= 0, \end{aligned}$$

где равенство нулю векторов понимается покомпонентно.

В случае $n = 2$ и $m = 1$ геометрическая задача об отыскании условного экстремума сводится к отысканию точки касания A кривой $\varphi = b$ к одной из кривых постоянного уровня $F = \text{const}$.

3.3. Целочисленное программирование

3.3.1. Особенности задач целочисленного программирования

Математическая формулировка задач целочисленного программирования аналогична задачам нелинейного программирования:

$$\min\{F(\mathbf{x}) \mid x_j \geq 0; \quad j = 1, 2, \dots, n; \quad \varphi_i(\mathbf{x}) \leq 0; \quad i = 1, 2, \dots, m\}. \quad (3.10)$$

Однако специфика задач целочисленного программирования заключается в том, что переменные x_j и функции $F(\mathbf{x})$, $\varphi_i(\mathbf{x})$ могут принимать только дискретные значения. Специальными преобразованиями в подавляющем большинстве случаев можно свести эти дискретные значения к целочисленным.

В общем случае целочисленными могут быть не все, а части переменных и функций. Иногда целочисленное программирование называется дискретным программированием. Однако, как уже указывалось выше, при полном рассмотрении всех методов оптимизации оказывается, что этот термин уже используется в другом смысле.

Если все функции $F(\mathbf{x})$, $\varphi_i(\mathbf{x})$ в задаче (3.10) линейные, то имеет место линейная задача целочисленного программирования. Методы решения таких задач получили наибольшее распространение. Очевидно, что могут встречаться задачи с целочисленными переменными x_j и функциями $F(\mathbf{x})$, $\varphi_i(\mathbf{x})$ и целочисленными переменными и непрерывными функциями (рис. 3.6). Как правило, именно этот последний случай рассматривается в большинстве руководств. Кроме того, встречаются задачи частично и полностью целочисленные в зависимости от того, часть или все переменные x_j принимают целочисленные значения. Как уже отмечалось, допустимые дискретные



Рис. 3.6. Классификация задач целочисленного программирования

значения, входящие в множество, могут быть даже нецелочисленными. В частности, это множество может быть бесконечным, конечным или даже состоящим всего из двух значений: 0 и 1. В этом случае имеет место **целочисленное программирование с булевыми переменными**, методы которого в значительной мере перекрываются логическим синтезом конечных автоматов.

Методы целочисленного программирования значительно отличаются от методов оптимизации, рассмотренных ранее, и по существу своему относятся к дискретной математике. Они не обладают таким единством, как методы вариационного исчисления, и в большинстве представляют собой набор частных приемов, пригодных для решения частных задач. Однако актуальность этих методов требует их дальнейшего развития и совершенствования, так как наиболее важные прикладные задачи типа оперативно-календарного планирования сводятся к задачам целочисленного программирования.

Со значительными оговорками все методы решения задач целочисленного программирования можно разделить на четыре группы (рис. 3.7):

- методы отсечения;
- комбинаторные методы;
- приближенные методы;
- другие методы, которые нельзя отнести ни к одной из трех первых групп.

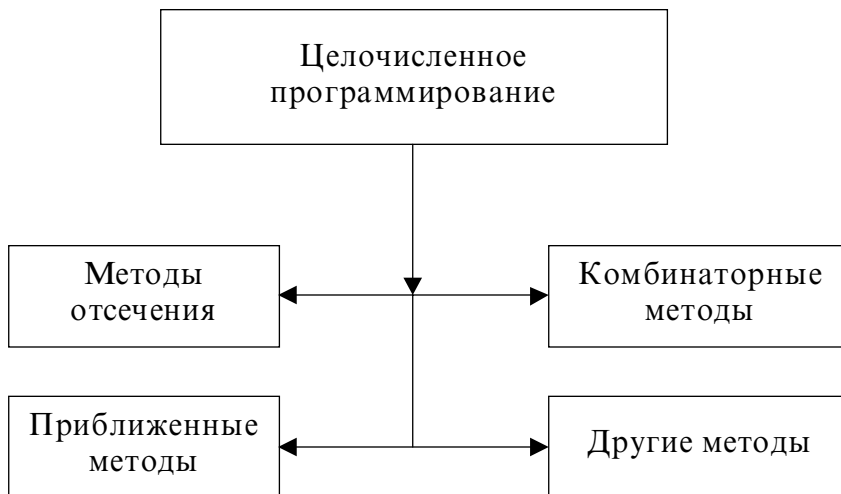


Рис. 3.7. Классификация методов решения задач целочисленного программирования

Первая группа использует процедуру линейного программирования для последовательности задач, в которые постепенно вводятся линейные ограничения, и тем самым реализуется процесс правильного отсечения. Основу всех методов этой группы составляют три алгоритма Гомори и их модификации.

Большинство **комбинаторных методов** не используют симплекс-методы, а достигают сокращения поиска оптимальных значений анализом исходного множества.

Дискретное динамическое программирование по-прежнему успешно применяется для решения задач целочисленного программирования и в приведенной на рис. 3.7 классификации содержится в разделе комбинаторных методов, так как по существу оно является направленным методом перебора. Значительно реже используется дискретный принцип максимума. Широкие возможности в части решения прикладных задач большой размерности открываются при использовании человеко-машинных методов оптимизации. По своей идеологии эти методы примыкают к лингвистическим с добавлением блоков эвристики, в качестве которых выступает человек при общении с ЭВМ.

Характерная особенность общей задачи целочисленного программирования заключается в ее нерегулярности. Под **регулярностью** понимают совокупность необходимых и достаточных условий экстремума, которые позволяют создать конечную процедуру его отыскания. Условиям регулярности удовлетворяют задачи линейного и выпуклого программирования. В регулярных задачах локальный экстремум совпадает с глобальным. Указанные выше

обстоятельства определяются характером области допустимых значений, которая становится многосвязной из-за требования целочисленности (дискретности), а также невыпуклой. Несовпадение целочисленного и нецелочисленного экстремумов рассматривается в следующем примере.

Пример. Требуется найти решение следующей задачи целочисленного программирования:

$$\begin{aligned} \frac{1}{2} x_1 - x_2 &\geq 2; \\ \frac{1}{5} x_1 + x_2 &\geq 3; \\ x_1 - \frac{1}{3} x_2 &\geq 4; \\ x_1, x_2 &\geq 0; \\ \max z &= \frac{1}{5} x_1 + x_2. \end{aligned}$$

При целочисленности переменных максимум $z_{\text{опт}} = 13/5$ достигается в точке $x_{1\text{опт}} = 3, x_{2\text{опт}} = 2$, в то время как при исключении этого требования $z_{\text{опт}} = 3$ в точке $x_{1\text{опт}} = 10/7, x_{2\text{опт}} = 19/7$. Округление результата до целых не дает оптимума исходной задачи, как это видно из рис. 3.8. Заметим, что условие целочисленности z при снятии этого требования к переменным x_1 и x_2 определяет оптимум при других значениях x_1 и x_2 .

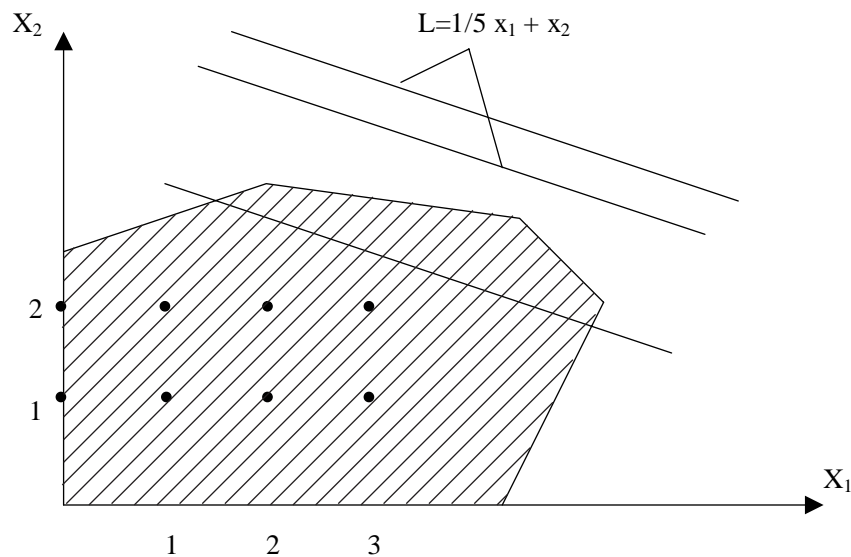


Рис. 3.8. К особенностям задач целочисленного программирования

Из этого простейшего примера виден комбинаторный характер задач целочисленного программирования, который часто требует прямого перебора. Задача дискретного нецелочисленного программирования может быть сведена к задаче целочисленного программирования. Допустим, имеется задача математического программирования в виде

$$\min\{F(\mathbf{x}) \mid x_j \geq 0; \quad j = 1, 2, \dots, n; \quad \varphi_i(\mathbf{x}) \leq 0; \quad i = 1, 2, \dots, m\}.$$

Будем считать, что переменная x_{j_0} может принимать только одно из заданного множества значений

$$x_{j_0} \in \{k_{j_0}^1, k_{j_0}^2, \dots, k_{j_0}^p\}.$$

Введем дополнительные булевы переменные y_1, y_2, \dots, y_p и запишем условие дискретности для x_{j_0} в виде:

$$\begin{aligned} x_{j_0} &= k_{j_0}^1 y_1 + k_{j_0}^2 y_2 + \dots + k_{j_0}^p y_p; \\ y_j &= 1 \text{ или } 0, \quad j = 1, 2, \dots, p; \\ y_1 + y_2 + \dots + y_p &= 1. \end{aligned} \quad (3.11)$$

Последнее соотношение требует, чтобы только одна переменная y_j была отлична от нуля. Нетрудно убедиться, что формулы (3.11) обеспечивают дискретность переменной x_{j_0} . Таким образом, путем введения булевых переменных y_j можно всегда свести нецелочисленную дискретную задачу математического программирования к целочисленной. Как видно, соотношения (3.11) представляют собой p -кратную логическую альтернативу:

$$\langle\langle x_{j_0} = k_{j_0}^1, \text{ или } x_{j_0} = k_{j_0}^2, \text{ или } \dots, \text{ или } x_{j_0} = k_{j_0}^p \rangle\rangle.$$

Для лучшего понимания связи задач целочисленного программирования с задачами на многосвязных и невыпуклых областях покажем, как эти последние сводятся к задачам целочисленного программирования. Пусть допустимая область изменения переменной x_{j_0} многосвязная и задается соотношениями:

$$\begin{aligned} 0 &\leq x_{j_0} \leq k_{j_0}; \\ \text{или } x_{j_0} &\leq a \text{ или } x_{j_0} \geq b; \\ 0 &\leq a < b \leq k_{j_0} \end{aligned} \quad (3.11a)$$

Введем дополнительную булеву переменную y_{j_0} , которую не будем включать в функцию цели, а заменим рассмотренные соотношения следующими:

$$\begin{aligned} x_{j_0} - b + by_{j_0} &\geq 0; \\ -x_{j_0} + ay_{j_0} + k_{j_0}(1 - y_{j_0}) &\geq 0; \\ y_{j_0} &= 1 \text{ или } 0. \end{aligned} \quad (3.11б)$$

Очевидно, что соотношения (3.11a) и (3.11б) эквивалентны. Действительно, когда $y_{j_0} = 1$, то (3.11б) сводится к $x_{j_0} \geq 0$ и $x_{j_0} \leq a$; когда $y_{j_0} = 0$, то $x_{j_0} \geq b$ и $x_{j_0} \leq k_{j_0}$. Таким образом, многосвязность области допустимых значений легко учитывается введением дополнительных булевых переменных. Этот метод без труда распространяется на случай многих переменных.

Процедура сведения невыпуклой области изменения переменных к стандартным ограничениям состоит в следующем. Область разбивается на выпуклые подобласти, и между некоторыми парами их вводятся альтернативные условия (дизъюнкции). Для каждой подобласти определяется верхняя граница и вводятся дополнительные ограничения, включающие для каждой

пары подобластей булевы переменные.

Пример. Рассмотрим процедуру сведения задачи невыпуклого программирования к целочисленному программированию. Пусть область допустимых значений задана системой неравенств:

$$\begin{aligned}x_1 + 1/4 x_2 &\leq 2; \\ 2/5 x_1 - x_2 &\geq -1; \\ x_1 - 1/3 x_2 &\leq 3; \\ x_1, x_2 &\geq 0.\end{aligned}$$

Соответствующая область представлена на рис. 3.9. Первая и вторая пара неравенств образуют выпуклые подобласти x_1, x_2 с учетом неотрицательности переменных с верхними границами $x_{1\text{макс}}, x_{2\text{макс}}$ соответственно. Введение булевой переменной y приводит к дополнительным неравенствам:

$$\begin{aligned}x_1 - yx_{\text{макс}} &\leq 0; \\ x_2 - (1 - y)x_{\text{макс}} &\leq 0.\end{aligned}$$

Здесь каждое условие относится ко всем неравенствам соответствующей области.

Нетрудно убедиться, что аналогичный способ применим в случае числа переменных, большего двух.

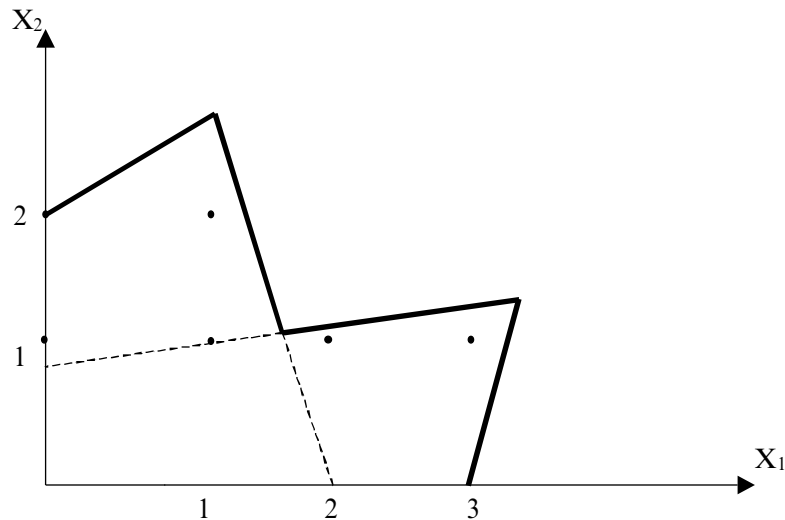


Рис.3.9. Невыпуклая область, состоящая из выпуклых многогранников

3.3.2. Нелинейное и целочисленное программирование

Можно показать, что любая задача выпуклого нелинейного программирования может быть приближенно сведена к соответствующей задаче целочисленного программирования. Допустим, что функция цели $F(\mathbf{x})$ имеет вид сепарабельной функции

$$F(x_1, x_2, \dots, x_n) = \sum_{j=1}^n F_j(x_j). \quad (3.12)$$

Будем считать, что каждое слагаемое $F_j(x_j)$ представляет собой выпуклую функцию, и в дальнейшем изложении опустим индекс j , используя обозначение $F(x)$. Разобьем весь интервал изменения x на участки $\Delta_k = [d_{k-1}, d_k]$ длиной $h_k = d_k - d_{k-1}$. Если ввести новые переменные, равные длине пересечения отрезка $[0, x]$ с отрезком Δ_k , т.е.

$$0 \leq y_k \leq h_k, \quad k = 1, 2, \dots, p, \quad (3.13)$$

то

$$x = y_1 + y_2 + \dots + y_p.$$

При этом функция $F(\mathbf{x})$ может быть приближенно заменена на

$$F \sim (\mathbf{x}) = c_0 + c_1 y_1 + c_2 y_2 + \dots + c_p y_p. \quad (3.14)$$

Для выпуклой функции $F(\mathbf{x})$ коэффициенты c образуют монотонную последовательность

$$c_1 \leq c_2 \leq \dots \leq c_p. \quad (3.15)$$

Задача минимизации функции $F(\mathbf{x})$, задаваемой формулой (3.14) при условиях (3.13), является задачей частично целочисленного программирования. Частичную целочисленность задачи можно пояснить следующим образом. Прежде всего, заметим, что оптимальное решение линейной задачи можно записать в виде

$$x_{\text{опт}} = y_1 + y_2 + \dots + y_{k_0}, \quad k_0 \leq p, \quad (3.16)$$

где $y_1 = h_1, \dots, y_{k_0-1} = h_{k_0-1}, 0 < y_{k_0} < h_{k_0}$, а все y_k для $k > k_0$ равны нулю. Здесь по существу намечена некоторая процедура поиска оптимального решения. В начале y_1 увеличивается до своего максимального значения. Если экстремум не достигнут, то, зафиксировав y_1 на максимальном значении, увеличивают y_2 от нулевого значения $y_2 = 0$ до максимального и т. д., пока не дойдут до y_{k_0} . Если y_{k_0} взять максимальной, то нарушится соотношение (3.16). Необходимо эту переменную уменьшить до значения, удовлетворяющего соотношению (3.16). Остальные y_k ($k > k_0$) следует положить равными нулю, что является следствием выпуклости функции цели, которая обуславливает соотношения (3.15). В соответствии с этим условием нецелесообразно брать переменную y_{k+1} , отличную от нуля, пока y_k не достигла своего максимального значения. Формально последнее утверждение может быть записано с помощью следующих логических соотношений, носящих альтернативный характер:

$$<< \text{или } y_k - h_k = 0, \text{ или } y_{k+1} = 0 >>. \quad (3.17)$$

Это условие в рассмотренной выше задаче минимизации (3.14) при условиях (3.13) не является дополнительным и автоматически выполняется при соблюдении условий (3.13).

Рассмотрим теперь невыпуклую функцию $F(x)$ (рис. 3.10). Опять заменим исходную функцию цели приближенно кусочно-линейной функцией в соот-

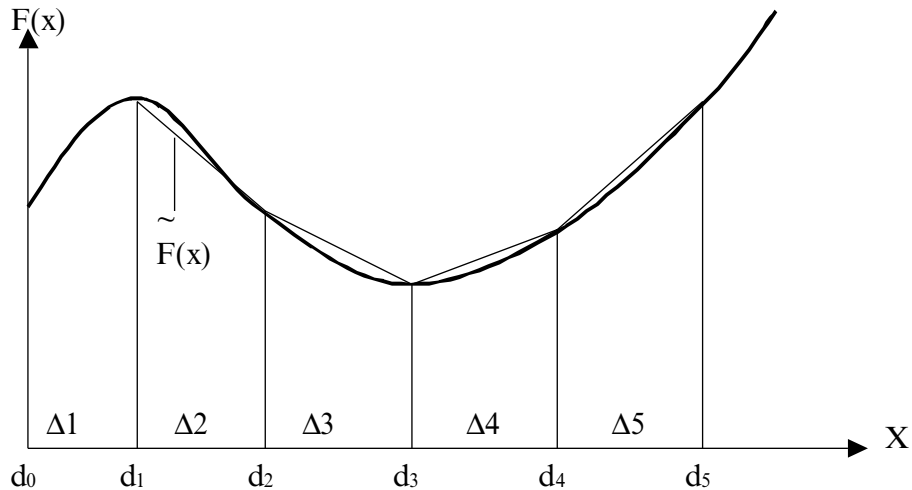


Рис 3.10. К методу сведения задач невыпуклого программирования к задачам целочисленного программирования

ветствии с формулой (3.14). При этом не соблюдается условие монотонности коэффициентов c_k [см. формулу (3.15)]. Можно попытаться довести до верхней границы вначале значение y_k , соответствующее наименьшему c_k , затем значение y_k , соответствующее следующему по величине c_k и т. д., однако при этом отрезки Δx , из которых комплектуется оптимальное значение x , получаются несвязными. Для обеспечения их связности необходимо условие (3.17) ввести как обязательное, так как оно в данном случае автоматически не выполняется. Сведем его к целочисленной программе. Для этого воспользуемся дополнительными переменными

$$z_k = 0 \text{ или } 1, \quad k = 1, 2, \dots, p. \quad (3.18)$$

Тогда условие (3.17) можно заменить соотношениями:

$$\begin{aligned} y_k &\geq h_k z_k; \\ y_{k+1} &\leq h_{k+1} z_k; \\ k &= 1, 2, \dots, p-1. \end{aligned} \quad (3.19)$$

Нетрудно убедиться, что $z_k = 1$ соответствует достижению y_k своей верхней границы, так как первое соотношение (3.19) превращается в $y_k \geq h_k$, т.е. $y_k = h_k$ (переменная y_k по условию (3.13) не может быть больше h_k). Значение z_k соответствует условию $y_k < h_k$, так как второе соотношение (3.19) превращается в $y_{k+1} \leq 0$, которое в силу (3.13) означает, что $y_{k+1} = 0$. Далее, замечая, что ограничения, наложенные сверху на y_k в соотношениях (3.13), уже содержатся в формулах (3.19), кроме ограничения на y_1 , можно заменить ограничения (3.13) и (3.19) на следующие:

$$y_k \geq 0, \quad k = 1, 2, \dots, p; \quad (3.20)$$

$$\begin{aligned}
y_1 &\leq h_1; \\
y_k &\geq h_k z_k; \\
y_{k+1} &\leq h_k z_k; \\
k &= 1, 2, \dots, p-1.
\end{aligned}
\tag{3.21}$$

В итоге исходная задача невыпуклого программирования свелась к задаче целочисленного программирования:

$$\begin{aligned}
F^*(\mathbf{x}) &= c_0 + c_1 y_1 + c_2 y_2 + \dots + c_p y_p = \min \\
y_k &\geq 0; \\
x &= y_1 + y_2 + \dots + y_p; \\
z &= 0 \text{ или } 1; \\
y_1 &\leq h_1; \\
y_k &\geq h_k z_k; \\
y_{k+1} &\leq h_k z_k; \\
k &= 1, 2, \dots, p.
\end{aligned}
\tag{3.22}$$

Кроме того, если в исходной нелинейной программе имеют место ограничения

$$\varphi_i(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m,$$

их следует добавить к ограничениям (3.22).

Приведенный выше способ сведения невыпуклой нелинейной программы к частично целочисленной программе реализуется за счет введения большого числа дополнительных переменных и ограничений.

3.4. Методы отсечения

В данном подразделе будет рассмотрена линейная задача целочисленного программирования и дано понятие первого алгоритма Гомори, который заложил фундамент теории целочисленного программирования.

Для удобства изложения методов целочисленного программирования используют специфические обозначения, на которых целесообразно прежде всего остановиться. Задачу целочисленного программирования рассмотрим в виде:

$$x_0 \equiv \sum_{j=1}^n c_j x_j = \max; \tag{3.23}$$

$$\sum_{j=1}^n a_{ij} x_j = b_i; \quad i = 1, 2, \dots, m; \tag{3.24}$$

$$x_j \geq 0; \quad j = 1, 2, \dots, n; \tag{3.25}$$

$$x_j - \text{целое}; \quad j = 1, 2, \dots, n_1. \tag{3.26}$$

При $n_1 = n$ задача (3.23) – (3.26) является полностью, а при $n_1 < n$ частично целочисленной задачей линейного программирования. По аналогии с задачей линейного программирования введем ряд понятий. Множество наборов (x_1, x_2, \dots, x_n) , удовлетворяющих условиям (3.24) – (3.26), называется **областью определения или допустимой областью задачи целочисленного программирования** и обозначается через L^u . Для сокращенного обозначения задачи (3.23) – (3.26) применяется запись (L^u, c) , где c – вектор-столбец, состоящий из элементов c_i . **Оптимальное решение или оптимальный план** обозначается через x (L^u, c). Наряду с оптимальным планом $x_{\text{опт}} = (x_{1\text{опт}}, x_{2\text{опт}}, \dots, x_{n\text{опт}})$ будет рассматриваться расширенный оптимальный план

$$x'_{\text{опт}} = (x_{0\text{опт}}, x_{1\text{опт}}, \dots, x_{n\text{опт}}),$$

где

$$x_0 = \sum_{j=1}^n c_j x_j.$$

Сокращенно оптимальный расширенный план обозначается как

$$x' (L^u, c).$$

В целочисленном программировании рассматриваются целочисленные многогранники, под которыми понимаются многогранники с целочисленными вершинами, и целочисленные симплекс-таблицы, состоящие из целочисленных элементов. Заметим, что множество точек, ограниченное целочисленным многогранником, не обязательно состоит из целочисленных точек, требуется только целочисленность вершин (крайних точек) многогранника.

Основная идея методов отсечения заключается в построении такой эквивалентной задачи линейного программирования (A, c) , при которой исходная задача целочисленного программирования (L^u, c) сводится к ее решению. Иногда это называют линейной аппроксимацией целочисленного программирования (L^u, c) .

Введем понятие правильного отсечения, физически означающее выбор такой гиперплоскости, которая разделяет оптимальные решения целочисленной линейной задачи x (L^u, c) и задачи линейного программирования x (L, c), причем последняя не удовлетворяет условию целочисленности

$$x (L, c) \not\subset L^u.$$

Математически правильное отсечение означает выбор такого числа β , при котором неравенство

$$\sum_j a_j x_j \leq \beta \quad (3.27)$$

выполняется при условиях

$$ax(L, c) > \beta; \quad (3.28)$$

$$\{ x (L^u, c) \mid ax \leq b \} \subset L^u, \quad (3.29)$$

где a – вектор-строка, состоящая из элементов.

Метод отсечений позволяет построить процедуру последовательного выделения оптимального решения целочисленной задачи по известному решению соответствующей линейной задачи, в которой исключены требования целочисленности. Эта процедура решения задачи (L^u, c) может быть описана следующим образом:

1. На k -м этапе решается вспомогательная задача линейного программирования (L_k, c) , $k = 0, 1, 2$, где $L_0 = L$.

2. Если на k -м этапе оптимальное решение $x(L_k, c)$ удовлетворяет условию целочисленности, то оно будет также оптимальным решением $x(L_0^u, c)$ исходной задачи (L_0^u, c) , так как на всех этапах множества целочисленных точек совпадают, т.е.

$$L_0^u = L_1^u = L_2^u = \dots,$$

что является специальным условием при отсечении.

3. Если на k -м этапе решение не удовлетворяет условиям целочисленности, то $x(L_k, c)$ не является оптимальным решением исходной целочисленной задачи. В этом случае переходят от k -го этапа к $(k + 1)$ -му, добавляя к линейным ограничениям задачи (L_k, c) условие правильного отсечения

$$a^k x \leq \beta^k, \quad (3.30)$$

которое превращает многогранник (множество) L_k в многогранник L_{k+1} . Каждый конкретный метод имеет свой способ задания отсечения, но в **алгоритмах Гомори** гарантируется конечное число процедур.

В этом алгоритме впервые была реализована идея метода отсечения. Он дает конечную процедуру решения полностью целочисленной задачи линейного программирования, заданной в виде:

$$x_0 \equiv \sum_{j=1}^n c_j x_j = \max; \quad (3.31)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i; \quad i = 1, 2, \dots, m; \quad (3.32)$$

$$x_j \geq 0; \quad j = 1, 2, \dots, n; \quad (3.33)$$

$$x_j - \text{целое}. \quad (3.34)$$

3.5. Комбинаторные методы

3.5.1. Решение задач целочисленного программирования с помощью динамического программирования

Все комбинаторные методы решения задач целочисленного программирования основаны на той или иной идее направленного перебора вариантов, в результате которого путем перебора сокращенного числа допустимых решений отыскивается оптимальное решение. Перебор осуществляется с помощью определенного набора правил, которые позволяют исключать подмножества вариантов, не содержащие оптимальной точки.

В целом эти методы легче справляются с проблемой округлений, чем методы отсечения, как правило, не используют симплекс-процедуру линейного

программирования и имеют более “простую арифметику” и более “сложную логику”.

Основное содержание этих методов составляют динамическое программирование и совокупность способов решения, объединенных общим термином – метод “ветвей и границ”.

Динамическое программирование является универсальным методом отыскания глобального экстремума любых задач, для которых справедлив принцип оптимальности. Принцип оптимальности заведомо применим к сепарабельным и линейным функциям цели.

Ограничения в виде неравенств и целочисленности учитываются в рекуррентном процессе решения функционального уравнения Беллмана.

Для случая сепарабельной функции цели можно составить общую вычислительную схему решения задачи нелинейного целочисленного программирования по методу динамического программирования. Причем размерность задачи динамического программирования определяется характером функции цели и ограничений.

Основная трудность при использовании метода динамического программирования заключается в сведении исходной задачи к модели многоэтапного процесса оптимизации, используемого в этом методе, с введением функции Беллмана. В каждой конкретной задаче при этом используются свои приемы, комплексом которых можно овладеть после известного опыта.

На примере задачи об опасной переправе (пример взят из: Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М., 1965) рассмотрим применение этого метода. Группа, состоящая из миссионеров и людоедов, должна переправиться через реку на лодке, которая имеет ограниченную грузоподъемность R . Лодка управляется одним или более пассажирами в любой комбинации. На каждом берегу и в лодке должно соблюдаться условие нелюдоедства, заключающееся в том, чтобы число людоедов в группе не превышало числа миссионеров ($m \geq l$).

Обозначим число миссионеров и людоедов на левом берегу через m_1 и l_1 , на правом берегу m_2 и l_2 и в лодке m и l . Соответственно сформулируем более общие условия нелюдоедства:

$$\begin{aligned} R_1(m_1, l_1) &\geq 0 \text{ на левом берегу;} \\ R_2(m_2, l_2) &\geq 0 \text{ на правом берегу;} \\ R_3(m, l) &\text{ в лодке.} \end{aligned}$$

Составим функциональное уравнение Беллмана, для чего введем функцию $S_N(m_1, l_1)$, равную максимальному числу людей на правом берегу после N шагов при условии, что процесс перевозки начинался с m_1 миссионеров и l_1 людоедов на левом берегу и m_2 и l_2 на правом берегу. Считается, что на любом шаге ни один человек не возвращается на первый берег со второго, если все уже находятся на втором берегу.

Один шаг заключается в перевозке x_1 миссионеров и y_1 людоедов с левого на правый берег и в обратной перевозке x_2 миссионеров и y_2 людоедов с правого на левый берег.

В соответствии с принципом оптимальности для $N \geq 0$ можно написать:

$$S_N(m_1, l_1) = \max_{x_1, y_1} S_{N-1}(m_1 - x_1 + x_2, l_1 - y_1 + y_2),$$

где величины x_1, y_1, x_2, y_2 должны удовлетворять следующим условиям:

$$\begin{aligned}
0 \leq x_1 \leq m_1, & \quad 0 \leq y_1 \leq l_1, \\
0 \leq x_2 \leq m_2 + x_1, & \quad 0 \leq y_2 \leq l_2 + y_1; \\
0 < x_1 + y_1 \leq K, & \quad 0 < x_2 + y_2 \leq K; \\
R_3(x_1, y_1) \geq 0, & \quad R_3(x_2, y_2) \geq 0. \\
R_1(m_1 - x_1, l_1 - y_1) \geq 0; \\
R_1(m_1 - x_1 + x_2, l_1 - y_1 + y_2) \geq 0; \\
R_2(m_2 + x_1, l_2 + y_1) \geq 0. \\
R_2(m_2 + x_1 - x_2, l_2 + y_1 - y_2) \geq 0;
\end{aligned} \tag{3.35}$$

Для $N = 1$ имеем:

$$S_1(m_1, l_1) = \max_{x_1, y_1} [(m_2 + x_1), (l_2 + y_1)]. \tag{3.36}$$

Минимальное число перевозок равно такому N , для которого

$$S_N = m_1 + m_2 + l_1 + l_2. \tag{3.37}$$

Пример. Допустим, что суммарное количество миссионеров $m_1 + m_2 = 3$ и суммарное количество людоедов $l_1 + l_2 = 3$, грузоподъемность лодки $K = 2$.

Обозначим через (i, j) состояние, при котором i миссионеров и j людоедов находятся на левом берегу, а $3 - i$ и $3 - j$ – на правом берегу. Тогда с точки зрения нелюдоедства допустимы только следующие состояния:

$(0, 1), (1, 1), (3, 1), (0, 2), (2, 2), (3, 2), (0, 3), (3, 3)$.

Используя функциональное уравнение, получаем:

$$\begin{aligned}
S_1(0, 1) = 6, \quad S_1(1, 1) = 6, \quad S_1(3, 1) = 2, \quad S_1(0, 2) = 6, \\
S_1(2, 2) = 3, \quad S_1(3, 2) = 2, \quad S_1(0, 3) = 4, \quad S_1(3, 3) = 1.
\end{aligned}$$

Если

$$S_v(i, j) = 6, \quad \text{то } S_{v+\mu}(i, j) = 6, \quad \mu = 1, 2, \dots$$

Учитывая это обстоятельство, с помощью рекуррентных функциональных уравнений получаем:

$$\begin{array}{lll}
S_2(3, 1) = 3, & S_2(2, 2) = 4, & S_2(3, 2) = 2, \\
S_2(0, 3) = 6, & S_2(3, 3) = 2, & S_3(3, 1) = 4, \\
S_3(2, 2) = 6, & S_3(3, 2) = 3, & S_4(3, 2) = 4, \\
S_4(3, 3) = 3, & S_5(3, 2) = 6, & S_5(3, 3) = 2, \\
S_5(3, 1) = 6, & S_5(3, 3) = 4, & S_6(3, 3) = 6.
\end{array}$$

Отсюда следует, что минимально требуемое число перевозок равно шести.

Процедуру решения рассмотренной задачи легко проследить с помощью графа, представленного на рис. 3.11. Каждое ребро графа соответствует перевозке с одного берега на другой. Таким образом, одному этапу динамического программирования соответствуют два ребра графа. Стрелки указывают направление действия. Цифры в скобках около каждой вершины означают: первая – число миссионеров, вторая – число людоедов на одном из берегов, последняя цифра обозначает левый (1) и правый (0) берега; две

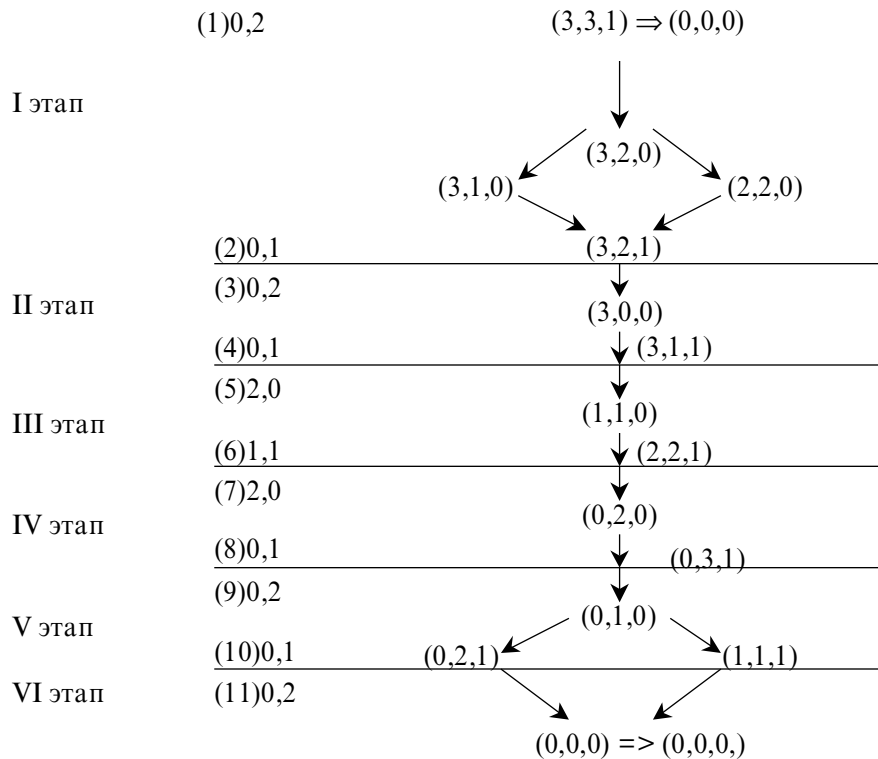


Рис. 3.11. Пояснение к задаче о миссионерах и людоедах

цифры слева соответствуют числу перевозимых миссионеров и людоедов.

Таким образом, процедура направленного поиска оптимального решения заключается в следующем:

- 1) перевозим двух людоедов с левого берега на правый;
- 2) возвращаем одного людоеда назад на левый берег;
-
- 6) перевозим одного миссионера и одного людоеда с правого на левый берег;
-
- 11) перевозим двух людоедов с левого берега на правый.

Из рис. 3.11 видно, что в проблеме миссионеров и людоедов существует более одного оптимального пути решения. Так, на первом этапе можно или перевезти двух людоедов и затем одного людоеда обратно или одного миссионера и одного людоеда, а потом одного миссионера обратно. Число перевозок и шагов поиска решения будет одинаковым.

3.5.2. Метод ветвей и границ

Комплексный подход по схеме ветвлений объединяет целый ряд близких комбинаторных методов, как-то: метод ветвей и границ, метод последовательного конструирования, анализа и отсева вариантов и др. Общая идея метода

проста. Множество допустимых планов некоторым способом разбивается на подмножество. В свою очередь каждое из подмножеств по этому же или другому способу снова разбивается на подмножества до тех пор, пока каждое подмножество не будет представлять собой точку в многомерном пространстве. В силу конечности наборов значений переменных дерево подмножеств (схема ветвлений) конечно.

Построение схемы ветвления есть не что иное, как формирование процедуры перебора. Перебор может осуществляться различными способами. Сечение исходной допустимой области G_0 гиперплоскостью на части G_{11} и G_{12} с последующим разделением G_{11} на G_{21} и G_{22} (рис. 3.12) представляет собой построение дерева ветвлений с соответствующими подмножествами, как это показано на рис. 3.13.

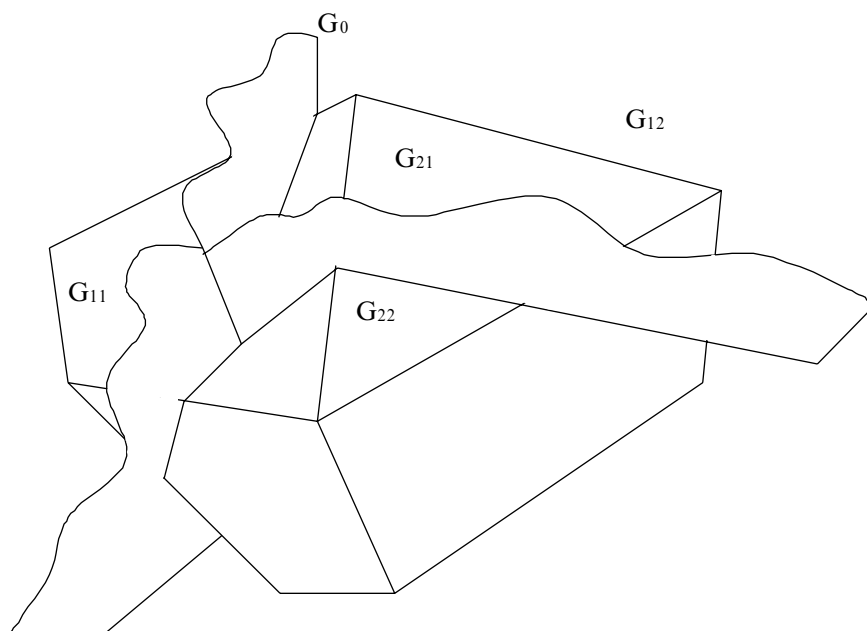


Рис. 3.12. К методу ветвей и границ

Возможность оценки образуемых подмножеств по наибольшему (наименьшему) значению для задач максимизации (минимизации) позволяет сократить перебор в силу того, что одно из подмножеств при выполнении определенных соотношений исключается и не нуждается в последующем анализе.

Понятно, что выбор оценок и границ и схемы ветвления взаимосвязаны и трудно указать общие рекомендации по использованию на практике этого метода.

Схема ветвления иллюстрируется решением обобщенной задачи одномерного раскроя с конкретными числовыми данными.

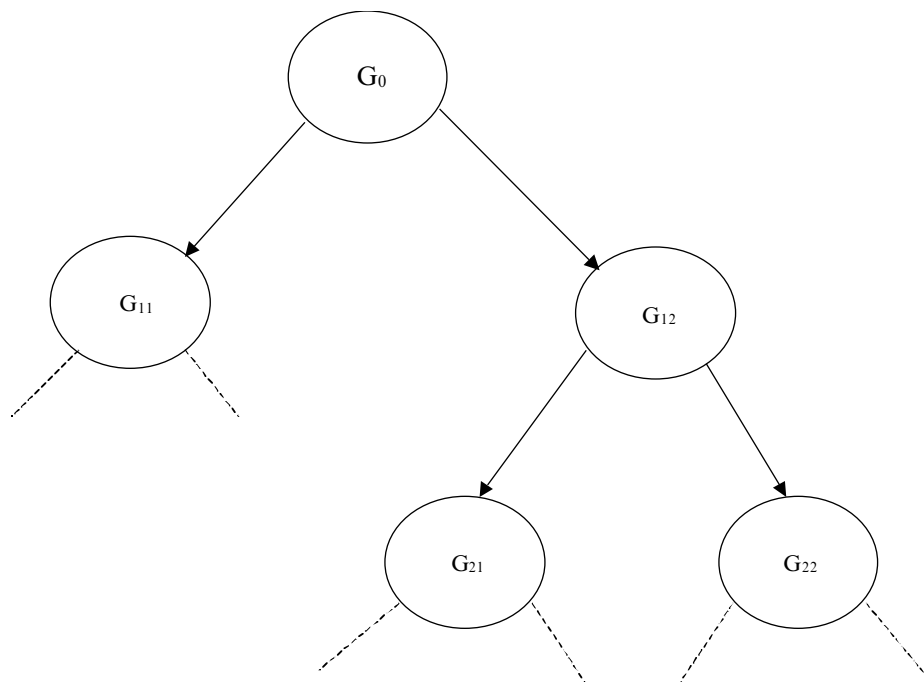


Рис. 3.13. Графическое решение для метода ветвей и границ

Пример. Имеются заготовки длиной $a_1 = 18$, $a_2 = 16$, $a_3 = 13$. Разрезая их на части, но не склеивая, можно получать детали длиной $b_1 = 12$, $b_2 = 10$, $b_3 = 8$, $b_4 = 6$, $b_5 = 5$, $b_6 = 4$. Стоимость каждой детали известна и в условных единицах численно равна их длине. Перечисленные детали представляют собой “портфель заказов”, который желательно обеспечить в первую очередь. В том случае, если это невозможно, максимизируется стоимость получаемых деталей из заданной номенклатуры. Если же портфель заказов обеспечивается, необходимо максимизировать стоимость дополнительной продукции.

Величину

$$\delta = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$$

будем называть дефицитом. Отрицательность дефицита еще не означает существование варианта раскрой, при положительном дефиците раскрой невозможен. Это свойство может быть использовано для указания перспективности варианта. Так, если заготовка a_2 раскраивается на b_3 и b_5 , то независимо от комбинаций остальных отрезков раскрой невыполним, поскольку для оставшихся отрезков дефицит положителен.

Первые этапы приводимой ниже схемы ветвления, использующей комбинаторные отношения подмножеств вариантов, показаны на рис. 3.14. На первом этапе формируются разбиения первой группы отрезков (заготовок) на вторую группу (деталей) независимо от того, какой именно отрезок первой

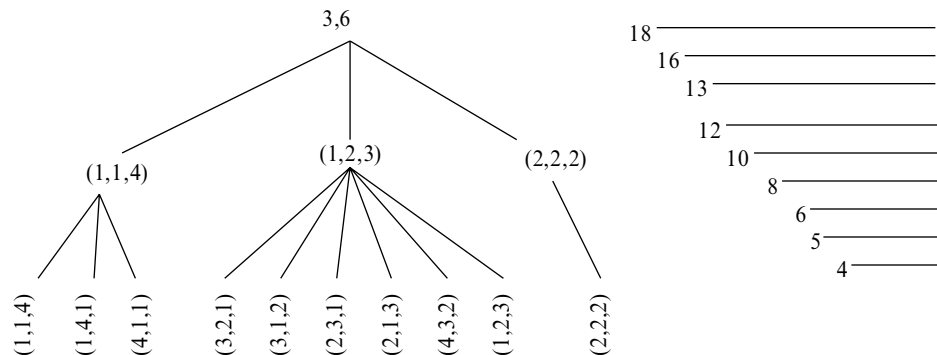


Рис. 3.14. К задаче оптимального раскроя заготовок

группы разрезается. Количество ветвей первого этапа можно вычислить лишь по рекуррентным соотношениям. Смысл подмножества $\{1, 2, 3\}$ заключается в том, что один из отрезков первой группы в результате раскроя дает один отрезок второй группы, один из оставшихся отрезков первой группы раскраивается на два отрезка второй группы из числа оставшихся и, наконец, последний отрезок первой группы делится на три части, образуя три отрезка второй группы. На втором этапе формируются все перестановки (с учетом порядка) элементов первого этапа. Скажем, вариант $\{2, 1, 3\}$ означает, что именно первый элемент первой группы а разрезается на две части и т. д.

Очередной этап ветвления образуется фиксацией отрезков второй группы при указанном на предыдущем этапе числе частей, на которое разрезается первый элемент. Другими словами, формируем сочетания по числу разделений первого элемента первого множества на число элементов второго множества. В дальнейшем фиксируем отрезки второй группы при втором элементе первой группы и, наконец, при оставшемся элементе первой группы получаем конкретные варианты раскроя, всего 447 вариантов. В подробно описанной схеме ветвления, в отличие от часто применяемых, ни на одном этапе не используется конкретный числовой материал. Рекомендуется построить для этого же примера иную схему ветвления, начиная с третьего этапа, по следующему признаку: отрезки второй группы фиксируются при элементе первой группы, разрезаемом на меньшее число частей. Затем надо сравнить схемы. Нетрудно убедиться, что на каждом этапе ветвление осуществляется по регулярной процедуре, что позволяет запоминать лишь один вариант. Сравнение дефицитов предыдущего и последующего вариантов позволяет выделить перспективный и соответствующую ему ветвь. Окончательный результат имеет вид:

$$\begin{aligned} & \{ (b_2, b_3), (b_4, b_5, b_6), (b_1) \}; \\ & \{ (b_1, b_4), (b_2, b_5), (b_3, b_6) \}; \\ & \{ (b_1, b_4), (b_2, b_6), (b_3, b_5) \}; \\ & \{ (b_1, b_5), (b_2, b_4), (b_3, b_6) \}; \\ & \{ (b_2, b_3), (b_1, b_6), (b_4, b_5) \}; \\ & \{ (b_1, b_6), (b_2, b_4), (b_3, b_5) \}; \\ & \{ (b_2, b_4), (b_1, b_6), (b_3, b_5) \}. \end{aligned}$$

Формально задачу можно было бы свести к общему виду линейной задачи целочисленного программирования, но такое сведение приведет к значительному увеличению количества переменных и другим трудностям. Матрица коэффициентов при этом приобретает квазиблочный вид с неплотным заполнением ее отличными от нуля элементами. Как правило, задачи многоиндексные с большим числом условий и сложной упорядоченностью рекомендуется решать по схеме ветвлений, используя специфику условий и ограничений.

Для задач дискретного типа этот метод получил наибольшее распространение в силу простоты и доступности и, кроме того, наиболее естественной формы описания систем условий и ограничений, являющихся, как правило, отправным пунктом построения дерева ветвления. По существу многие оригинальные алгоритмы являются модификациями метода ветвей и границ с учетом специфики условий задачи.

3.6. Другие методы оптимизации

Рассмотрим прежде всего приближенные методы решения задач целочисленного программирования, не гарантирующие сходимости за конечное число шагов, а также вероятностные методы, обеспечивающие сходимость в вероятностном смысле. Сюда же относятся эвристические методы в чистом виде и в комбинации с вероятностными.

Решение задач целочисленного программирования с помощью лингвистических моделей. Специфические свойства лингвистических методов позволяют эффективно их использовать при разработке систем управления большими системами. К таким свойствам в первую очередь отнесем простоту общения человека с машиной, так как лингвистические методы используют в качестве входных языки, более понятные неспециалисту, чем известные алгоритмические языки, а также автоматизированный (в режиме диалога) процесс формирования модели решения задачи.

Проследим основные моменты метода лингвистического моделирования на решении задачи об опасной переправе. Обозначим присутствие миссионера символом m , присутствие людоеда символом l . Тогда цепочка символов (которую в дальнейшем будем понимать как фразу языка описания состояний управляемой системы) $mmmmllll$ будет описанием начального состояния на левом берегу. При этом способе описания состояний каждая перевозка соответствует правилу вида $\alpha \rightarrow \epsilon$, где α – цепочка из символов m и (или) l , а ϵ – пустой символ. Отметим, что длина цепочки α не должна превышать грузоподъемность лодки, а количество символов m не должно быть меньше числа символов l (условие нелюдоедства в лодке). Так, для случая, когда на левом берегу находятся 4 миссионера и 3 людоеда, имеет место состояние $mmmmlll$. Пусть мы хотим перевезти двух миссионеров и одного людоеда, тогда правило выглядит как подстановка $mm \rightarrow \epsilon$, применение которой к описанию состояния дает $mmmmlll \rightarrow mml \rightarrow mml$. Введем в рассмотрение двойные перевозки, т.е. предположим, что лодка совершает рейс с левого берега на правый и обратно. Такие перевозки запишутся как правила вида $\alpha \rightarrow \beta$, где α и β – непустые цепочки, по длине не превышающие величины грузоподъемности лодки, и, кроме того, в них соблюдаются условия нелюдоедства. Если состояние на левом берегу имеет описание $mmmmlll$ и мы перевозим с левого берега на правый двух миссионеров и одного людоеда, а обратно одного миссионера и одного людоеда, то правило приобретает вид $mm \rightarrow ml$, а результат его применения – $mmmmlll \rightarrow mmlml$. Таким образом, мы сформировали знакомую модель задачи, содержащую Σ – язык описания состояний, который определяется как подмножество множества Σ всех

цепочек, составленных из символов m и l , и множество правил преобразования, каждое из которых имеет вид $\alpha \rightarrow \beta$, где α и $\beta \in \Sigma$, а Σ – алфавит, состоящий из символов m и l .

Построенная модель не дает еще ничего нового. В самом деле, она представляет собой просто удобную запись порождения графа управления – ориентированного графа, вершины которого отождествлены с состояниями на левом берегу, а ребра – с двойными перевозками. Эта модель не дает никаких сведений о том, какую перевозку из нескольких возможных для каждого состояния следует применить, чтобы общее число перевозок было минимальным. Заметим, что метод динамического программирования имеет дело практически с тем же самым описанием.

Основная **особенность методов лингвистического моделирования** заключается в том, что они вводят некие макроописания задачи, которые позволяют объединять сходные по структуре описания состояния в макросостояния. Этим достигается рассмотрение задачи с более общей точки зрения, т.е. решение ищется не на графе, а на макрографе управления, имеющем гораздо меньше вершин и ребер.

В таких задачах эффективно применяются **графовые модели** – модели, использующие концепции топологических геометрий и пространств, а также **графовые потоковые модели** в виде плоского графа, в котором отсутствуют петли, есть начальная вершина (исток) и конечная вершина (сток).

Построение макросостояний осуществляется при помощи специального аппарата порождающих грамматик, разработанного в целях моделирования естественных языков. Порождающая грамматика G представляется как совокупность терминального словаря T , нетерминального словаря N , $T \cap N = \emptyset$, начального символа $S \in N$ и множества правил вывода P , имеющих вид $\alpha \rightarrow \beta$, где в простейшем случае $\alpha \in N$, $\beta \in (N \cup T)$ – цепочка, составленная из символов терминального и нетерминального словарей.

Существуют специальные методы построения грамматик по данному подмножеству фраз языка. Эти методы в некоторых случаях допускают автоматическое построение грамматики, но в большинстве случаев оказывается возможным автоматизированное (в режиме диалога) построение.

В случае задачи об опасной переправе удастся осуществить автоматическое построение грамматик, в результате которого получаются три грамматики, разбивающие все множество состояний на три подмножества, соответствующие условиям нелюдоедства.

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

- 1. Составьте логическую схему базы знаний по теме юниты.*

2. Найдите значения максимума и минимума функции

$$f(x) = x(x - 1)^2$$

3. Методом градиентного спуска найдите минимум

$$F(x) = x_1^2/4 + x_2^2$$

4. Найдите решение следующей задачи методом наискорейшего спуска

$$F(x_1, x_2) = 4 - (x_1 - 4)^2 - (x_2 - 4)^2 \rightarrow \max$$

при $x_1 + x_2 \leq 10$
 $x_1, x_2 \geq 1$
 $x_1, x_2 \geq 0$

5. Решите следующую задачу линейного программирования

Ограничения: $x_1 + 2x_2 \leq 5$
 $3x_1 + x_2 \leq 8$
Условие неотрицательности: $x_1, x_2 \geq 0$
Целевая функция: $z = x_1 + x_2 \rightarrow \max$

6. Найдите в области $D = \{ |x_1| < 1, |x_2| < 1 \}$

$$\min_D F(x_1, x_2) = x_1^4 + x_2^4 - x_1^2 - x_2^2$$

через решение системы уравнений

$$\begin{aligned} 2x_1^3 - x_1 &= 0 \\ 2x_2^3 - x_2 &= 0 \end{aligned}$$

ТРЕНИНГ УМЕНИЙ

1. Пример выполнения упражнения тренинга на умение № 1

Задание

Для функции $F(x_1, x_2)$ найти условный экстремум, если

$$F(x_1, x_2) = x_1 x_2$$

при ограничениях

$$x_1^2 + x_2^2 = 2$$

Решение

№ п/п	Алгоритм	Конкретное действие, соответствующее предложенному алгоритму
1.	Построить функцию Лагранжа	$L(x_1, x_2, \lambda) = x_1 x_2 + \lambda(2 - x_1^2 - x_2^2)$
2	Найти частные производные первого порядка для $L(x_1, x_2, \lambda)$	$\partial L / \partial x_1 = x_2 - 2\lambda x_1; \quad \partial L / \partial x_2 = x_1 - 2\lambda x_2;$ $\partial L / \partial \lambda = 2 - x_1^2 - x_2^2$
3.	Составить систему уравнений для нахождения стационарных точек $L(x_1, x_2, \lambda)$	$\begin{cases} x_2 - 2\lambda x_1 = 0 \\ x_1 - 2\lambda x_2 = 0 \\ 2 - x_1^2 - x_2^2 = 0 \end{cases}$
4.	Решить систему уравнений	$x_1^0 = 1; x_2^0 = 1; \lambda^0 = 1/2;$ $x_1^1 = -1; x_2^1 = -1; \lambda^1 = 1/2;$ $x_1^2 = 1; x_2^2 = -1; \lambda^2 = -1/2;$ $x_1^3 = -1; x_2^3 = 1; \lambda^3 = 1/2$
5.	Найти экстремумы функции	$F(1,1) = 1; F(-1, -1) = 1; F(-1,1) = -1; F(1, -1) = -1.$ Откуда: $F_{\max} = 1$ при $(x_1 = 1, x_2 = 1)$ и $(x_1 = -1, x_2 = -1);$ $F_{\min} = -1$ при $(x_1 = -1, x_2 = 1)$ и $(x_1 = 1, x_2 = -1).$

Решите самостоятельно:

Задание 1.1

Для функции $F(x_1, x_2)$ найти условный экстремум, если

$$F(x_1, x_2) = x_1 + x_2$$

при ограничениях

$$1/x_1 + 1/x_2 = 2 \text{ и } x_1, x_2 \geq 0.$$

Задание 1.2

Для функции $F(x_1, x_2)$ найти условный экстремум, если

$$F(x_1, x_2) = x_1^2 + x_2^2$$

при ограничениях

$$x_1 + x_2 = 2 \quad \text{и} \quad x_1, x_2 \geq 0.$$

2. Пример выполнения упражнения тренинга на умение № 2

Задание

Минимизировать функцию

$$F = 1/4 X_1 + 1/3 X_2$$

при ограничениях

$$\begin{aligned} 5 X_1 + X_2 - 5 &\geq 0 \\ 2 X_1 + 5 X_2 - 10 &\geq 0 \\ X_1, X_2 &\geq 0. \end{aligned}$$

Решение

№ п/п	Алгоритм	Конкретное действие, соответствующее предложенному алгоритму
1.	Ввести вспомогательные переменные	$x_1 = X_1, \quad x_2 = 5X_1 + X_2 - 5,$ $x_3 = 2 X_1 + 5X_2 - 10, \quad x_4 = X_2$
2.	Преобразовать задачу к стандартной форме	$F = 1/4 x_1 + 1/3 x_4 = \min$ $5x_1 - x_2 + x_4 = 5,$ $2x_1 - x_3 + 5x_4 = 10,$ $x_1, x_2, x_3, x_4, \geq 0$
3.	Получить каноническую форму задачи, принимая в качестве базисных переменных x_1, x_2	$x_1 - 1/2x_3 + 5/2x_4 = 5 > 0,$ $x_2 - 5/2 x_3 + 23/2x_4 = 20 > 0,$ $24F = 3x_3 - 7 x_4 + 30$
4.	Провести анализ поведения целевой функции в зависимости от изменений x_4	Коэффициент при x_4 отрицателен ($24F = 3 x_3 - 7x_4 + 30$), следовательно, F убывает при возрастании x_4 . Необходимо заменить на x_4 одну из базисных переменных
5.	Провести анализ того, какую базисную переменную необходимо заменить	При возрастании C первой обращается в нуль базисная переменная x_2 . Следовательно, x_4 заменит x_2 .
6.	Найти значения коэффициентов в новой канонической форме после замены базиса	$x_2 = x_3 = 0,$ $x_4 = 20/(23/2) = 40/23,$ $x_1 = 15/23.$
7.	Получить новую каноническую форму после замены базиса на x_1 и x_4	$x_1 - 5/23 x_2 + 1/23 x_3 = 15/23 > 0,$ $x_4 + 2/23 x_3 - 5/23 x_2 = 40/23 > 0,$ $12 \cdot 23 F = 7 x_2 + 17 x_3 + 205$
8.	Провести анализ поведения целевой функции в зависимости от изменений свободных переменных x_3 и x_2	В целевой функции коэффициенты при свободных переменных x_3 и x_2 положительны, значит, можно получить единственное оптимальное решение.

№ п/п	Алгоритм	Конкретное действие, соответствующее предложенному алгоритму
9.	Найти оптимальное решение	Приравниваем свободные переменные нулю и получаем: $F = 205/12 \cdot 23 = 205/276$ единственное оптимальное решение

Решите самостоятельно:

Задание 2.1

Минимизировать функцию

$$F = 1/8 X_1 + 1/2 X_2$$

при ограничениях

$$\begin{aligned} 3 X_1 + X_2 - 5 &\geq 0 \\ 2 X_1 + 3 X_2 - 10 &\geq 0 \\ X_1, X_2 &\geq 0. \end{aligned}$$

Задание 2.2

Минимизировать функцию

$$F = 4 X_1 + 3 X_2$$

при ограничениях

$$\begin{aligned} 4 X_1 + X_2 - 3 &\geq 0 \\ X_1 + 5X_2 - 15 &\geq 0 \\ X_1, X_2 &\geq 0. \end{aligned}$$

3. Пример выполнения упражнения тренинга на умение № 3

Задание

Исследовать функцию

$$F(x_1, x_2) = x_1^4 + x_2^4 - 2x_1x_2 - x_1^2 - x_2^2$$

на безусловный экстремум.

Решение

№ п/п	Алгоритм	Конкретное действие, соответствующее предложенному алгоритму
1.	Найти частные производные первого порядка	$\partial F / \partial x_1 = 4x_1^3 - 2x_2 - 2x_1;$ $\partial F / \partial x_2 = 4x_2^3 - 2x_2 - 2x_1$
2.	Построить систему уравнений	$\begin{cases} 4x_1^3 - 2x_2 - 2x_1 = 0 \\ 4x_2^3 - 2x_2 - 2x_1 = 0 \end{cases}$
3.	Найти стационарные точки	$(x_1 = 1, x_2 = 1); (x_1 = 0, x_2 = 0);$ $(x_1 = -1, x_2 = -1)$
4.	Найти частные производные второго порядка	$\partial^2 F / \partial x_1^2 = 12x_1^2 - 2;$ $\partial^2 F / \partial x_2^2 = 12x_2^2 - 2;$ $\partial^2 F / \partial x_1 \partial x_2 = -2$
5.	Найти значения частных производных второго порядка в первой стационарной точке $(x_1 = 0, x_2 = 0)$	$A_1 = -2; A_2 = -2; A_{12} = -2$
6.	Определить наличие экстремума в точке $(x_1 = 0, x_2 = 0)$	$A_1 A_2 - A_{12}^2 = 0$ В точке $(x_1 = 0, x_2 = 0)$ нужны дополнительные исследования
7.	Найти значения частных производных второго порядка во второй стационарной точке $(x_1 = 1, x_2 = 1)$	$A_1 = 10; A_2 = -10; A_{12} = -2$
8.	Определить наличие экстремума в точке $(x_1 = 1, x_2 = 1)$	$A_1 A_2 - A_{12}^2 = 96 > 0, A_1 = 10 > 0.$ В точке $(x_1 = 1, x_2 = 1)$ имеется минимум
9.	Найти минимум функции в точке $(x_1 = 1, x_2 = 1)$	$F_{\min} = -2$ в точке $(x_1 = 1, x_2 = 1)$
10.	Найти значения частных производных второго порядка во 2-й стационарной точке $(x_1 = -1, x_2 = -1)$	$A_1 = 10; A_2 = 10; A_{12} = -2$

№ п/п	Алгоритм	Конкретное действие, соответствующее предложенному алгоритму
11.	Определить наличие экстремума в точке $(x_1 = -1, x_2 = -1)$	$A_1 A_2 - A_{12}^2 = 96 > 0, A_1 = 10 > 0$. В точке $(x_1 = -1, x_2 = -1)$ имеется минимум
12.	Найти минимум функции в точке $(x_1 = -1, x_2 = -1)$	$F_{\min} = -2$ в точке $(x_1 = -1, x_2 = -1)$

Решите самостоятельно:

Задание 3.1

Исследовать функцию

$$F(x_1, x_2) = x_1^3 + x_2^3 - 3x_1x_2 - x_1^2 - x_2^2$$

на безусловный экстремум.

Задание 3.2

Исследовать функцию

$$F(x_1, x_2) = x_1^4 - 2x_1x_2 + x_1^2 - x_2 + 2$$

на безусловный экстремум.

МЕТОДЫ ОПТИМИЗАЦИИ

ЮНИТА 2

ПРЯМЫЕ МЕТОДЫ ОТЫСКАНИЯ ЭКСТРЕМУМА И ОСНОВЫ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Редактор Н.М. Пилипенко
Оператор компьютерной верстки Д.В. Федотов

Изд. лиц. ЛР № 071765 от 07.12.1998
НОУ “Современный Гуманитарный Институт”
Тираж

Сдано в печать
Заказ
