

## 1.1. Область допустимых значений вектора управляемых параметров $\mathbf{X}$

Проектирование технических объектов всегда включает в себя элементы оптимизации – стремление получить наилучший вариант среди возможных вариантов. Это стремление реализуется перебором вариантов структуры объекта (структурный синтез) и варьированием значений параметров объекта при заданной структуре (параметрическая оптимизация или просто оптимизация).

Внутренние параметры объекта проектирования обозначим  $n$ -мерным вектором  $\mathbf{X}=(x_1, x_2, \dots, x_n)$ , выходные параметры –  $m$ -мерным вектором  $\mathbf{Y}=(y_1, y_2, \dots, y_m)$ , а внешние параметры (параметры окружающей среды) –  $l$ -мерным вектором  $\mathbf{Q}=(q_1, q_2, \dots, q_l)$ .

Таким образом, в самом общем виде модель объекта проектирования можно задать в следующем виде:

$$\mathbf{Y}=\mathbf{F}(\mathbf{X}, \mathbf{Q}),$$

где  $\mathbf{F}$  – некоторая вектор-функция, которая может задаваться различными способами: с помощью формул, графиков, таблиц, алгоритмов вычисления и т.д.

Если внешние параметры  $\mathbf{Q}$  известны и фиксированы, то

$$\mathbf{Y}=\mathbf{F}(\mathbf{X}),$$

Такая модель называется детерминированной моделью объекта проектирования (в том смысле, что значения  $\mathbf{Y}$  однозначно определяются значениями  $\mathbf{X}$ ).

Модель, в которой внешние параметры  $\mathbf{Q}$  являются случайными величинами, называется стохастической моделью объекта проектирования или вероятностной моделью объекта проектирования.

Варьируемые при оптимизации параметры называются управляемыми параметрами (варьируемыми параметрами) или переменными. Будем их также обозначать вектором  $\mathbf{X}$  и называть вектором управляемых параметров (вектором варьируемых параметров) или переменных. Важно понимать, что в этот вектор не обязательно включаются все внутренние параметры объекта проектирования.

Требования к проектируемому объекту обычно можно сформулировать в виде системы неравенств:

$$x_i^- \leq x_i \leq x_i^+, i \in [1, n] \quad (1)$$

$$y_j^- \leq y_j \leq y_j^+, j \in [1, m] \quad (2)$$

Здесь  $x_i^-$ ,  $x_i^+$  – значения  $i$ -й управляемой переменной, определяющие область ее возможных значений,  $y_j^-$ ,  $y_j^+$  – предельные допустимые значения выходного параметра  $Y_j$ .

Поскольку существует функциональная связь, ограничения (2) эквивалентны системе

$$\{g_j(\mathbf{X}), j \in [1, m]\} = g(\mathbf{X}) \geq 0, \quad (3)$$

где  $g(\mathbf{X})$  –  $m$ -мерная вектор-функция. Заметим, что в виде (3) можно записать и ограничения типа равенств  $g(\mathbf{X})=0$  путем замены их парой неравенств  $g(\mathbf{X}) \geq 0, -g(\mathbf{X}) \geq 0$ .

Одной из особенностей задач проектирования является то, что в систему ограничений (3) могут входить функции, зависящие от одной из компонент вектора  $\mathbf{Q}$  – некоторого

параметра  $q$ , заданного на интервале  $[q_{\min}, q_{\max}]$ . Таким параметром может быть время, частота, температура и т.п. В этом случае

$$g(\mathbf{X}, q) \geq 0, q \in [q_{\min}, q_{\max}] \quad (4)$$

Для перехода от (4) к (3) можно использовать: 1) сеточный метод, 2) принцип гарантированного результата.

Идея сеточного метода основана на дискретизации интервала  $[q_{\min}, q_{\max}]$  и рассмотрении функции  $g(\mathbf{X}, q)$  на дискретной совокупности точек  $q_1, q_2, \dots, q_M$  (см. рис. 1).



Рис. 1. Дискретизация интервала  $[q_{\min}, q_{\max}]$

При этом выполнение ограничений (4) сводится к требованию выполнения системы из  $M$  неравенств:

$$g(\mathbf{X}, q_i) \geq 0, i = 1, 2, \dots, M. \quad (5)$$

Недостатки подхода: трудно обоснованно выбрать число  $M$ ; вместо одного неравенства (4) приходится рассматривать систему из  $M$  неравенств (5).

Принцип гарантированного результата состоит в том, что ограничения (4) проверяются для наиболее неблагоприятного значения параметра  $q^* \in [q_{\min}, q_{\max}]$ :

$$g(\mathbf{X}, q^*) = \min_{q \in [q_{\min}, q_{\max}]} g(\mathbf{X}, q)$$

Недостатки подхода: в общем случае  $q^* = q^*(\mathbf{X})$  и отыскание  $q^*$  является самостоятельной проблемой.

Далее будем полагать, что ограничения на параметр  $q$ , заданные на интервале  $[q_{\min}, q_{\max}]$ , с помощью сеточного метода или принципа гарантированного результата сведены к системе неравенств (3). Тогда условие (3) определяет множество допустимых значений вектора  $\mathbf{X}$

$$D_g = \{ \mathbf{X} | g(\mathbf{X}) \geq 0 \}.$$

Ограничения (1) определяют следующее множество допустимых значений вектора  $\mathbf{X}$ :

$$D_x = \{ \mathbf{x} | x_i^- \leq x_i \leq x_i^+, i \in [1, n] \}.$$

Множество, полученное пересечением множеств  $D_g$  и  $D_x$ , будем называть множеством допустимых значений вектора управляемых параметров  $\mathbf{X}$ , т.е.

$$D = D_g \cap D_x.$$

Любой вектор управляемых переменных  $\mathbf{X} \in D$  называется допустимым вектором управляемых параметров.

Если вектор  $\mathbf{X} \in D$ , то будем такой вектор называть также точкой (множества  $D$ ).

Допущение 1: если не оговорено противное, множество допустимых значений вектора варьируемых параметров  $D$  является ограниченным и замкнутым (компактным).

Чаще всего ограничения (1) – (2) записывают единообразно - в виде ограничений вида  $g(\mathbf{X}) \geq 0$ . Будем называть такие ограничения ограничениями типа неравенств, а функции  $g(\mathbf{X})$  - ограничивающими функциями.

Если на вектор варьируемых параметров наложены только ограничения типа неравенств, то множество допустимых значений вектора варьируемых параметров определяется следующим образом:

$$D = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0, i \in [1, m]\}. \quad (6)$$

На вектор варьируемых параметров могут быть наложены, как отмечалось, также ограничения типа равенств. Эти ограничения можно либо сведены к ограничениям типа неравенств, либо выделить в отдельную группу ограничений. В последнем случае множество допустимых значений вектора варьируемых параметров определяется следующим образом:

$$D = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0, h_j(\mathbf{X}) = 0, i \in [1, m], j \in [1, l]\}.$$

Функции  $h(\mathbf{X})$ , с помощью которых задаются ограничения типа равенств, также будем называть ограничивающими функциями.

## 1.2. Выпуклое множество допустимых значений вектора варьируемых параметров

Широкий класс методов оптимизации ориентирован на решение задач оптимизации, у которых множество допустимых значений вектора варьируемых параметров является выпуклым множеством. Множество допустимых значений вектора варьируемых параметров  $D$  называется выпуклым множеством допустимых значений вектора варьируемых параметров, если для любых точек  $\mathbf{X}_1, \mathbf{X}_2 \in D$  и для любого  $\lambda \in [0,1]$  выполняется соотношение  $\lambda\mathbf{X}_1 + (1-\lambda)\mathbf{X}_2 \in D$ . Например, на рис. 1, который иллюстрирует двумерный случай ( $n=2$ ), все точки отрезка  $[\mathbf{X}_1, \mathbf{X}_2]$  принадлежат множеству  $D$  и поэтому это множество выпукло. В противном случае множество допустимых значений  $D$  называется не выпуклым множеством допустимых значений вектора варьируемых параметров. Например, на рис.2 часть  $[A, B]$  отрезка  $[\mathbf{X}_1, \mathbf{X}_2]$  не принадлежат множеству  $D$ , которое поэтому не является выпуклым.

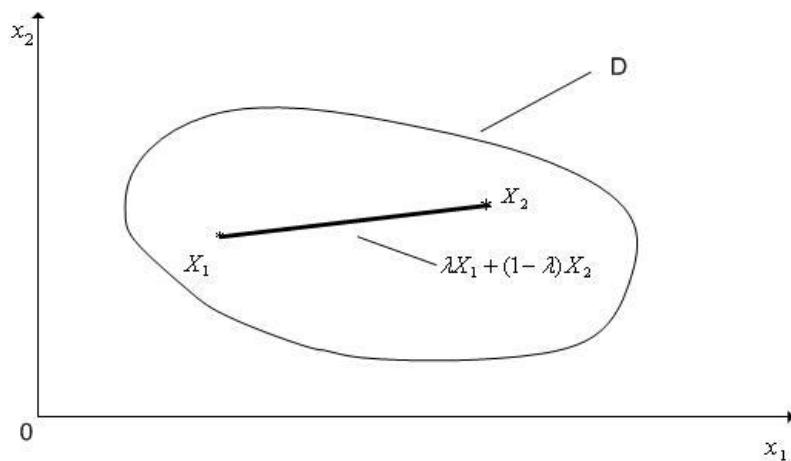


Рис. 1. К определению выпуклого множества ( $n=2$ ).

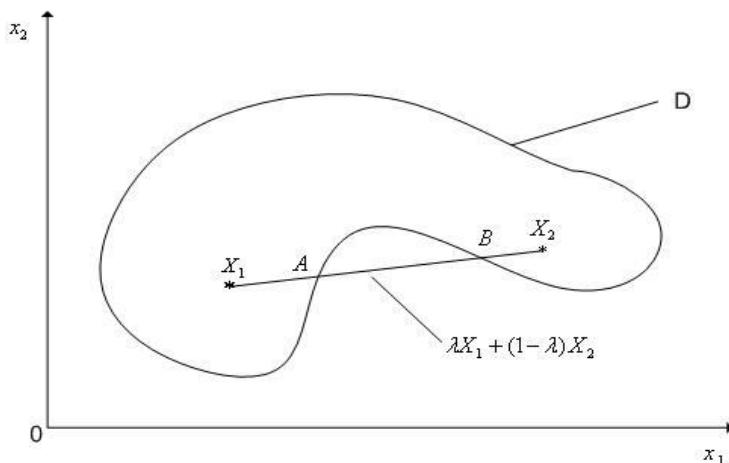


Рис. 2. Пример не выпуклого множества ( $n=2$ ).

Множество  $L(\mathbf{X}_1, \mathbf{X}_2) = \{\mathbf{X} | \mathbf{X} = \lambda\mathbf{X}_1 + (1-\lambda)\mathbf{X}_2 \in D, \lambda \in [0,1]\}$  будем называть отрезком с концами  $\mathbf{X}_1, \mathbf{X}_2$  и обозначать  $[\mathbf{X}_1, \mathbf{X}_2]$ .

На основе введенного понятия, можно дать другое определение выпуклого множества  $D$ . Множество допустимых значений вектора варьируемых параметров  $D$  называется

выпуклым множеством, если оно наряду с любыми точками  $\mathbf{X}_1, \mathbf{X}_2 \in D$  содержит в себе также отрезок  $[\mathbf{X}_1, \mathbf{X}_2]$ .

Из определения следует, что все евклидово пространство  $R^n$  является выпуклым множеством.

Легко показать, что любой отрезок  $[\mathbf{X}_1, \mathbf{X}_2]$  в пространстве  $R^n$  является выпуклым множеством.

Из курса аналитической геометрии известно, что в пространстве  $R^n$  уравнение

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b \quad (1)$$

определяет гиперплоскость. Здесь  $a_1, a_2, \dots, a_n, b$  некоторые константы.

**Теорема 1.** Гиперплоскость  $a_1x_1 + a_2x_2 + \dots + a_nx_n = b$  является выпуклым множеством в пространстве  $R^n$ .

Доказательство. Уравнение (1) перепишем в виде скалярного произведения

$$(\mathbf{X}, \mathbf{A}) = b, \quad (2)$$

где  $n$ -мерный вектор  $\mathbf{A} = (a_1, a_2, \dots, a_n)$ . Возьмем произвольные точки  $\mathbf{X}_1, \mathbf{X}_2$ , удовлетворяющие уравнению (2):  $(\mathbf{X}_1, \mathbf{A}) = b$ ,  $(\mathbf{X}_2, \mathbf{A}) = b$ . Рассмотрим некоторую точку  $\mathbf{X} = \lambda\mathbf{X}_1 + (1-\lambda)\mathbf{X}_2$ ,  $\lambda \in [0, 1]$  принадлежащую отрезку  $[\mathbf{X}_1, \mathbf{X}_2]$ . По свойствам скалярного произведения имеем

$$(\mathbf{X}, \mathbf{A}) = (\lambda\mathbf{X}_1 + (1-\lambda)\mathbf{X}_2, \mathbf{A}) = \lambda(\mathbf{X}_1, \mathbf{A}) + (1-\lambda)(\mathbf{X}_2, \mathbf{A}) = \lambda b + (1-\lambda)b = b.$$

Это означает, что точка  $\mathbf{X}$  принадлежит гиперплоскости (2) •

Очевидно, что в пространстве  $R^2$  уравнения (1), (2) определяют прямую, а в пространстве  $R^3$  - плоскость.

Опять же из курса аналитической геометрии известно, что в пространстве  $R^n$  неравенство

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = (\mathbf{X}, \mathbf{A}) \leq b \quad (3)$$

определяет полупространство. Здесь  $a_1, a_2, \dots, a_n, b$  также некоторые константы.

**Теорема 2.** Полупространство  $a_1x_1 + a_2x_2 + \dots + a_nx_n = (\mathbf{X}, \mathbf{A}) \leq b$  является выпуклым множеством в пространстве  $R^n$  •

**Теорема 3.** Если функции  $g_i(\mathbf{X}) \geq 0$ ,  $i \in [1, m]$  выпуклы в  $R^n$ , то множество  $D = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0\} = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0, i \in [1, m]\}$  является выпуклым •

Определение выпуклой функции дано ниже.

Для доказательства выпуклости множеств часто бывает полезна следующая теорема.

**Теорема 4.** Пересечение любого конечного числа выпуклых множеств является выпуклым множеством.

Доказательство. Пусть  $\bar{D} = \{D_1, D_2, \dots, D_p\}$  – совокупность  $p$  выпуклых множеств пространства  $R^n$ . Рассмотрим множество

$$D = \bigcap_{i \in [1, p]} D_i,$$

являющееся пересечением множеств совокупности  $\bar{D}$ . Пусть точки  $\mathbf{X}_1, \mathbf{X}_2$  принадлежат множеству  $\bar{D}$ . Тогда они также принадлежат каждому из множеств  $D_i \in \bar{D}$ . Так как множество выпукло, то точка  $\mathbf{X} = \lambda\mathbf{X}_1 + (1-\lambda)\mathbf{X}_2$ ,  $\lambda \in [0, 1]$  принадлежит этому множеству.

Поскольку множество  $D_i$  выбрано произвольно, то точка  $\mathbf{X} = \lambda \mathbf{X}_1 + (1-\lambda) \mathbf{X}_2$ ,  $\lambda \in [0,1]$  принадлежит множеству  $D_{\bullet}$

Из теоремы 4 следует, что выпуклыми множествами являются, например, следующие множества:

гиперпараллелепипед  $a_1 \leq x_1 \leq b_1; a_2 \leq x_2 \leq b_2; \dots; a_n \leq x_n \leq b_n;$

симплекс  $x_1 \geq 0; x_2 \geq 0; \dots; x_n \geq 0; x_1 + x_2 + \dots + x_n \leq h;$

выпуклый многогранник в пространстве  $R^n$ , определяемый системой неравенств (каждое из которых задет полупространство)

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = (\mathbf{X}, \mathbf{A}_1) \leq b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = (\mathbf{X}, \mathbf{A}_2) \leq b_2,$$

... ...

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = (\mathbf{X}, \mathbf{A}_m) \leq b_m.$$

**Утверждение 5.** Если функции  $g_i(\mathbf{X}) \geq 0, i \in [1, m]$  выпуклы в  $R^n$ , а функции  $h_j(\mathbf{X}) \geq 0, j \in [1, l]$  линейны, то множество

$$D = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0, h_j(\mathbf{X}) = 0, i \in [1, m], j \in [1, l]\} \quad (4)$$

является выпуклым.

Справедливость утверждения следует из того факта, что по теореме 3 ограничения типа неравенств и по теореме 1 ограничения типа равенств определяют в соотношении (4) выпуклые множества  $\bullet$

Покажем в заключение, что гипершар

$$(x_1 - x_1^0)^2 + (x_2 - x_2^0)^2 + \dots + (x_n - x_n^0)^2 \leq r^2, \quad (5)$$

является выпуклым множеством ( $r > 0$ ).

Известно, что евклидова норма  $n$ -мерного вектора  $\mathbf{X}$  есть

$$\|\mathbf{X}\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}.$$

Поэтому неравенство (5) можно переписать в виде  $\|\mathbf{X} - \mathbf{X}^0\| \leq r$ , где  $n$ -мерный вектор  $\mathbf{X}^0 = (x_1^0, x_2^0, \dots, x_n^0)$ . Пусть произвольные точки  $\mathbf{X}_1, \mathbf{X}_2$  принадлежат гипершару (5) и  $\lambda \in [0,1]$ . Тогда по свойству нормы имеем

$$\|\mathbf{X} - \mathbf{X}^0\| = \|(\lambda \mathbf{X}_1 + (1-\lambda) \mathbf{X}_2) - \mathbf{X}^0\| = \|(\lambda(\mathbf{X}_1 - \mathbf{X}^0) + (1-\lambda)(\mathbf{X}_2 - \mathbf{X}^0))\| \leq \lambda \|\mathbf{X}_1 - \mathbf{X}^0\| + (1-\lambda) \|\mathbf{X}_2 - \mathbf{X}^0\| \leq \lambda r + (1-\lambda)r = r$$

### 1.3. Постановка детерминированной задачи оптимизации

Под решением задачи оптимального проектирования понимается процесс выбора управляемых переменных  $\mathbf{X} \in D$ , обеспечивающих оптимальное значение некоторой функции  $\Phi(\mathbf{X})$ . Эта величина, показывающая относительное предпочтение одних значений компонент вектора  $\mathbf{X}$  по отношению к другим значениям этих компонент, называется критерием оптимальности (функцией цели, критерием эффективности, функцией полезности и т.д.).

В зависимости от цели проектирования необходимо либо максимизировать, либо минимизировать критерий оптимальности. Будем полагать, что требуется минимизировать критерий оптимальности.

Детерминированная задача оптимального проектирования (детерминированная задача оптимизации) формулируется следующим образом:

$$\min_{\mathbf{X} \in D} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где  $\mathbf{X}^*$  - оптимальное значение вектора варьируемых параметров (переменных),  $\Phi(\mathbf{X}^*) = \Phi^*$  - наименьшее, т.е. оптимальное значение критерия оптимальности  $\Phi(\mathbf{X})$ .

Задача оптимизации, в которой критерий оптимальности  $\Phi(\mathbf{X})$  и/или ограничивающие функции  $\xi(\mathbf{X})$  зависят от случайного вектора внешних параметров  $\mathbf{Q}$ , называется стохастической задачей оптимального проектирования (стохастической задачей оптимизации).

Заместим, что задача максимизации критерия оптимальности  $\Phi(\mathbf{X})$  сводится к задаче минимизации критерия  $(-\Phi(\mathbf{X}))$ :

$$\max_{\mathbf{X} \in D} \Phi(\mathbf{X}) = \min_{\mathbf{X} \in D} (-\Phi(\mathbf{X})).$$

Имея в виду специфику задач оптимального проектирования, сделаем следующее допущение.

Допущение 2. Если не оговорено противное, функция  $\Phi(\mathbf{X})$  в своей области допустимых значений  $D$  определена, непрерывна и, за исключением, быть может отдельных точек, имеет в этой области конечные частные производные.

Из курса математического анализа известна следующая теорема.

**Теорема 1** (теорема Вейерштрасса). Если функция  $\Phi(\mathbf{X})$  определена и непрерывна в ограниченной замкнутой области  $D$ , то она достигает в этой области своего наименьшего и наибольшего значений ●

Таким образом, в допущениях 1,2 детерминированная задача оптимального проектирования (1) имеет решение.

Пример 1

Тело массой  $m$  с двигателем находится в момент времени  $t=0$  в состоянии покоя в точке с координатой  $x_1=x_2=0$ . Тело может перемещаться вдоль оси  $Ox$  под действием силы тяги двигателя. Максимальная сила тяги двигателя равна  $a$ . При движении тела на него действует сила сопротивления среды  $F_c = -kV$ , где  $V$  - скорость тела. Расход горючего при работе двигателя пропорционален квадрату силы тяги, т.е. равен  $\gamma u^2$ , где  $u$ -сила тяги двигателя.

Необходимо на интервале времени  $[0, T]$  так управлять работой двигателя, чтобы при минимальном расходе топлива закон движения тела  $\mathbf{x}(t)$  как можно меньше отличался от требуемого закона движения  $\Phi(t)$ . Т.е. требуется найти оптимальную функцию  $u(t)$ .

Пусть  $\mathbf{X}(t) = (x_1(t), x_2(t))$ , где

$$x_2(t) = \dot{x}_1(t) = V(t).$$

Тогда по второму закону Ньютона имеем

$$m\ddot{x}_2(t) = u(t) + F_c = u(t) - kx_2(t).$$

Здесь  $\ddot{x}_2(t)$  – ускорение тела.

Таким образом, получаем следующую задачу Коши для системы обыкновенных дифференциальных уравнений

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \quad x_1(0) = 0, \\ \dot{x}_2(t) &= -\frac{k}{m}x_2(t) + \frac{1}{m}u(t), \quad x_2(0) = 0 \end{aligned} \tag{2}$$

где  $t \in [0, T]$ ,  $|u(t)| \leq a$ .

Требование близости траектории тела к заданной траектории формализуем с помощью критерия оптимальности

$$\Phi_1(u) = \int_0^T (x_1(t) - \varphi(t))^2 dt, \tag{3}$$

а требование минимизации расхода топлива – с помощью критерия

$$\Phi_2(u) = \int_0^T \gamma u^2(t) dt. \tag{4}$$

В качестве критерия оптимальности задачи рассмотрим взвешенную сумму указанных «частных» критериев

$$\Phi(u) = \Phi_1(u) + \lambda \Phi_2(u), \quad \lambda \in [0, 1]. \tag{5}$$

Будем искать оптимальное управление  $u^*(t)$  на классе кусочно-постоянных функций (см. рис. 1).

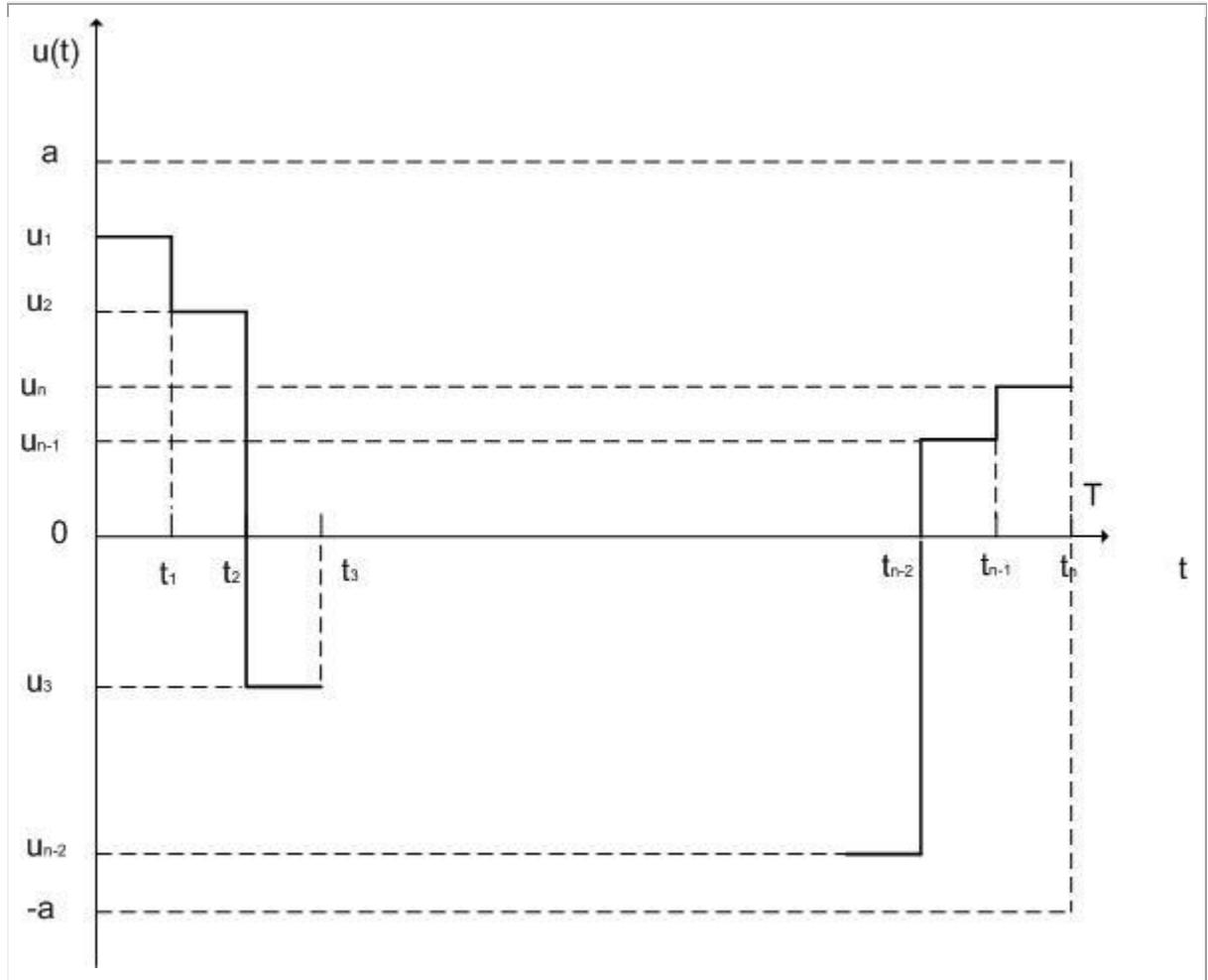


Рис. 1. Структура допустимых управлений  $u(t)$ .

В качестве вектора варьируемых параметров будем рассматривать  $n$ -мерный вектор

$$\mathbf{U} = (u_1, u_2, \dots, u_n), |u_i| \leq a, i \in [1, n].$$

Тогда задача поиска оптимального управления  $\mathbf{u}^*(t)$  сводится к  $n$ -мерной детерминированной задаче оптимизации

$$\min_{\mathbf{U} \in D} \Phi(\mathbf{U}) = \Phi(\mathbf{U}^*), D = \{u_i = u(t_i), |u_i| \leq a, i \in [1, n]\},$$

где значения функции  $\Phi(\mathbf{U})$  вычисляются по следующей схеме:

задаем значения  $u_1, u_2, \dots, u_n$  управления  $u(t)$  в точках  $t_1, t_2, \dots, t_n$ ;

с кусочно-постоянным управлением, показанным на рис. 1, интегрируем систему обыкновенных дифференциальных уравнений (2) – получаем вектор  $\mathbf{X}(t), t \in [0, T]$ ;

по формулам (3), (4) вычисляем значения частных критериев оптимальности  $\Phi_1(u)$ ,  $\Phi_2(u)$ , а затем по формуле (5) – значение критерия оптимальности  $\Phi(u)$ .

**Замечание.** Рассмотренная задача представляет собой задачу оптимального управления динамической системой. Позже будут рассмотрены более корректные методы численного решения этой задачи.

#### 1.4. Классификация критериев оптимальности

Если случайный вектор внешних параметров  $\mathbf{Q}$  не входит в критерий оптимальности  $\Phi(\mathbf{X})$ , то этот критерий называется детерминированным критерием оптимальности. Если критерий оптимальности имеет вид  $\Phi(\mathbf{X}, \mathbf{Q})$ , где  $\mathbf{Q}$  – случайный вектор внешних параметров, то этот критерий называется стохастическим критерием оптимальности. Прежде, чем говорить о классификации детерминированных критериев оптимальности, дадим определения некоторых свойств функций.

Вектор  $\mathbf{X}^*$  называется точкой локального минимума функции  $\Phi(\mathbf{X})$ , если для всех точек  $\mathbf{X}$ , принадлежащих некоторой малой окрестности  $d(\mathbf{X})$  точки  $\mathbf{X}^*$  имеем

$$\Phi(\mathbf{X}^*) \leq \Phi(\mathbf{X}), \mathbf{X} \in d(\mathbf{X}^*) \subset D.$$

Значение функции  $\Phi(\mathbf{X})$  в точке локального минимума называется локальным минимумом функции  $\Phi(\mathbf{X})$ . Таким образом, если точка  $\mathbf{X}^*$  является точкой локального минимума функции  $\Phi(\mathbf{X})$ , то величина  $\Phi(\mathbf{X}^*)$  есть локальный минимум этой функции.

Точка  $\mathbf{X}^*$  называется точкой глобального минимума функции  $\Phi(\mathbf{X})$ , если

$$\Phi(\mathbf{X}^*) \leq \Phi(\mathbf{X}), \mathbf{X} \in D$$

таким образом, точка наименьшего из всех локальных минимумов называется точкой глобального минимума функции  $\Phi(\mathbf{X})$ . Соответствующее значение функции  $\Phi(\mathbf{X})$  называется глобальным минимумом этой функции. Например, на рис. 1, который иллюстрирует одномерный случай ( $n=1$ ),  $x_1^*, x_2^*, x_3^*$  -точки локального минимума функции  $\Phi(x)$ , а величины  $\Phi_1^*, \Phi_2^*, \Phi_3^*$ - соответствующие локальные минимумы этой функции,  $x_3^*$  - точка глобального минимума функции  $\Phi(x)$ , а  $\Phi_3^*$ - глобальный минимум этой функции.

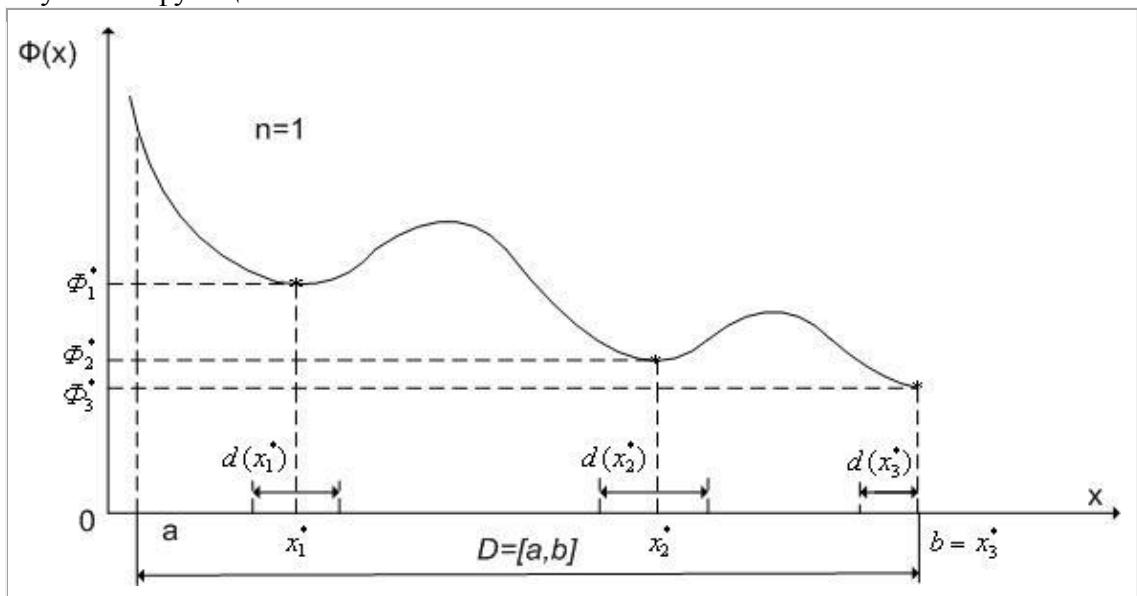


Рис. 1. К определению локального и глобального минимумов функции

Критерий оптимальности  $\Phi(x)$ , где  $x \in [a, b]$  скаляр, называется унимодальным критерием оптимальности, если в области определения  $[a, b]$  функции  $\Phi(x)$  существует точка  $x^* \in [a, b]$  такая, что на полуинтервале  $[a, x^*)$  функция  $\Phi(x)$  убывает, а на полуинтервале  $(x^*, b]$ -возрастает. Заметим, что определение одномерного унимодального критерия оптимальности не требует непрерывности функции  $\Phi(x)$ . Например, на рис. 2 одномерная функция  $\Phi(x)$  на интервале  $[a, b]$  является унимодальной, хотя и имеет в точках  $x_1, x_2$  разрывы. Заметим также, что точка  $x^*$  может быть как внутренней точкой отрезка  $[a, b]$ , так и совпадать с одним из его концов.

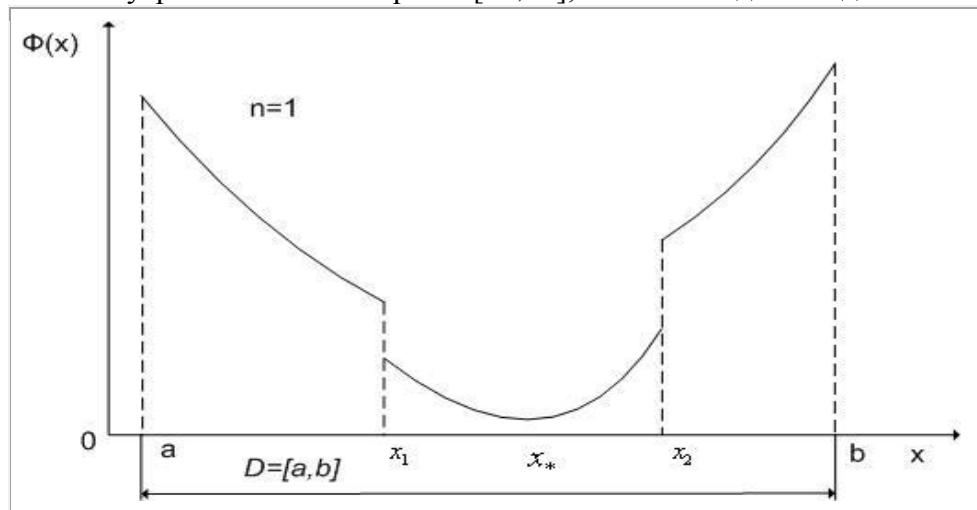


Рис. 2. К определению унимодального критерия оптимальности:  $x_1, x_2$  - точки разрыва критерия оптимальности  $\Phi(x)$ .

Непрерывный в своей области определения одномерный критерий оптимальности  $\Phi(x)$ ,  $x \in [a, b]$  называется выпуклым критерием оптимальности (выпуклым вниз критерием оптимальности), если для любых точек  $x_1, x_2 \in [a, b], x_1 \neq x_2$  выполняется неравенство  $\Phi(\lambda x_1 + (1-\lambda)x_2) \leq \lambda \Phi(x_1) + (1-\lambda)\Phi(x_2)$ ,

где число  $\lambda \in [0, 1]$ . Приведенное определение имеет простой геометрический смысл: если критерий оптимальности  $\Phi(x)$  выпукл на интервале  $[a, b]$ , то все точки любой дуги графика лежат под соответствующей хордой (рис. 3).

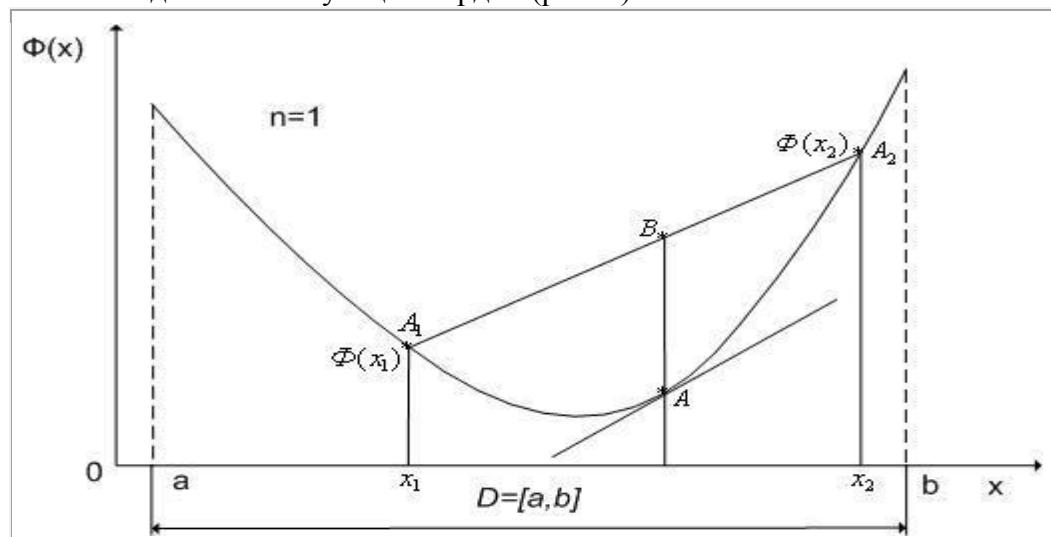


Рис. 3. К определению выпуклого одномерного критерия оптимальности.

Заметим, что определение выпуклого критерия оптимальности не требует его унимодальности. Так что, например, выпуклым является критерий оптимальности, график которого изображен на рис. 4.

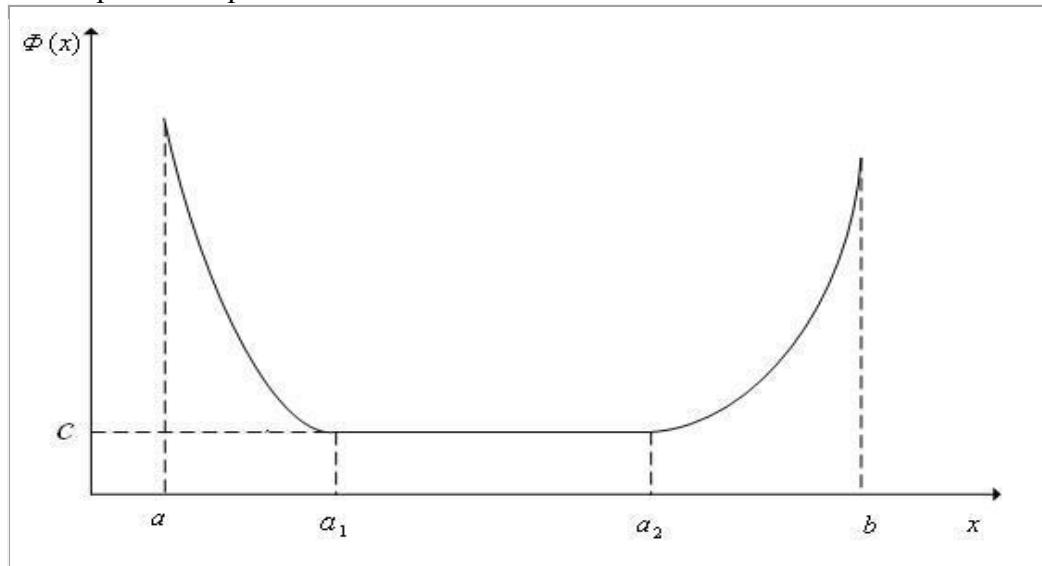


Рис. 4. Пример выпуклого критерия оптимальности: на интервале  $[a_1, a_2]$  значения критерия оптимальности постоянны и равны  $c$ .

Непрерывный в своей области определения одномерный критерий оптимальности  $\Phi(x)$ ,  $x \in [a, b]$  называется строго выпуклым критерием оптимальности (строго выпуклым вниз критерием оптимальности), если для любых точек  $x_1, x_2 \in [a, b]$ ,  $x_1 \neq x_2$  выполняется неравенство

$$\Phi(\lambda x_1 + (1-\lambda)x_2) < \lambda\Phi(x_1) + (1-\lambda)\Phi(x_2),$$

где произвольное число  $\lambda \in [0, 1]$ .

Строго вогнутый критерий является унимодальным критерием.

Если множество  $D$  является выпуклым множеством, то в многомерном случае  $n \geq 2$  также определено понятие выпуклого критерием оптимальности.

Непрерывный критерий оптимальности  $\Phi(\mathbf{X})$ , где  $\mathbf{X} \in D$  и множество  $D$  является выпуклым множеством, называется выпуклым критерием оптимальности (выпуклым вниз критерием оптимальности), если для любых  $\mathbf{X}_1, \mathbf{X}_2 \in D$ ,  $\mathbf{X}_1 \neq \mathbf{X}_2$  и любого  $\lambda \in [0, 1]$  выполняется неравенство

$$\Phi(\lambda \mathbf{X}_1 + (1-\lambda)\mathbf{X}_2) \leq \lambda\Phi(\mathbf{X}_1) + (1-\lambda)\Phi(\mathbf{X}_2).$$

Аналогично, критерий оптимальности  $\Phi(\mathbf{X})$ , где  $\mathbf{X} \in D$  и множество  $D$  является выпуклым множеством, называется строго выпуклым критерием оптимальности (строго выпуклым вниз критерием оптимальности), если для любых  $\mathbf{X}_1, \mathbf{X}_2 \in D$ ,  $\mathbf{X}_1 \neq \mathbf{X}_2$  и любого  $\lambda \in [0, 1]$  выполняется неравенство

$$\Phi(\lambda \mathbf{X}_1 + (1-\lambda)\mathbf{X}_2) < \lambda\Phi(\mathbf{X}_1) + (1-\lambda)\Phi(\mathbf{X}_2).$$

Отметим, что выпуклая функция может иметь более одной точки локального минимума (см. Пример 1.4.1), а строго выпуклая функция – только одну точку.

Пример 1

Рассмотрим выпуклую квадратичную функцию  $\Phi(\mathbf{X}) = \Phi(x_1, x_2) = (x_1 + x_2)^2$ . Легко видеть, что эта функция достигает в точке  $(0,0)$  минимума, равного нулю. Но это же значение

функция принимает во всех точках вида  $(x_1, x_2 = -x_1)$  - см. рис. 5. Заметим, что рис. 5 получен с помощью следующей MATLAB-программы:

```
x=-2:0.06:2;
y=x;
[X,Y]=meshgrid(x);
Z=(X+Y).^2;
V=[0.025,0.5,1,2,4,8];
[C,h]=contour(X,Y,Z,V);
clabel(C,h);
```

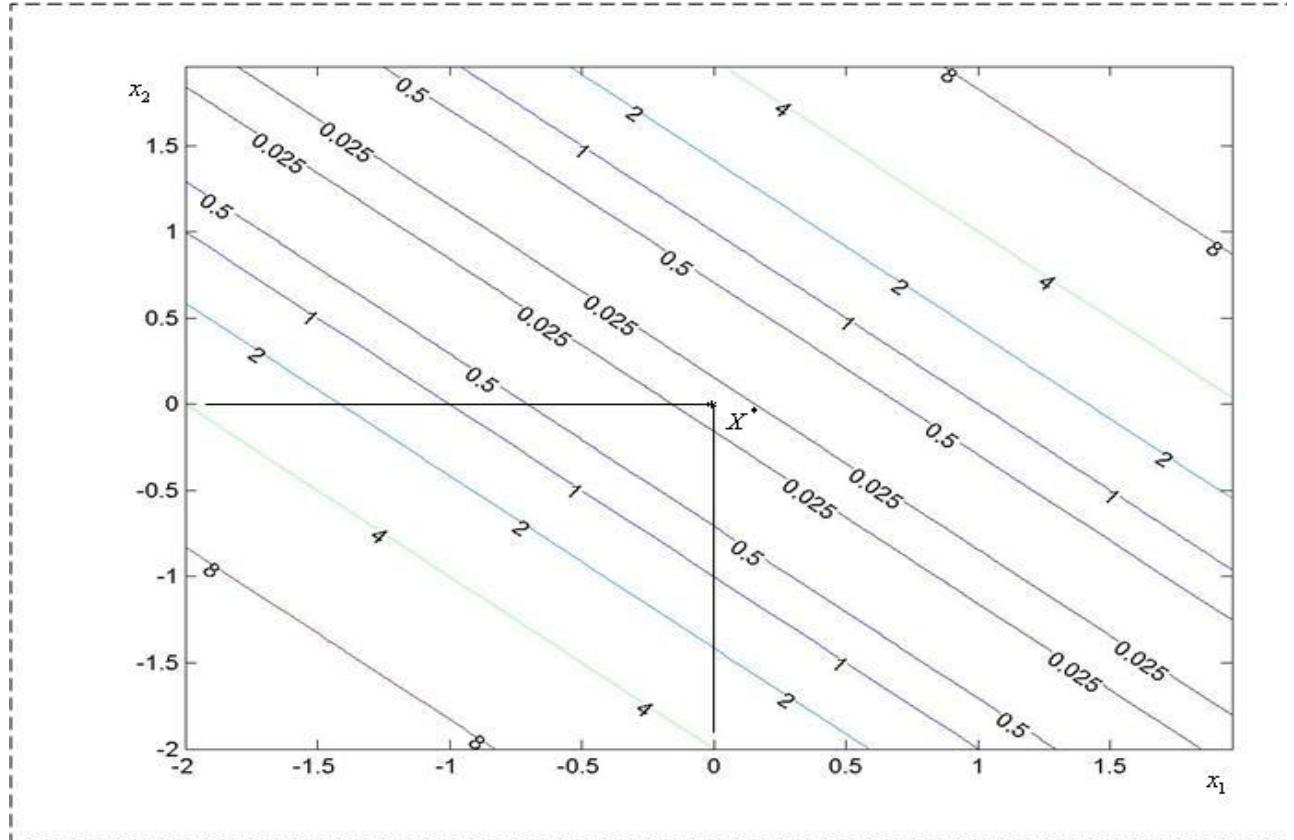


Рис. 5. К прим. 1.

Критерий оптимальности  $\Phi(\mathbf{X})$ , имеющий в области определения несколько локальных минимумов, называется многоэкстремальным критерием оптимальности или мультимодальным критерием оптимальности (см. прим. 2).

Если размерность вектора варьируемых параметров  $\mathbf{X}$  больше единицы ( $n > 1$ ), то критерий оптимальности  $\Phi(\mathbf{X})$  может быть в своей области допустимых значений  $D$  "овражным" критерием оптимальности. Критерий оптимальности называется овражным в своей области допустимых значений, если в этой области имеют место слабые изменения частных производных функции  $\Phi(\mathbf{X})$  по одним направлениям и значительные изменения этих производных по другим направлениям (см. прим. 2).

### Пример 2

Рассмотрим функцию Розенброка  $\Phi(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$  ( $n=2$ ). Легко видеть, что минимум этой функции достигается в точке  $(0,0)$  и равен нулю. Линии уровня

функции приведены на рис. 6, который получен с помощью следующей MATLAB-программы:

```
x=-2:0.06:2;
y=x;
[X,Y]=meshgrid(x);
Z=100.*((Y-X.^2).^2+(1-X).^2;
V=[1,5,50,500];
contour(X,Y,Z,V);
```

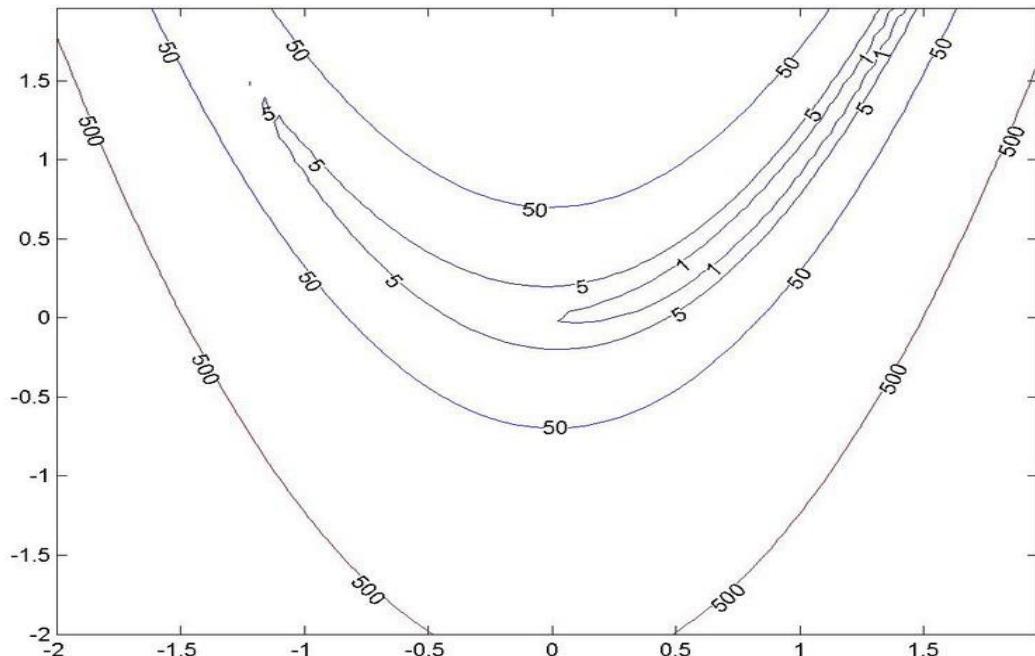


Рис. 6. Линии уровня функции Розенброка. Функция медленно изменяется вдоль дна V-образного оврага и быстро – перпендикулярно этому дну.

Критерий оптимальности  $\Phi(\mathbf{X})$  называется сепарабельным критерием оптимальности, если функция  $\Phi(\mathbf{X})$  является сепарабельной, т.е. представляет собой сумму функций, каждая из которых зависит только от одной компоненты вектора  $\mathbf{X}$ :

$$\Phi(\mathbf{X}) = \sum_{i=1}^n \phi_i(x_i).$$

Критерий оптимальности  $\Phi(\mathbf{X})$  называется позиномиальным критерием оптимальности, если функция  $\Phi(\mathbf{X})$  есть позином, т.е. если

$$\Phi(\mathbf{X}) = \sum_{i=1}^k c_i p_i(\mathbf{X}),$$

где  $c_i$  и все компоненты вектора  $\mathbf{X}$  – положительные действительные числа, а функции  $p_i(\mathbf{X})$  имеют вид

$$p_i(\mathbf{X}) = \prod_{j=1}^n x_j^{\alpha_{ij}},$$

$\alpha_{ij}$  - любые действительные числа.

## 1.5. Свойства выпуклых критериев оптимальности

### Одномерный критерий оптимальности

Если функция  $\Phi(x)$  является дифференцируемой, то можно сформулировать простые признаки ее выпуклости.

Теорема 1. Пусть критерий оптимальности  $\Phi(x)$  определен и непрерывен на интервале  $[a, b]$  и имеет в нем конечную первую производную  $\Phi'(x)$ . Для того, чтобы функция  $\Phi(x)$  была выпуклой (строго выпуклой) на  $[a, b]$ , необходимо и достаточно, чтобы ее производная  $\Phi'(x)$  не убывала (возрастала) на этом интервале •

Теорема 2 Пусть критерий оптимальности  $\Phi(x)$  определен и непрерывен вместе со своей первой производной  $\Phi'(x)$  на интервале  $[a, b]$  и имеет в нем конечную вторую производную  $\Phi''(x)$ . Для выпуклости функции  $\Phi(x)$  на интервале  $[a, b]$  необходимо и достаточно, чтобы внутри этого интервала имело место неравенство  $\Phi''(x) \geq 0$  •

Заметим, что условие  $\Phi''(x) > 0$  не является необходимым для строгой выпуклости функции  $\Phi(x)$ . Так функция  $\Phi(x) = x^4$  является строго выпуклой на всей числовой оси, хотя в начале координат ее вторая производная равна нулю:  $\Phi''(0) = 0$ .

Укажем еще одну очевидную, но важную геометрическую характеристику выпуклого критерия оптимальности. Если критерий оптимальности  $\Phi(x)$  определен на интервале  $[a, b]$ , имеет на этом интервале конечную первую производную  $\Phi'(x)$  и выпукл, то график функции  $\Phi(x)$  всеми своими точками лежит над или на любой своей касательной (см. Рис.4.3).

### Многомерный критерий оптимальности

Суть понятия вогнутого (строго вогнутого) критерия оптимальности в многомерном случае удобно определить с помощью сечения. Сечением критерия оптимальности  $\Phi(\mathbf{X})$ ,  $\mathbf{X} \in D$ , где  $D$  - выпуклое множество, называется одномерная функция  $\varphi(\lambda) = \lambda\Phi(\mathbf{X}_1) + (1-\lambda)\Phi(\mathbf{X}_2)$ , где  $\lambda \in [0, 1]$  - вещественный скаляр, а  $\mathbf{X}_1, \mathbf{X}_2$  - любые точки множества  $D$ .

Теорема 3. Для того, чтобы критерий оптимальности  $\Phi(\mathbf{X})$ , определенный на выпуклом множестве  $D$ , был выпуклым критерием (строго выпуклым критерием), необходимо и достаточно, чтобы любое сечение этого критерия было выпуклой функцией (строго выпуклой функцией) •

Теорема 4. Если критерий оптимальности  $\Phi(\mathbf{X})$ , определенный на выпуклом множестве  $DCR^n$ , является дважды дифференцируемым на этом множестве, то необходимым и достаточным условием его выпуклости является неотрицательная определенность на этом множестве матрицы вторых производных функции  $\Phi(\mathbf{X})$  – матрицы Гессе (гессиана)

$$\mathbf{H}(\mathbf{X}) = \frac{\partial^2 \Phi(\mathbf{X})}{\partial x_i \partial x_j}, i, j \in [1, n].$$

Замечание. Симметричная ( $n \times n$ ) матрица  $\mathbf{A}$  называется неотрицательно определенной, если квадратичная форма  $\mathbf{X}^T \mathbf{A} \mathbf{X} = (\mathbf{AX}, \mathbf{X})$  для любых векторов  $\mathbf{X} \in R^n$  принимает неотрицательные значения. Если матрица  $\mathbf{A}$  неотрицательно определена, то ее

собственные числа являются неотрицательными. Для проверки неотрицательной определенности матрицы удобно использовать критерий Сильвестра: если все главные миноры матрицы  $\mathbf{A}$  неотрицательны, то матрица  $\mathbf{A}$  неотрицательно определена •  
По аналогии с одномерным случаем, положительная определенность матрицы Гессе не является достаточным условием строгой выпуклости функции  $\Phi(\mathbf{X})$ . Так функция  $\Phi(\mathbf{X}) = x_1^4 + x_2^4$  строго выпукла в пространстве  $R^2$ , но ее матрица Гессе в точке (0,0) является нулевой:  $\mathbf{H}(0,0)=0$ .

Наряду с матрицей Гессе нам далее потребуется  $n$ -мерный вектор градиента функции  $\Phi(\mathbf{X})$

$$\nabla \Phi(\mathbf{X}) = \left\{ \frac{\partial \Phi(\mathbf{X})}{\partial x_i}, i \in [1, n] \right\}$$

## 1.6. Классификация детерминированных задач оптимизации.

Рассмотрим детерминированную задачу оптимизации

$$\min_{\mathbf{X} \in D} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где область допустимых значений  $D \subset R^n$ ,

$$D = \{\mathbf{X} | g(\mathbf{X}) \geq 0, h(\mathbf{X}) = 0\} = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0, h_j(\mathbf{X}) = 0, i \in [1, m], j \in [1, l]\}. \quad (2)$$

Классификация задачи (1.2) возможна по многим признакам. Рассмотрим основные из этих признаков.

Классификация по виду критерия оптимальности и ограничивающих функций.

Если критерий оптимальности  $\Phi(\mathbf{X})$  – линейная функция, а множество  $D$  – выпуклый многогранник, то задача (1),(2) называется задачей линейного программирования.

Если критерий оптимальности  $\Phi(\mathbf{X})$  – есть отношение двух линейных функций, а множество  $D$  – выпуклый многогранник, задача (1),(2) называется задачей дробно-линейного программирования.

Пусть область  $D$  определяется только ограничениями типа неравенств:

$$D = \{\mathbf{X} | g(\mathbf{X}) \geq 0\} = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0, i \in [1, m]\}. \quad (3)$$

Тогда если функция  $\Phi(\mathbf{X})$  и функции  $g_i(\mathbf{X}), i \in [1, m]$  являются сепарабельными, то задача (1), (3) называется задачей сепарабельного программирования.

Тогда если функция  $\Phi(\mathbf{X})$  и ограничивающие функции  $g_i(\mathbf{X}), i \in [1, m]$  являются позиномами, то задача (1), (3) называется задачей геометрического программирования.

Если  $\Phi(\mathbf{X})$  – квадратичная функция, т.е.  $\Phi(\mathbf{X}) = \mathbf{X}^T \mathbf{G} \mathbf{X} + \mathbf{C}^T \mathbf{X}$ , а множество  $D$  есть выпуклое множество, то задача (1),(2) называется задачей квадратичного программирования. Здесь  $\mathbf{G}_{-(n * n)}$  симметричная матрица,  $\mathbf{C}_{-(n * 1)}$  вектор.

Задачи линейного, дробно-линейного, сепарабельного и геометрического программирования редко возникают в САПР и в данном курсе не рассматриваются.

Если множество  $D$  является конечным множеством, то задача (1), (2) называется задачей дискретного программирования.

Если множество  $D$  является множеством целых чисел, то задача (1), (2) называется задачей целочисленного программирования.

Задачи дискретного и целочисленного программирования обычно изучаются в курсах исследования операций и в данном курсе не рассматриваются.

Если функция  $\Phi(\mathbf{X})$  является выпуклой, то задача (1,2) называется задачей выпуклого программирования. Заметим, что определение выпуклой функции  $\Phi(\mathbf{X})$  требует выпуклости области ее определения  $D$ .

В общем случае задача (1),(2) называется задачей нелинейного программирования. Часто задачи выпуклого программирования также относят к задачам нелинейного программирования.

Классификация по наличию или отсутствию ограничений.

Если ограничения на вектор  $\mathbf{X}$  отсутствуют ( $D = \mathbb{R}^n$ ), то задача (1),(2) называется задачей оптимизации без ограничений или задачей безусловной оптимизации.

Если имеются ограничения на вектор  $\mathbf{X}$  ( $D \subset \mathbb{R}^n$ ) то задача (1),(2) называется задачей оптимизации с ограничениями или задачей условной оптимизации.

Классификация характеру ограничений.

Среди задач условной оптимизации выделяют следующие классы задач:

задачи условной оптимизации с ограничениями типа неравенств, когда  $D = \{\mathbf{X} | g(\mathbf{X}) \geq 0\} = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0, i \in [1, m]\}$ ;

задачи условной оптимизации с ограничениями типа равенств, когда  $D = \{\mathbf{X} | h(\mathbf{X}) = 0\} = \{\mathbf{X} | h_j(\mathbf{X}) = 0, j \in [1, l]\}$ ;

задачи условной оптимизации с ограничениями общего вида, когда имеются как ограничения типа неравенств, так и ограничения типа равенств, т.е. когда  $D = \{\mathbf{X} | g(\mathbf{X}) \geq 0, h(\mathbf{X}) = 0\} = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0, h_j(\mathbf{X}) = 0, i \in [1, m], j \in [1, l]\}$ .

Классификация по размерности вектора  $\mathbf{X}$ .

Если размерность вектора  $\mathbf{X}$  равна 1 ( $n=1$ ), то задача (1),(2) называется однопараметрической задачей оптимизации (одномерной задачей оптимизации).

Если размерность вектора  $\mathbf{X}$  больше 1 ( $n > 1$ ), то задача (1),(2) называется многопараметрической задачей оптимизации (многомерной задачей оптимизации).

Классификация по количеству точек минимума.

Если функция  $\Phi(\mathbf{X})$  имеет в области допустимых значений  $D$  один минимум, то задача (1),(2) называется одноэкстремальной задачей оптимизации.

Если функция  $\Phi(\mathbf{X})$  имеет в области допустимых значений  $D$  более одного минимума, то задача (1),(2) называется многоэкстремальной задачей оптимизации.

Классификация по характеру искомого решения.

Если отыскивается любой локальный минимум функции  $\Phi(\mathbf{X})$ , то задача (1),(2) называется задачей локальной оптимизации. Если отыскивается любой локальный минимум функции  $\Phi(\mathbf{X})$  и задача (1),(2) является задачей безусловной оптимизации, то эта задача называется задачей безусловной локальной оптимизации. Аналогично, если отыскивается любой локальный минимум функции  $\Phi(\mathbf{X})$  и задача (1),(2) является задачей условной оптимизации, то эта задача называется задачей условной локальной оптимизации.

Если отыскивается глобальный минимум функции  $\Phi(\mathbf{X})$ , то задача (1),(2) называется задачей глобальной оптимизации. Если отыскивается глобальный минимум функции  $\Phi(\mathbf{X})$  и задача (1),(2) является задачей безусловной оптимизации, то эта задача называется задачей безусловной глобальной оптимизации. Аналогично, если отыскивается глобальный минимум функции  $\Phi(\mathbf{X})$  и задача (1),(2) является задачей условной оптимизации, то эта задача называется задачей условной глобальной оптимизации.

## 2.1. Примеры задач линейного программирования

Преобладающая часть рассматриваемых содержательных экономических задач являются задачами линейного программирования, но чтобы ввести основные понятия, достаточно рассмотреть две классические задачи.

**Задача 1. Планирование производства.** Пусть возможен выпуск  $n$  различных видов продукции, которые обозначим  $\Pi_j$ , где  $(j = \overline{1, n})$ . Используемые для этого ресурсы (виды сырья, группы оборудования, рабочая сила различной квалификации, производственные площади, финансовые средства и т. д.) общим числом  $m$ , ограничены величинами  $b_i$ , где  $(i = \overline{1, m})$ .

Обычно построение модели начинают с определения ее переменных. В рассматриваемой задаче за переменные естественно принять объемы выпуска каждого из возможных видов продукции, которые традиционно обозначают  $x_j$ , где  $(j = \overline{1, n})$ .

Будем искать такой план выпуска продукции, который обеспечит максимальную прибыль  $Z$ . Это наиболее часто используемая, хотя далеко не единственная, трактовка поставленной задачи. Но преимущество математического подхода в том и состоит, что обеспечивается точность всех формулировок. В данном случае достаточно знать прибыль от производства единицы каждого вида продукции, которую обозначим  $c_j$ , где  $(j = \overline{1, n})$ .

Тогда получаем задачу:

$$\text{Максимизировать } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n.$$

Осталось сформулировать ограничения, которые состоят в том, что:

$$\left\{ \begin{array}{l} \text{Необходимый объем ресурсов} \\ \text{для производства всей продукции} \end{array} \right\} \leq \left\{ \begin{array}{l} \text{Располагаемые} \\ \text{ресурсы} \end{array} \right\}.$$

Для записи этих ограничений необходимо знать коэффициенты затрат ресурсов на производство единицы каждого вида продукции, которые удобно представить в табличной форме. Чтобы сгруппировать все используемые данные, занесем в эту таблицу и коэффициенты целевой функции.

Таблица 2.1

Вид ресурса, $i$	Расход $i$ -го ресурса на единицу $j$ -й продукции, $a_{ij}$				Запас ресурса, $b_i$
	1	2	...	$n$	
1	$a_{11}$	$a_{12}$	...	$a_{1n}$	$b_1$
2	$a_{21}$	$a_{22}$	...	$a_{2n}$	$b_2$
...	...	...	...	...	...
$m$	$a_{m1}$	$a_{m2}$	...	$a_{mn}$	$b_m$
Прибыль, $c_j$	$c_1$	$c_2$	...	$c_n$	

Математическая модель задачи примет вид:

$$\begin{aligned}
\max Z = & c_1x_1 + c_2x_2 + \dots + c_nx_n; \\
& a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1; \\
& a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2; \\
& \dots \dots \dots \dots \dots \dots \\
& a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m; \\
& x_1 \geq 0, \quad x_2 \geq 0, \quad \dots, \quad x_n \geq 0.
\end{aligned}
\right\}$$

Дополнительно введенные условия неотрицательности переменных очевидным образом определяются экономическим смыслом задачи, но если их не задать при обращении к используемому пакету прикладных программ, то весьма вероятно, что решение не будет получено. Действительно, предположим, что один из видов продукции, требующий расхода всех видов ресурсов, дает весьма незначительную прибыль. Если соответствующей переменной формально придать отрицательное значение, то получим прирост всех ресурсов при малых затратах (соответствующие слагаемые при переносе в правую часть получат знак плюс). Тогда прибыль будет неограниченно возрастать за счет увеличения производства более выгодных видов продукции.

Следует заметить, что при решении реальных экономических задач приведенные выше ресурсные ограничения надо будет дополнять условиями, учитывающими ограниченные возможности реализации продукции и имеющиеся обязательства по ее поставке, взаимосвязи между отдельными видами продукции и т. д. Поэтому процесс построения хорошей модели и является искусством, требующим знания предметной области и ясного экономического мышления.

**Задача 2. Формирование минимальной продовольственной корзины.** Эту задачу часто называют задачей о рационе. Пусть определен ассортимент продуктов и известны их характеристики (цена; калорийность; содержание белков, жиров, углеводов, витаминов). Требуется найти набор минимальной стоимости, обеспечивающий формирование минимальной продовольственной корзины.

Введем обозначения:

- $n$  – число рассматриваемых продуктов;
- $m$  – количество учитываемых веществ (в т. ч. калорий);
- $a_{ij}$  – содержание  $i$ -го вещества в  $j$ -м продукте, ( $i = \overline{1, m}$ ); ( $j = \overline{1, n}$ );
- $b_i$  – потребность человека в  $i$ -м веществе, ( $i = \overline{1, m}$ );
- $c_j$  – стоимость единицы  $j$ -го продукта, ( $i = \overline{1, m}$ )
- $x_j$  – количество  $j$ -го продукта в решении, ( $j = \overline{1, n}$ ).

Получим модель, симметричную по отношению к ранее записанной. Только вместо максимума теперь нужно найти минимум, изменились и знаки ограничений

$$\begin{aligned}
\min Z = & c_1x_1 + c_2x_2 + \dots + c_nx_n; \\
& a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1; \\
& a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2; \\
& \dots \dots \dots \dots \dots \dots \\
& a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \geq b_m; \\
& x_1 \geq 0, \quad x_2 \geq 0, \quad \dots, \quad x_n \geq 0.
\end{aligned}
\right\}$$

Разумеется, и в этом случае может потребоваться введение определенных соотношений между отдельными группами продуктов, ограничение содержания некоторых ингредиентов и т. д.

## 2.2. Формы записи задачи линейного программирования

*Общая форма задачи линейного программирования.* Так называют задачу найти

$$\max (\min) Z = \sum_{j=1}^n c_j x_j$$

при ограничениях

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad (i = \overline{1, m_1});$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad (i = \overline{m_1 + 1, m_2});$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad (i = \overline{m_2 + 1, m_3});$$

$$x_j \geq 0, \quad (j = \overline{1, n_1});$$

$x_j$  – любого знака  $(j = \overline{n_1 + 1, n_2})$ .

*Симметричная форма задачи линейного программирования.* Примеры таких задач уже приведены в предыдущем параграфе. Это

$$\max Z = \sum_{j=1}^n c_j x_j ;$$

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

$$x_j \geq 0, \quad (j = \overline{1, n})$$

или

$$\min Z = \sum_{j=1}^n c_j x_j;$$

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

$$x_j \geq 0, \quad (j = \overline{1, n}).$$

*Каноническая форма задачи линейного программирования* используется, чтобы применить аппарат линейной алгебры для решения оптимизационных задач. Предполагается, что задача решается только на максимум и что все условия записаны как равенства

$$\max Z = \sum_{j=1}^n c_j x_j ;$$

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

$$x_j \geq 0, \quad (j = \overline{1, n}).$$

Если ввести обозначения:

$\mathbf{c} = [c_1 \ c_2 \ \dots \ c_n]$  – матрица-строка коэффициентов целевой функции;

$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{11} & a_{12} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$  – матрица системы уравнений;

$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$  – матрица-столбец переменных;

$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$  – матрица-столбец свободных членов,

то каноническая форма задачи в матричной записи примет вид:

$$\max Z = \mathbf{c}\mathbf{x};$$

$$\mathbf{Ax} = \mathbf{b}, \quad \mathbf{x} \geq 0,$$

где  $\mathbf{c}\mathbf{x}$  – скалярное произведение векторов  $\mathbf{c} = (c_1 \ c_2 \ \dots \ c_n)$  и  $\mathbf{x} = (x_1 \ x_2 \ \dots \ x_n)$ .

Используется и векторная форма записи. Если обозначить:

$$\mathbf{A}_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \dots \\ a_{m1} \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} a_{12} \\ a_{22} \\ \dots \\ a_{m2} \end{bmatrix}, \dots, \quad \mathbf{A}_j = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \dots \\ a_{mj} \end{bmatrix}, \dots, \quad \mathbf{A}_n = \begin{bmatrix} a_{1n} \\ a_{2n} \\ \dots \\ a_{nn} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix},$$

то получим:

$$\max Z = \mathbf{c}\mathbf{x}$$

$$\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \dots + \mathbf{A}_j x_j + \dots + \mathbf{A}_n x_n = \mathbf{b}, \quad x_j \geq 0, \quad (j = \overline{1, n}).$$

## 2.3. Приведение задачи линейного программирования к канонической форме

Каноническая форма введена для обоснования метода поиска оптимального решения, но экономические задачи первоначально записываются в общей или симметричной форме. Покажем, что путем эквивалентных преобразований всегда можно перейти к канонической форме.

**Преобразование целевой функции.** Задачу поиска минимума легко заменить задачей поиска максимума. Если  $x^*$  – точка минимума функции

$y = f(x)$ , то для функции  $y = -f(x)$  она будет точкой максимума.

Очевидно, что поиск максимума можно заменить поиском минимума и что эти утверждения справедливы и для функции  $n$  переменных:

$$\min(f(x_1^*, x_2^*, \dots, x_n^*)) = -\max(-f(x_1^*, x_2^*, \dots, x_n^*)).$$

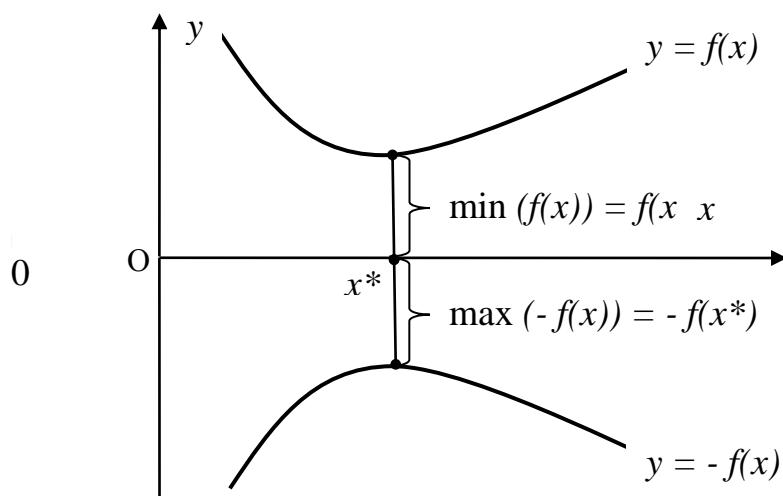


Рис. 2.1

**Замена неравенства типа «меньше или равно» на равенство.** Для каждого ограничения типа  $a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b$  введем свою дополнительную неотрицательную переменную  $x_{n+1}$ . Если вектор  $\mathbf{x} = (x_1 \ x_2 \ \dots \ x_n)$  удовлетворяет исходному ограничению, то всегда можно подобрать ее значение так, чтобы выполнялось равенство

$$a_1x_1 + a_2x_2 + \dots + a_nx_n + x_{n+1} = b.$$

Если это равенство выполняется при некотором неотрицательном  $x_{n+1}$ , то будет выполняться и рассматриваемое ограничение.

Ограничения типа «меньше или равно» были использованы при построении модели оптимального планирования производства. Заменяя их равенствами, получим:

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + x_{n+1} = b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + x_{n+2} = b_2; \\ \vdots \quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + x_{n+m} = b_m; \\ x_j \geq 0, \quad (j = \overline{1, n+m}). \end{array} \right\}$$

Эти условия часто называют «ограничения-равенства», так как равенствами они являются только по форме, а по содержанию – ресурсными ограничениями. Каждое из них содержит свою дополнительную переменную  $x_{n+k}$ , где ( $k = \overline{1, m}$ ). Иногда такие переменные называют остаточными, поскольку их значения показывают величины неиспользованных ресурсов. Если продажа (сдача в аренду) свободных ресурсов не предполагается, то в целевую функцию модели дополнительные переменные войдут с нулевыми коэффициентами, иначе говоря, функционал не изменится.

**Замена неравенства типа «больше или равно» на равенство** выполняется аналогично. От левой части каждого ограничения типа  $a_1x_1 + a_2x_2 + \dots + a_nx_n \geq b$  отнимем дополнительную неотрицательную переменную  $x_{n+1}$ , чтобы обеспечить выполнение равенства  $a_1x_1 + a_2x_2 + \dots + a_nx_n - x_{n+1} = b$ . Если это равенство выполняется при некотором неотрицательном  $x_{n+1}$ , то выполняется и исходное ограничение. Такие ограничения были использованы для построения модели формирования минимальной продовольственной корзины. После замены неравенства «больше или равно» равенством получим:

$$\left. \begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n - x_{n+1} &= b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n - x_{n+2} &= b_2; \\ \vdots & \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n - x_{n+m} &= b_m; \\ x_j \geq 0 \quad (j = \overline{1, n+m}) & \end{aligned} \right\}$$

В этой модели содержательный смысл дополнительных переменных в том, что они показывают превышение содержания отдельных веществ в рационе сверх заданных минимальных величин. Поэтому их иногда называют избыточными. Очевидно, что в целевую функцию модели они войдут с нулевыми коэффициентами. При расчете подобной модели обеспечение норм по белкам и витаминам приведет, вероятно, к избыточной насыщенности.

**Замена переменных любого знака на неотрицательные** выполняется путем замещения их разностью неотрицательных переменных. Действительно, такую переменную  $x_j$  можно представить как  $x_j = u_j - v_j$ , где  $u_j$  и  $v_j$  неотрицательные переменные.

Итак, доказано, что от любой формы записи линейной модели можно перейти к канонической форме. Справедливо и обратное, равенство эквивалентно двум

неравенствам с противоположными знаками:  $\sum_{j=1}^n a_{ij}x_j = b_i$  равносильно  $\sum_{j=1}^n a_{ij}x_j \leq b_i$  и

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

## 2.4. Графическое построение области допустимых решений

Задачи линейного программирования могут содержать тысячи переменных, однако на примере двух переменных можно наглядно показать структуру задачи, идею ее решения и возникающие проблемы.

Начнем с одного ограничения вида  $a_1x_1 + a_2x_2 \leq b$ . Определим множество точек плоскости, удовлетворяющих этому условию. Сначала построим прямую линию  $a_1x_1 + a_2x_2 = b$  и получим две полуплоскости. Для всех точек одной из них ограничение будет выполняться, а для другой нет, что и показано на рисунке. При его построении принято, что  $x_1 \geq 0$  и  $x_2 \geq 0$ , как это имеет место в реальных задачах, следовательно, можно ограничиться только первым квадрантом координатной плоскости.

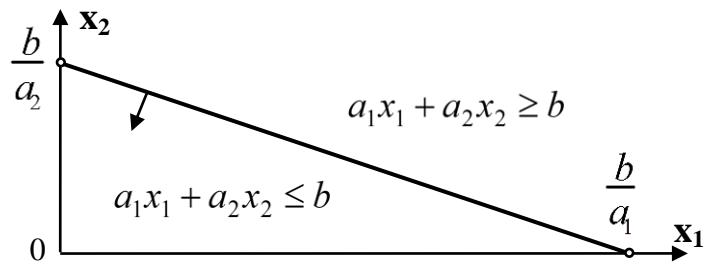


Рис. 2.2

Для построения прямой проще всего найти точки ее пересечения с координатными осями, полагая поочередно одну из координат равной нулю. Чтобы определить, какая именно полуплоскость нужна, достаточно взять произвольную точку, не принадлежащую построенной прямой, и подставить ее координаты в неравенство. Если в качестве такой точки взять начало координат, то при  $b > 0$  ограничение будет выполняться. На рис. 2.2 это показано стрелкой, указывающей на выделяемую полуплоскость. Второй вариант выделения полуплоскости – около проведенной прямой с нужной стороны наносится короткая штриховка.

Для определения полуплоскости можно рассуждать и так: рассматриваемое ограничение при  $a_1 > 0, a_2 > 0, b > 0$  – это ресурсное ограничение, которое не исключает неограниченный рост любой из переменных, но допускает их умеренные значения.

Ограничение вида  $a_1x_1 + a_2x_2 \geq b$  при положительных коэффициентах не может выполняться при малых (в частности, нулевых) значениях обеих переменных одновременно, т. е. не ограничивает их рост. Ясно, что надо выделять полуплоскость, не содержащую начало координат.

Рассмотрим частный случай, когда  $b = 0$ , а знаки коэффициентов  $a_1$  и  $a_2$  разные, например,  $a_1x_1 - a_2x_2 \geq 0$ . Такая прямая (рис. 2.3) проходит через начало координат, но для ее построения нужна еще одна точка.

Проще всего взять точку  $x_1 = a_2, x_2 = a_1$ . Чтобы правильно выделить полуплоскость достаточно заметить, что при  $x_2 = 0$  и любом  $x_1 \geq 0$  ограничение удовлетворяется.

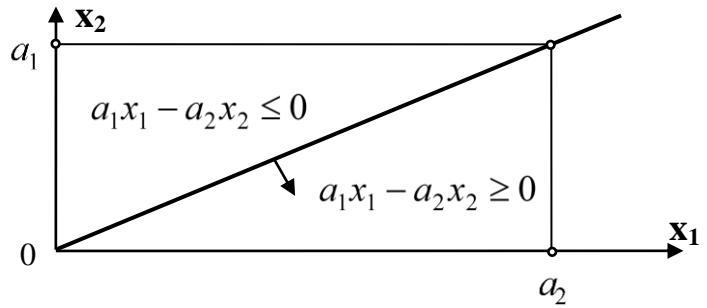
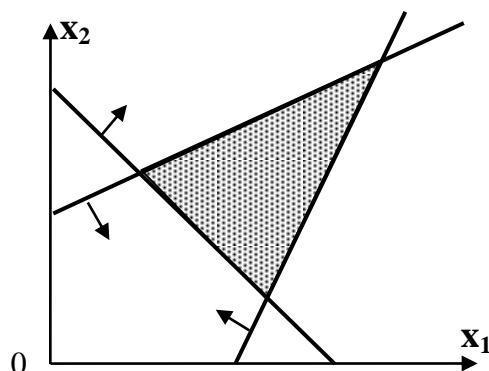
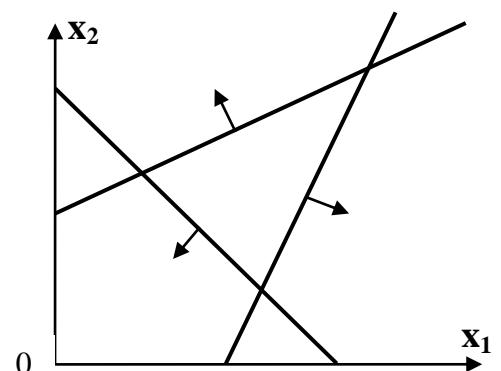


Рис 2.3

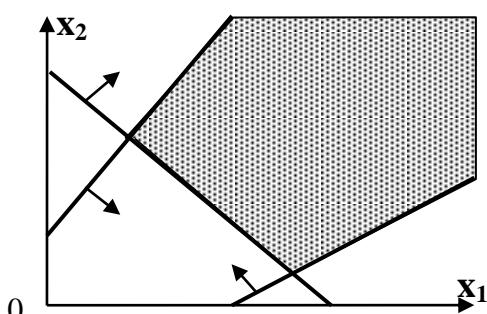
Если последовательно нанести все заданные ограничения, то в зависимости от их сочетания получатся различные варианты области допустимых решений (рис. 2.4).



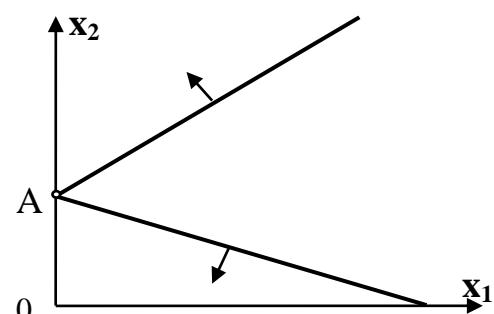
Выпуклый многоугольник



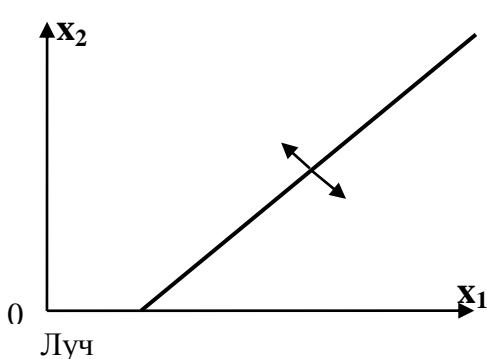
Пустое множество



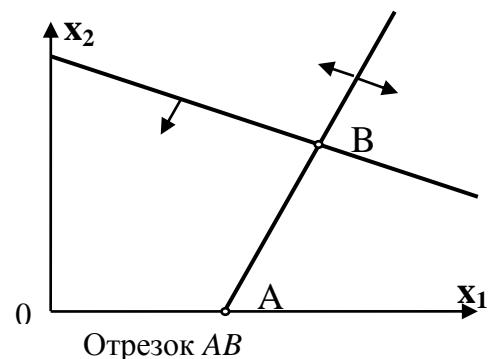
Выпуклая неограниченная  
допустимая область



Точка A



Луч



Отрезок AB

Рис 2.4

Будем предполагать переменные неотрицательными, что не ограничивает общность, и перечислим все возможные случаи. Например, при трех ограничениях может получиться как треугольник, так и пустое множество. При увеличении числа ограничений, возрастет и число вершин, но многоугольник всегда будет выпуклым. Для случая пустого множества можно указать области, в которых выполняются любые два ограничения, но нет ни одной точки, удовлетворяющей всем трем ограничениям.

Когда одно из ограничений является равенством, что равносильно заданию двух неравенств с разными знаками, получаются луч и отрезок. Направленные в противоположные стороны стрелки показывают, что только точки прямой могут принадлежать области допустимых решений.

Во всех этих случаях получаются выпуклые фигуры. По определению, множество называется выпуклым, если каждая точка отрезка, соединяющего любые две его точки, также принадлежит этому множеству (см. рис. 2.5).

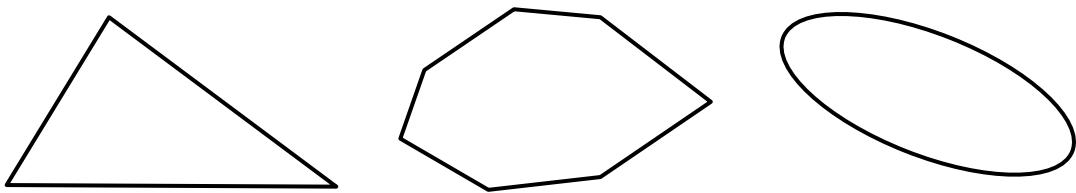


Рис. 2.5

Ниже приведены примеры невыпуклых множеств (рис. 2.6).

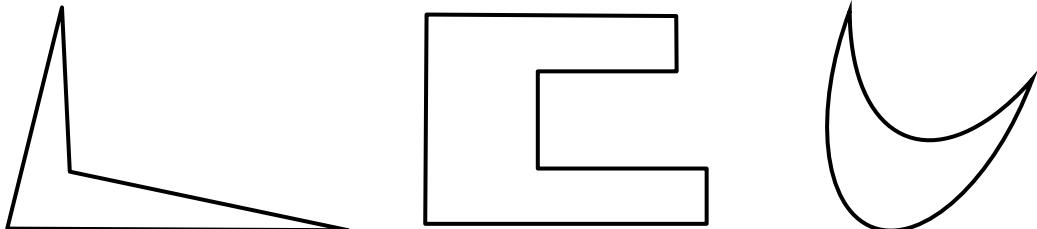


Рис. 2.6

## 2.5. Графическое нахождение оптимального решения

Предположим, что область допустимых решений – непустое множество, содержащее более одной точки. Если множество состоит из одной точки, то оптимизация не требуется, достаточно эту точку найти. В общем же случае необходимо определить в области допустимых решений точку (или точки), в которой функционал  $Z = c_1x_1 + c_2x_2$  приобретает наибольшее или наименьшее значение. Возьмем произвольное фиксированное значение  $Z = Z_1$ . Эти значения функционала будет принимать во всех точках некоторой прямой. Если изменить это значение, например, принять  $Z = Z_2$ , то прямая, соответствующая новому значению функционала, будет параллельна уже построенной (см. рис. 2.7).

Теперь из семейства параллельных прямых, соответствующих фиксированным значениям функционала, следует выделить те две прямые, при которых достигаются его минимальное и максимальное значения. Эти прямые будут соприкасаться с областью допустимых решений, но не могут ее пересекать. При построении за область допустимых решений принят многоугольник, а линии фиксированных значений функционала изображены пунктиром; принято также, что  $c_1 > 0, c_2 > 0$ .

При возрастании значения функционала, прямая, в точках которой эти значения достигаются, будет удаляться от начала координат. Первое соприкосновение с областью допустимых решений произойдет в точке  $\mathbf{x}^*_{\min}$ , последнее в точке  $\mathbf{x}^*_{\max}$ .

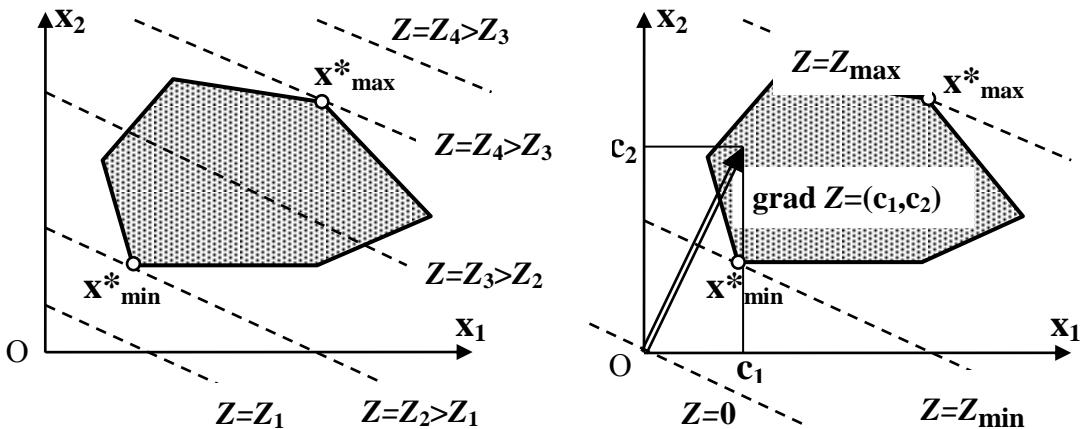


Рис 2.7

Направление наискорейшего возрастания любой дифференцируемой функции определяется вектором градиента, координаты которого есть ее частные производные по всем переменным. Если  $Z = Z(x_1, x_2, \dots, x_n)$ , то  $\text{grad } Z = \left( \frac{\partial Z}{\partial x_1}, \frac{\partial Z}{\partial x_2}, \dots, \frac{\partial Z}{\partial x_n} \right)$ . Для рассматриваемой линейной модели функционал имеет вид  $Z = c_1x_1 + c_2x_2$  и, следовательно,  $\text{grad } Z = (c_1, c_2)$  во всех точках плоскости. Отложим вектор градиента из начала координат, тогда, поскольку любая прямая (для которой значения функционала постоянны) перпендикулярна вектору градиента, можно найти координаты искомых экстремумов. При реализации этого способа удобно сначала проводить прямую, перпендикулярную вектору градиента, через начало координат, чтобы не ошибиться в построении прямых, проходящих через точки экстремумов.

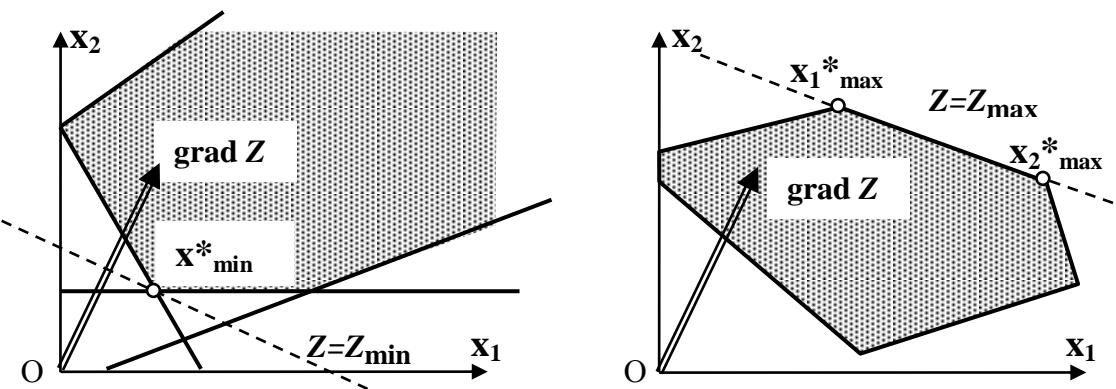


Рис. 2.8

Для случая двух переменных очевидно, что одна из угловых точек всегда обеспечит искомый экстремум, если только он существует. Например, когда область допустимых планов неограниченная, можно найти только минимальное значение функционала (см. рис. 2.8). Но экстремум может одновременно достигаться в двух угловых точках ( $x_1^{*\max}$  и  $x_2^{*\max}$ ), тогда во всех точках отрезка, соединяющего эти угловые точки, функционал будет иметь то же самое значение. В этом случае существует бесконечное множество решений.

Очевидно, что решение нельзя найти в случае пустого множества допустимых планов. Нет оснований для оптимизационного подхода и тогда, когда это множество состоит из одной единственной точки. В подобных ситуациях, а они часто возникают на нижних уровнях управления при директивном планировании, необходим пересмотр постановки задачи.

## 2.6. Графический анализ чувствительности

В действительности исходные данные для модели не только определяются с некоторой погрешностью, но и изменяются по различным причинам. Конечно, всегда можно, изменения отдельные параметры, рассчитать серию вариантов, чтобы оценить их влияние на оптимальное решение. Такой прием, как правило, полезен на заключительном этапе анализа задачи, но простой перебор вариантов не лучший способ анализа ситуации.

Рассмотрим, основываясь на графическом решении, основные идеи анализа чувствительности. Определим, как влияют изменения величин коэффициентов целевой функции и правых частей ограничений на оптимальное решение. Пусть задана линейная модель следующего вида:

$$\begin{aligned} \max (\min) Z &= 4x_1 + 8x_2 \\ 3x_1 + 4x_2 &\leq 60 \quad (1) \\ 10x_1 + 9x_2 &\leq 180 \quad (2) \\ 2x_1 + 5x_2 &\geq 30 \quad (3) \\ 6x_1 - 2x_2 &\geq 0 \quad (4) \\ x_1 &\geq 0, x_2 \geq 0 \end{aligned} \quad .$$

При построении удобно нумеровать все заданные ограничения. Тогда легко проверять правильность построения как линий, так и области допустимых решений (см. рис. 2.9).

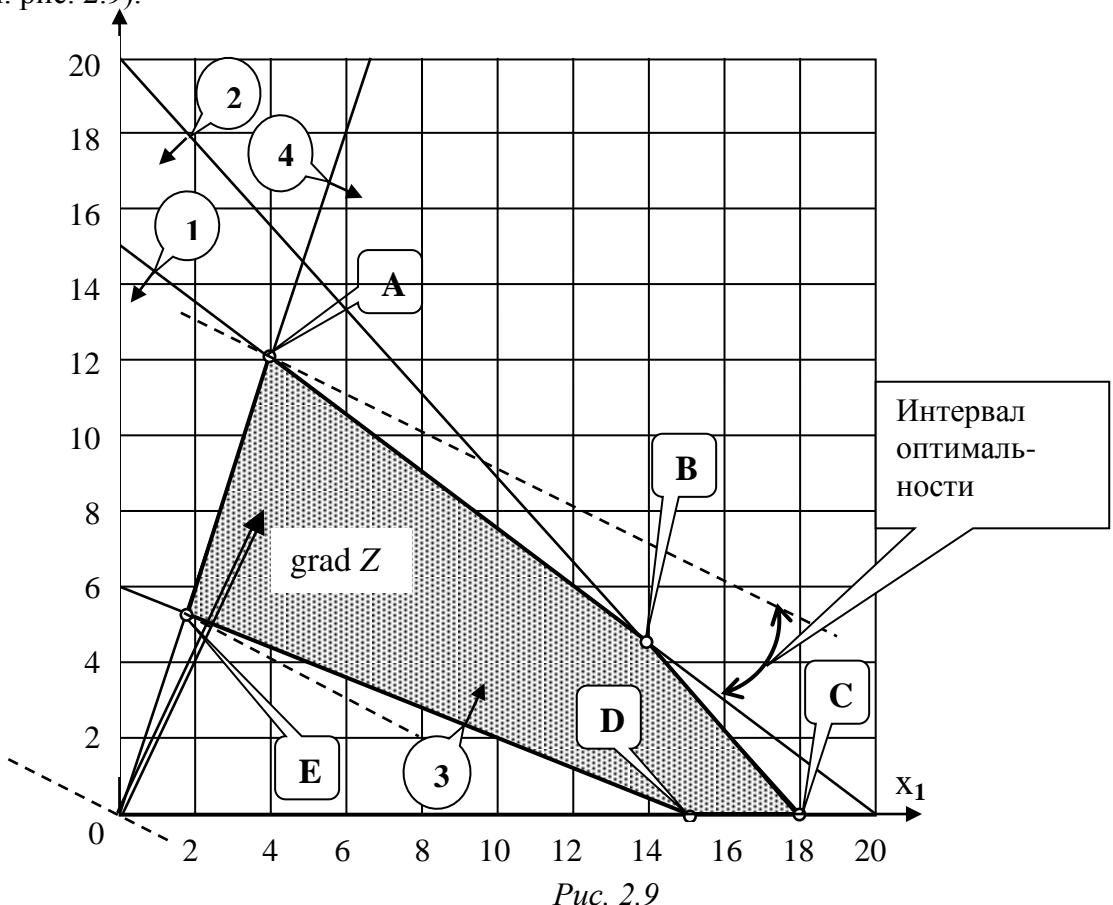


Рис. 2.9

После построения вектора  $\text{grad } Z = (4, 8)$ , перемещая перпендикулярную к нему линию фиксированных значений функционала, найдем, что искомые экстремумы достигаются в точках:

$$A(4; 12), \quad Z_{\max} = 120 \quad \text{и} \quad E(1,8; 5,3), \quad Z_{\min} = 49,4.$$

Очевидно, что эти точки останутся экстремальными и в некотором диапазоне изменения вектора градиента. Изменение модуля этого вектора влечет только пропорциональное изменение величин функционала и не влияет на координаты точек экстремума. Поэтому задача определения чувствительности решения по отношению к коэффициентам целевой функции сводится к определению интервала изменения отношения  $c_1/c_2$ , для которого остается неизменной рассматриваемая точка экстремума.

Пусть  $c_1 \geq 0$  и  $c_2 > 0$ , тогда любое увеличение  $c_2$  (уменьшение  $c_1$ ) приведет только к повороту вектора градиента против часовой стрелки, что не влияет на положение точки максимума. Однако как только этот вектор станет перпендикулярен прямой  $DE$  (третье ограничение), а это произойдет, если  $c_1/c_2 = 2/5$ , все точки этого отрезка будут обеспечивать минимальные значения функционалу. Дальнейшее увеличение  $c_2$  повлечет перемещение точки минимума в точку  $D$ .

Границное значение отношения  $c_1/c_2$  легко находится, если вспомнить, что вектор  $\vec{n} = (a_1, a_2)$  – это перпендикуляр к прямой  $a_1x_1 + a_2x_2 = b$ . Уравнение прямой  $DE$  – это  $2x_1 + 5x_2 = 30$ , а  $\vec{n}_3 = (2, 5)$  – это вектор ее нормали. На рисунке показан интервал оптимальности и для точки максимума при возрастании коэффициента  $c_1$ . Максимум переместится в точку  $B$  как только  $c_1/c_2 > 3/4$ , а при дальнейшем увеличении этого отношения даже в точку  $C$ .

Аналогичный подход можно применить и по отношению ко всем остальным коэффициентам модели. Изменение правых частей ограничений приведет параллельному перемещению соответствующих прямых, изменение коэффициентов при переменных – к изменению углов их наклона. Если прямая определяет точку экстремума, то можно определить зависимость приращения координат этой точки от таких изменений. Например, при уменьшении правой части первого ограничения, эта точка будет плавно перемещаться по отрезку  $AE$ .

Такой анализ устойчивости решения важен в реальных экономических задачах, когда необходимо учитывать и погрешности в исходных данных и возможные изменения ситуации. Поэтому современные компьютерные программы предусматривают получение отчетов по устойчивости решения, если только это возможно. Проведенный выше анализ иллюстрирует идею такого подхода.

## 2.7. Теорема об экстремальном значении целевой функции на выпуклом множестве

Графический способ решения задачи линейного программирования позволяет установить, что оптимальное решение достигается всегда в угловых точках пространства решений. На этой идеи и основан алгебраический симплекс-метод решения любой задачи линейного программирования.

**Теорема.** *Если задача линейного программирования имеет решение, то целевая функция достигает экстремального значения в одной из угловых точек многогранника допустимых планов. Если же целевая функция достигает экстремального значения более чем в одной угловой точке, то она принимает это же значение в любой точке, являющейся выпуклой линейной комбинацией этих условных точек.*

Это основная теорема линейного программирования. Для двух переменных она уже обоснована, а для случая  $n$  переменных ее надо доказать. Сначала уточним использованные термины.

**Угловая точка.** Точка выпуклого множества называется угловой, если ее нельзя представить в виде выпуклой линейной комбинации двух различных точек этого множества.

Выпуклой линейной комбинацией двух точек  $\vec{x}_1$  и  $\vec{x}_2$  называют точку  $\vec{x} = \lambda\vec{x}_1 + (1-\lambda)\vec{x}_2$ , где  $0 \leq \lambda \leq 1$ . Геометрически это отрезок, соединяющий точки  $\vec{x}_1$  и  $\vec{x}_2$ , которые получатся соответственно при значениях  $\lambda=1$  и  $\lambda=0$ . Остальные точки отрезка получатся при промежуточных значениях  $\lambda$ . В частности, середине отрезка соответствует значение  $\lambda=0,5$ .

В общем случае для  $n$  точек  $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$  выпуклой линейной комбинацией называется точка

$$\vec{x} = \lambda_1\vec{x}_1 + \lambda_2\vec{x}_2 + \dots + \lambda_n\vec{x}_n, \text{ где } \lambda_i \geq 0, \quad i = \overline{1, n} \text{ и } \sum_{i=1}^n \lambda_i = 1.$$

Если такие точки являются вершинами выпуклого многоугольника на плоскости, то любую его точку, как внутреннюю, так и граничную, можно представить в виде их выпуклой линейной комбинации.

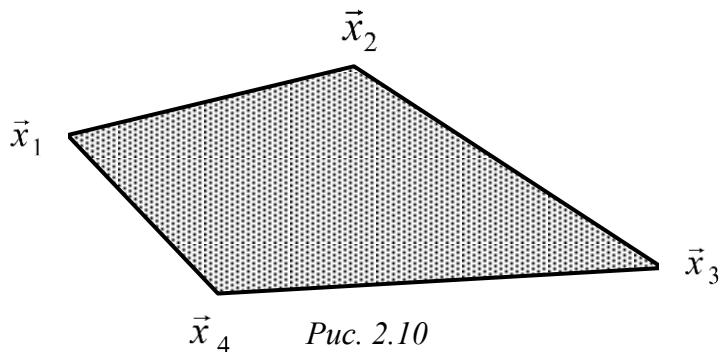


Рис. 2.10

При этом для всех точек границы кроме вершин потребуется использовать точно по две различные точки. Для любой внутренней точки достаточно трех. Вершины в виде выпуклой линейной комбинации различных точек представить нельзя и, согласно определению, они являются угловыми точками.

Можно доказать и в общем случае, что множество допустимых решений задачи линейного программирования, если оно не пусто, является выпуклым. Действительно, в  $n$ -мерном пространстве каждое неравенство вида  $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i$  исключает

из числа допустимых решений некоторое полупространство. При любом числе таких ограничений область допустимых решений останется выпуклой. Представим себе для наглядности обработку алмаза на плоскошлифовальном станке. Полученный бриллиант всегда будет иметь форму выпуклого многогранника.

Без ограничения общности изложения в дальнейшем будем предполагать, что область допустимых решений является ограниченной. Всегда можно предполагать, что сумма всех переменных реальной задачи не превышает некоторое достаточно большое положительное число.

Докажем теперь сформулированную выше теорему, рассуждая от противного. Пусть  $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_k$  – все угловые точки многогранника допустимых решений, а экстремум целевой функции достигается в некоторой внутренней точке многогранника  $\vec{x}^*$ , например,  $\max Z = Z(\vec{x}^*)$ . Но любую внутреннюю точку выпуклого многогранника можно представить как выпуклую линейную комбинацию угловых точек, т. е.

$$\vec{x}^* = \lambda_1 \vec{x}_1 + \lambda_2 \vec{x}_2 + \dots + \lambda_k \vec{x}_k, \text{ где } \lambda_i \geq 0, (i = \overline{1, k}) \text{ и } \sum_{i=1}^k \lambda_i = 1.$$

Следовательно,  $Z(\vec{x}^*) = \lambda_1 Z(\vec{x}_1) + \lambda_2 Z(\vec{x}_2) + \dots + \lambda_k Z(\vec{x}_k)$ .

Обозначим  $M = \max(Z(\vec{x}_1), Z(\vec{x}_2), \dots, Z(\vec{x}_k))$ , тогда

$$Z(\vec{x}^*) \leq \lambda_1 M + \lambda_2 M + \dots + \lambda_k M. \text{ И получается, что } Z(\vec{x}^*) \leq M.$$

Поскольку  $M$  – значение функционала в одной из угловых точек, то утверждение теоремы, что экстремум достигается в одной из угловых точек многогранника допустимых планов доказано. Остается доказать, что если целевая функция достигает экстремального значения более чем в одной угловой точке, то она принимает это же значение в любой точке, являющейся их выпуклой линейной комбинацией.

Пусть в угловых точках  $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_p$  целевая функция принимает свое максимальное значение равное  $M$ . Любая точка  $\vec{x}$ , являющаяся их выпуклой линейной комбинацией, может быть представлена как

$$\vec{x} = \lambda_1 \vec{x}_1 + \lambda_2 \vec{x}_2 + \dots + \lambda_p \vec{x}_p, \text{ где } \lambda_i \geq 0, i = (\overline{1, p}) \text{ и } \sum_{i=1}^p \lambda_i = 1.$$

Подставляя в целевую функцию, получим:

$$Z(\vec{x}) = \lambda_1 M + \lambda_2 M + \dots + \lambda_p M.$$

Значит  $Z(\vec{x}) = M$ , что и требовалось доказать.

## 2.8. Общая идея симплексного метода

Основная теорема линейного программирования может навести на мысль, что можно найти все угловые точки многогранника допустимых планов и сравнить в них значения целевой функции. Пусть задача записана в канонической форме и содержит  $m$  равенств с  $n$  переменными ( $m < n$ ),

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j x_j, \\ \sum_{j=1}^n a_{ij} x_j = b_i \quad & (i = \overline{1, m}), \\ x_j \geq 0 \quad & (j = \overline{1, n}). \end{aligned}$$

Без ограничения общности можно предполагать, что все уравнения линейно независимы, иначе некоторые можно было бы опустить, и что ранг матрицы коэффициентов системы равен  $m$ , в противном случае система несовместна. Произвольно выбирая  $m$  переменных из их общего числа  $n$ , положим оставшиеся  $n - m$  переменных равными нулю. Общее количество полученных таким путем миноров и соответственно групп уравнений – это число сочетаний из  $n$  элементов по  $m$ , которое равно

$$C_n^m = \frac{n!}{m!(n-m)!}.$$

Среди решений этих групп уравнений содержатся все угловые точки множества допустимых планов, но количество таких групп даже для относительно простых задач настолько велико, что поиск оптимального решения простым перебором вершин не имеет смысла. Даже для рассмотренного примера графического решения задачи  $C_n^2 = 15$ . (Кроме четырех ограничений надо учитывать и две координатные оси, поэтому  $n = 6$ ). Попарные пересечения этих прямых дадут все вершины многоугольника области допустимых решений, в том числе и точки экстремума. Часть полученных точек окажется вне области допустимых решений. Видно, что затраты труда при таком подходе сильно увеличиваются. В реальных задачах число переменных измеряется сотнями и тысячами, поэтому разработаны методы рационального перебора угловых точек. К наиболее простым из них относится симплексный метод, в основу которого положена идея последовательного улучшения плана. Если в сложной ситуации не удается определить наилучший путь к цели, то логично сделать шаг, приближающий к этой цели. Чтобы реализовать эту идею сначала введем необходимые термины.

**Базисное решение (план).** Так называют решение одной из выделенных групп уравнений, если для этой группы оно является единственным. Соответствующие  $m$  переменных называют базисными, а  $n - m$  переменных, которым были присвоены нулевые значения, небазисными, или свободными.

**Опорное базисное решение (опорный план).** Так называют базисное решение, если все  $m$  базисных переменных неотрицательны.

**Недопустимое базисное решение.** Имеет место если хоть одна базисная переменная отрицательна.

**Вырожденное базисное решение.** Имеет место тогда, когда хоть одна базисная переменная нулевая.

Поскольку ранг матрицы коэффициентов  $m$  и вектор правых частей ненулевой, то среди ее миноров  $m$ -го порядка обязательно будут ненулевые. Значит, базисные решения

всегда существуют. Множество базисных решений содержит в своем составе, как подмножество, все угловые точки области допустимых планов.

Алгоритм симплекс-метода нахождения оптимального плана всегда начинается с некоторого допустимого базисного решения. Далее следует проверка, можно ли улучшить значение целевой функции, если увеличить одну из небазисных (нулевых) переменных, ввести ее в базис. Если такой переменной нет, то оптимальное решение найдено. Если есть, то необходимо перейти к новому, лучшему (или хотя бы не худшему) базисному решению. Но базисное решение содержит точно  $m$  переменных, поэтому после определения переменной, вводимой в базис, необходимо определить исключаемую из базиса переменную.

Каждый такой цикл преобразования называют итерациями, а методы вычислений, основанные на последовательных итерациях, – итерационными методами.

Многократное повторение однотипных вычислений весьма трудоемко и утомительно, если выполнять его вручную, но если его запрограммировать, то при современной вычислительной технике размерность решаемых задач обычно определяется содержательными проблемами их экономической постановки, а не техническими трудностями расчета.

## **2.9. Построение опорного плана**

Рассмотрим сначала частный случай ограничений типа «меньше или равно», использованных при построении модели оптимального планирования производства. После приведения к каноническому виду система ограничений будет:

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n + x_{n+1} = b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n + x_{n+2} = b_2; \\ \vdots \quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n + x_{n+m} = b_m; \\ x_j \geq 0 \quad j = (\overline{1, n+m}). \end{array} \right\}$$

Про такую систему говорят, что она имеет *предпочтительный вид*, поскольку выполняются два условия:

- Правые части ограничений-равенств неотрицательны  $b_i \geq 0$ ,  $i = \overline{1, m}$  (для рассматриваемой задачи это объемы имеющихся ресурсов).
  - Каждое ограничение содержит предпочтительную переменную, входящую в него с коэффициентом, равным единице, и не входящую в остальные ограничения (для приведенного примера это  $x_{n+k}$ , где  $k = \overline{1, m}$ ).

Для такой системы ограничений всегда можно найти опорное решение, если принять предпочтительные переменные за базисные, а остальные считать свободными и приравнять нулю. Тогда базисные переменные равны правым частям ограничений-равенств и опорный план будет:

$$\mathbf{x}_0 = (\underbrace{0, 0, \dots, 0}_n, \underbrace{b_1, b_2, \dots, b_m}_m).$$

Экономический смысл такого плана очевиден: производство отсутствует, все ресурсы зарезервированы. Он будет оптимален только в тривиальном случае, когда все виды продукции являются убыточными.

В общем случае любую задачу линейного программирования можно преобразовать к каноническому виду:

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j x_j, \\ \text{subject to } \quad & \sum_{j=1}^n a_{ij} x_j = b_i \quad (i = \overline{1, m}), \\ & x_j \geq 0 \quad (j = \overline{1, n}). \end{aligned}$$

Без ограничения общности можно считать, что правые части ограничений-равенств неотрицательны, иначе соответствующие ограничения было бы достаточно умножить на  $-1$ , но предполагать предпочтительный вид даже для отдельных ограничений оснований нет.

Наиболее общим способом построения опорного плана является использование искусственных переменных  $\omega_i$ , которые добавляются ко всем левым частям ограничений-равенств, не имеющих предпочтительного вида. В частности, для модели формирования минимальной продовольственной корзины получим:

Если эти искусственные переменные приравнять правым частям ограничений, то при таком опорном плане затраты на приобретение продуктов минимальны (равны нулю), что было бы хорошо, но вот только корзина будет пустая, а это уже неприемлемо.

Чтобы в последующих итерациях освободиться от искусственных переменных, введем для каждой из них достаточно большой штраф в выражении для целевой функции. Если решается задача на максимум, то переменной  $\omega_i$  в функционале должно соответствовать слагаемое  $-M\omega_i$ , где  $M > 0$ , а при решении задачи на минимум выражение  $+M\omega_i$ .

Такой метод называют *M-методом*, или *методом больших штрафов*, а полученную задачу – *M-задачей*, соответствующей исходной. Естественно ожидать, что все искусственные переменные в процессе оптимизации получат нулевые значения. Однако, если исходная задача несовместна, то при любом штрафе исключить все искусственные переменные не удастся. Например, если задачу оптимального планирования производства дополнить условиями по минимальным объемам производства отдельных видов продукции, то при недостатке наличных ресурсов придется приобретать их по любой цене. Зато сразу видно, что задача не имеет допустимого решения и более того видно, сколько и каких именно ресурсов не хватает.

Остается открытым вопрос, как понимать термин «достаточно большой». Теоретически надо  $M \rightarrow \infty$ . При аналитических вычислениях пишут  $M$ , и при обнулении искусственных переменных соответствующее выражение исчезает.

При компьютерных вычислениях всегда имеют место ошибки округления, особенно, когда в операциях сложения или вычитания одновременно участвуют и большие, и малые числа. В этом легко убедиться экспериментом, вычисляя, например, значение  $10^{15} + 1$ . Скорее всего, единица просто не будет прибавлена. Если это не так, то достаточно увеличить первое или уменьшить второе число. Поэтому лучше не следовать бездумно теоретическому требованию о «большом значении»  $M$ , а назначать его исходя из «разумной достаточности». Если же в решение необоснованно войдет искусственная переменная, то соответствующий ей штраф можно увеличить.

В реальных компьютерных программах, реализующих симплекс-метод, для подавления ошибок округления используется двухэтапный метод. На первом этапе решается  $M$ -задача, но за критерий принимается просто сумма значений искусственных переменных. Если достигается нулевое значение этой суммы, то переходят ко второму этапу, принимая полученное оптимальное решение за опорный план исходной задачи. Если же на первом этапе целевая функция не обнуляется, следовательно, исходная система ограничений несовместна.

## 2.10. Признак оптимальности опорного плана

Если начальный опорный план найден, то далее следует проверить, можно ли улучшить значение целевой функции, если увеличить одну из небазисных (нулевых) переменных и ввести ее в базис.

Пусть задача линейного программирования преобразована к канонической форме и ее ограничения имеют предпочтительный вид:

$$\begin{aligned} \max Z &= \sum_{j=1}^n c_j x_j, \\ x_i + \sum_{j=m+1}^n a_{ij} x_j &= b_i, \quad b_i \geq 0 \quad (i = \overline{1, m}), \\ x_j &\geq 0 \quad (j = \overline{1, n}). \end{aligned}$$

Выражая базисные переменные через свободные, получим

$$x_i = b_i - \sum_{j=m+1}^n a_{ij} x_j \quad (i = \overline{1, m})$$

и после подстановки в целевую функцию выразим ее значение только через небазисные переменные, как

$$\begin{aligned} Z &= \sum_n^m c_i b_i + \left( c_{m+1} - \sum_{i=1}^m c_i a_{i,m+1} \right) x_{m+1} + \left( c_{m+2} - \sum_{i=1}^m c_i a_{i,m+2} \right) x_{m+2} + \dots + \\ &\quad + \left( c_n - \sum_{i=1}^m c_i a_{i,n} \right) x_n. \end{aligned}$$

Используем векторные обозначения для столбцов матрицы коэффициентов и столбца свободных членов

$$\mathbf{A}_{m+1} = \begin{bmatrix} a_{1,m+1} \\ a_{2,m+1} \\ \dots \\ a_{m,m+1} \end{bmatrix}, \quad \mathbf{A}_{m+2} = \begin{bmatrix} a_{1,m+2} \\ a_{2,m+2} \\ \dots \\ a_{m,m+2} \end{bmatrix}, \dots, \quad \mathbf{A}_n = \begin{bmatrix} a_{1n} \\ a_{2n} \\ \dots \\ a_{nn} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix};$$

а также для коэффициентов целевой функции при базисных переменных

$$\mathbf{c}_B = (c_1 \ c_2 \ \dots \ c_m).$$

Тогда целевая функция может быть представлена как

$$Z = \mathbf{c}_B \mathbf{b} + (c_{m+1} - \mathbf{c}_B \mathbf{A}_{m+1}) x_{m+1} + (c_{m+2} - \mathbf{c}_B \mathbf{A}_{m+2}) x_{m+2} + \dots + (c_n - \mathbf{c}_B \mathbf{A}_n) x_n,$$

а если ввести обозначения для входящих в нее слагаемых

$$\Delta_0 = \mathbf{c}_B \mathbf{b} \text{ и } \Delta_j = \mathbf{c}_B \mathbf{A}_j - c_j,$$

то окончательно получим  $Z = \Delta_0 - \sum_{j=m+1}^n \Delta_j x_j$ .

Переменные  $x_j$  ( $j = \overline{m+1, n}$ ) – это свободные переменные, поэтому величины  $\Delta_j$  называют оценками свободных переменных. Если все эти оценки неотрицательны, то опорный план  $(b_1, b_2, \dots, b_m, \underbrace{0, 0, \dots, 0}_{n-m})$  является оптимальным, так как введение любой из

этих переменных в базис только уменьшит величину функционала. Его значение для текущего опорного плана есть  $\Delta_0$  и определяется базисными переменными.

Если для нескольких свободных переменных оценки отрицательны, то включение любой из них в базис приведет к увеличению функционала. Обычно для очередной итерации вводят в базис ту свободную переменную, которой соответствует наибольшая по абсолютной величине отрицательная оценка. Однако вполне может случиться, что такая переменная требует значительных затрат дефицитных ресурсов и при последующих итерациях ее место займет переменная, не столь требовательная к ресурсам и обеспечивающая за счет своей величины более значительный прирост функционала, несмотря на меньшую оценку.

На практике симплекс-метод применяется для решения очень больших задач, а приведенные выше соображения поясняют, почему количество необходимых итераций зависит не только от размерности модели, но и от ее структуры. Довольно часто число итераций для нахождения оптимального решения оказывается примерно таким же, как и число ограничений модели, но можно искусственно построить задачу, содержащую  $n$  переменных, при решении которой потребуется  $2n$  итераций.

Если все оценки свободных переменных неотрицательны, но хотя бы одна из них нулевая, то при включении этой переменной в базис функционал не изменится и мы получим альтернативное оптимальное решение. Это будет еще одна угловая точка, и получается бесконечное множество решений, которым соответствует отрезок, соединяющий найденные угловые точки. Для интерпретации возникшей проблемы выбора одного из множества решений предположим, что одинаковые результаты получаются при производстве любого из двух видов товаров. Скорее всего, будет принято решение производить оба товара, чтобы в условиях конкуренции «не класть все яйца в одну корзину».

## 2.11. Переход к нехудшему опорному плану

Представим все данные рассматриваемой задачи в виде таблицы, которую называют симплексной. Это естественный способ организации данных и на бумаге, и в памяти компьютера. В первой строке табл. 2.2 записаны все переменные, входящие в модель, а под ними – соответствующие им коэффициенты целевой функции. Очевидно, что при переходе к нехудшему опорному плану эти строки изменяться не должны. Первоначальный порядок перечисления переменных модели может быть любым.

В столбце  $B\bar{P}$  перечислены все переменные, входящие в текущий базис, в столбцах  $\mathbf{c}_B$  и  $\mathbf{b}$  – коэффициенты функционала при базисных переменных и сами значения этих переменных.

Таблица 2.2

БП	$\mathbf{c}_B$	$\mathbf{b}$	$x_1$	$x_2$	...	$x_p$	...	$x_m$	$x_{m+1}$	...	$x_{\bar{P}}$	...	$x_n$
			$c_1$	$c_2$	...	$c_p$	...	$c_m$	$c_{m+1}$	...	$c_{\bar{P}}$	...	$c_n$
$x_1$	$c_1$	$b_1$	1	0	...	0	...	0	$a_{1,m+1}$	...	$a_{1,\bar{P}}$	...	$a_{1,n}$
$x_2$	$c_2$	$b_2$	0	1	...	0	...	0	$a_{2,m+1}$	...	$a_{2,\bar{P}}$	...	$a_{2,n}$
...	...	...	...	...	...	...	...	...	...	...	...	...	...
$x_P$	$c_P$	$b_P$	0	0	...	1	...	0	$a_{P,m+1}$	...	$a_{P,\bar{P}}$	...	$a_{P,n}$
...	...	...	...	...	...	...	...	...	...	...	...	...	...
$x_m$	$c_m$	$b_m$	0	0	...	0	...	1	$a_{m,m+1}$	...	$a_{m,\bar{P}}$	...	$a_{m,n}$
Оценки	$\Delta_0$		0	0	...	0	...	0	$\Delta_{m+1}$	...	$\Delta_{\bar{P}}$	...	$\Delta_n$

Основная часть таблицы содержит все коэффициенты матрицы модели, а последняя строка – текущее значение функционала  $\Delta_0 = \mathbf{c}_B \mathbf{b}$  и оценки свободных переменных  $\Delta_j = \mathbf{c}_B \mathbf{A}_j - c_j$ , ( $j = \overline{m+1, n}$ ). При выполнении вычислений эти значения удобно находить, попарно перемножая соответствующие столбцы таблицы, суммируя результаты и отнимая записанные во второй строке коэффициенты функционала. Для базисных переменных можно сразу записать нулевые значения, которые всегда получаются при использовании формул для оценок.

Предположим, что переменная  $x_{\bar{P}}$  имеет наибольшую по абсолютной величине отрицательную оценку. Такую переменную называют перспективной, поскольку введение ее в базис обеспечивает переход к нехудшему опорному плану. Прирост целевой функции будет пропорционален величине значения этой переменной в новом опорном плане, но ни одно из ограничений-равенств, коэффициенты которых записаны в строках симплексной таблицы, нарушать нельзя.

Значения свободных переменных в текущем опорном плане равны нулю, поэтому каждому ограничению соответствуют равенства

$$x_i + a_{i,\bar{P}} x_{\bar{P}} = b_i, \quad x_i \geq 0, \quad b_i \geq 0, \quad (i = \overline{1, m}).$$

Перспективная переменная  $x_{\Pi}$  тоже равна нулю, но при введении ее в базис ей нельзя придать значение большее, чем  $\min_{\substack{i=1,m \\ a_{i,\Pi} > 0}} \left\{ \frac{b_i}{a_{i,\Pi}} \right\}$ , которое и называется

наименьшим симплексным отношением. Условие  $a_{i,\Pi} > 0$  означает, что для строк, в которых коэффициенты  $a_{i,\Pi} \leq 0$ , симплексные отношения вычислять излишне, так как они не ограничивают рост переменной  $x_{\Pi}$ .

Пусть наименьшее симплексное отношение есть  $\frac{b_P}{a_{P,\Pi}}$  и, следовательно,

переменная  $x_P$  должна получить нулевое значение и быть исключена из базиса. Эту строку называют разрешающей, а коэффициент  $a_{P,\Pi}$  – разрешающим элементом. Именно разрешающая строка «является самым слабым звеном цепи» и симплексное отношение для нее и есть максимально возможное значение перспективной переменной в новой итерации. Если минимальное значение симплексного отношения достигается более чем в одной строке, то в качестве разрешающей можно выбрать любую из них. Если предположить, что минимальное симплексное отношение не определяется, так как  $a_{i,\Pi} \leq 0$  ( $i = \overline{1, m}$ ), то перспективная переменная и функционал могут неограниченно возрастать, значит, надо искать ошибку, которая к этому привела.

## 2.12. Симплексные преобразования

Если перспективная переменная  $x_{\Pi}$  должна заменить в базисе переменную  $x_p$ , то необходимо найти ее значение из уравнения

$$x_p + a_{p,m+1}x_{m+1} + \dots + a_{p,\Pi}x_{\Pi} + \dots + a_{p,n}x_n = b_p.$$

Получаем, что

$$x_{\Pi} = \frac{b_p}{a_{p,\Pi}} - \left( \frac{1}{a_{p,\Pi}}x_p + \frac{a_{p,m+1}}{a_{p,\Pi}}x_{m+1} + \dots + \frac{a_{p,\Pi-1}}{a_{p,\Pi}}x_{\Pi-1} + \frac{a_{p,\Pi+1}}{a_{p,\Pi}}x_{\Pi+1} + \dots + \frac{a_{p,n}}{a_{p,\Pi}}x_n \right)$$

и подставляем это выражение в остальные уравнения исходной системы ограничений. Так, вместо уравнения

$$x_i + a_{i,m+1}x_{m+1} + \dots + a_{i,\Pi}x_{\Pi} + \dots + a_{i,n}x_n = b_i,$$

получим:

$$\begin{aligned} & x_i + a_{i,m+1} + \dots + a_{i,\Pi} \frac{b_p}{a_{p,\Pi}} - \\ & - a_{i,\Pi} \left( \frac{1}{a_{p,\Pi}}x_p + \frac{a_{p,m+1}}{a_{p,\Pi}}x_{m+1} + \dots + \frac{a_{p,\Pi-1}}{a_{p,\Pi}}x_{\Pi-1} + \frac{a_{p,\Pi+1}}{a_{p,\Pi}}x_{\Pi+1} + \dots + \frac{a_{p,n}}{a_{p,\Pi}}x_n \right) + \\ & + a_{i,n}x_n = b_i, \end{aligned}$$

откуда

$$\begin{aligned} & x_i = \frac{b_i a_{p,\Pi} - b_p a_{i,\Pi}}{a_{p,\Pi}} - \\ & - \left( \frac{a_{i,m+1}a_{p,\Pi} - a_{p,m+1}a_{i,\Pi}}{a_{p,\Pi}}x_{m+1} + \dots + \frac{a_{i,\Pi}}{a_{p,\Pi}}x_p + \dots + \frac{a_{i,n}a_{p,\Pi} - a_{p,n}a_{i,\Pi}}{a_{p,\Pi}}x_n \right). \end{aligned}$$

Аналогично, при подстановке в выражение для целевой функции

$$Z = \Delta_0 - (\Delta_{m+1}x_{m+1} + \dots + \Delta_{\Pi}x_{\Pi} + \dots + \Delta_nx_n)$$

получим:

$$\begin{aligned}
Z = & \Delta_0 - \Delta_{m+1}x_{m+1} - \dots - \Delta_\Pi \frac{b_p}{a_{p,\Pi}} + \\
& + \Delta_\Pi \left( \frac{1}{a_{p,\Pi}} x_p + \frac{a_{p,m+1}}{a_{p,\Pi}} x_{m+1} + \dots + \frac{a_{p,\Pi-1}}{a_{p,\Pi}} x_{\Pi-1} + \frac{a_{p,\Pi+1}}{a_{p,\Pi}} x_{\Pi+1} + \dots + \frac{a_{p,n}}{a_{p,\Pi}} x_n \right) - \dots \text{ и} \\
& - \Delta_n x_n \\
& \text{окончательно} \\
Z = & \frac{\Delta_0 a_{p,\Pi} - \Delta_\Pi b_p}{a_{p,\Pi}} - \\
& - \left( \frac{\Delta_{m+1} a_{p,\Pi} - \Delta_\Pi a_{p,m+1}}{a_{p,\Pi}} x_{m+1} + \dots + \frac{\Delta_\Pi}{a_{p,\Pi}} x_p + \dots + \frac{\Delta_n a_{p,\Pi} - \Delta_\Pi a_{p,n}}{a_{p,\Pi}} x_n \right).
\end{aligned}$$

Приведенные формулы полностью описывают переход к новому базису задачи линейного программирования, который и называется *симплексным преобразованием*. Для его выполнения не требуется их запоминать, так как они эквивалентны следующим правилам, которые следуют из сопоставления этих формул и симплексной таблицы.

1. Элементы разрешающей строки  $i = p$  новой таблицы получаются делением старых элементов этой строки на разрешающий элемент:

$$b_p^* = \frac{b_p}{a_{p,\Pi}}, \quad a_{p,j}^* = \frac{a_{p,j}}{a_{p,\Pi}} \quad (j = 1, n).$$

2. Элементы остальных строк разрешающего столбца  $j = \Pi$  обнуляются:

$$a_{i,\Pi}^* = 0 \quad (i \neq p), \quad a_{p,\Pi}^* = 1.$$

3. Все остальные элементы новой симплексной таблицы, включая значения базисных переменных, оценки свободных переменных и значение функционала, вычисляются по правилу прямоугольника. Для этого в исходной таблице выделим такой прямоугольник, чтобы одна из его диагоналей определялась вычисляемым и разрешающим элементами. Эту диагональ называют *главной*, а другую – *побочной*.

Чтобы получить элемент новой симплексной таблицы нужно из произведения угловых элементов главной диагонали вычесть произведение угловых элементов побочной диагонали и разделить полученное число на разрешающий элемент. Это правило отображено на рис. 2.11.

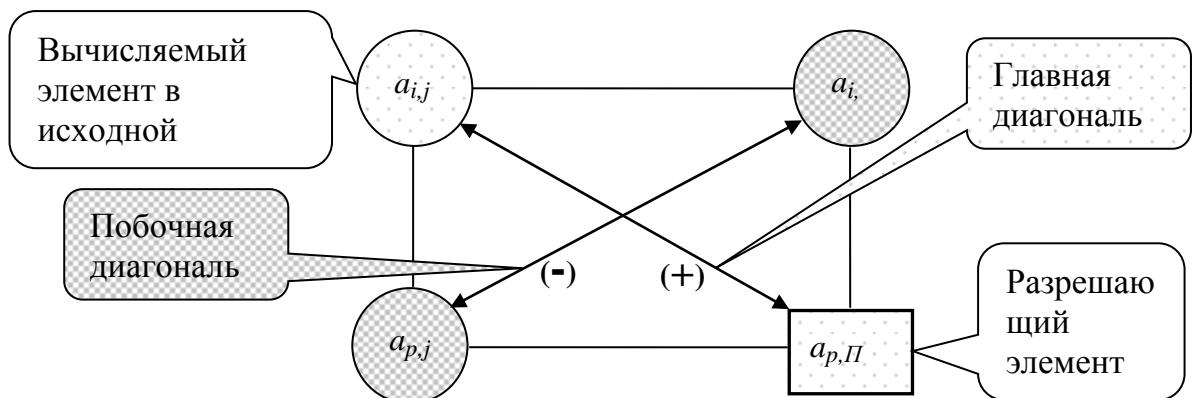


Рис. 2.11

Соответствующие формулы для новой симплексной таблицы получаются из вышеприведенных соотношений и имеют вид:

$$\dot{a_{i,j}} = \frac{a_{i,j}a_{p,\Pi} - a_{p,j}a_{i,\Pi}}{a_{p,\Pi}}, \quad \dot{b_i} = \frac{b_i a_{p,\Pi} - b_p a_{i,\Pi}}{a_{p,\Pi}}, \quad (i \neq p; j \neq \Pi),$$

$$\dot{\Delta_j} = \frac{\Delta_j a_{p,\Pi} - \Delta_\Pi a_{p,j}}{a_{p,\Pi}}, \quad \dot{\Delta_0} = \frac{\Delta_0 a_{p,\Pi} - \Delta_\Pi b_p}{a_{p,\Pi}}, \quad (j = \overline{1, n}).$$

Эти выражения легко преобразовать, тогда получаемый порядок вычислений называют правилом треугольника. Например, для нового значения функционала получим:

$\dot{\Delta_0} = \Delta_0 - \frac{b_p}{a_{p,\Pi}} \Delta_\Pi$ . Но величина  $\frac{b_p}{a_{p,\Pi}}$  – это наименьшее симплексное отношение, равное значению переменной  $x_\Pi$  в новой симплексной таблице, следовательно  $\dot{\Delta_0} = \Delta_0 - x_\Pi \Delta_\Pi$ .

Таким образом, прирост функционала на каждую единицу прироста переменной  $x_\Pi$  (до вхождения в базис она была свободной и имела нулевое значение) равен абсолютному значению оценки этой переменной  $\Delta_\Pi$ . Эта оценка всегда отрицательна, иначе переменная  $x_\Pi$  не была бы выбрана как перспективная.

## 2.13. Организация и контроль вычислений

Рассмотрим организацию вычислений на простейшем примере, имея в виду, что при увеличении объема вычислений следует использовать компьютер. Пусть фирма может выпускать 3 вида продукции  $\Pi_j$ , где  $(j = \overline{1,3})$ . При ее изготовлении используются три вида ресурсов (труд, сырье и оборудование), расход которых  $b_i$ , где  $(i = \overline{1,3})$ , ограничен величинами  $b_1 = 180$ ,  $b_2 = 60$  и  $b_3 = 120$ . Затраты ресурса  $i$ -го вида на единицу продукции  $j$ -го вида составляют  $a_{ij}$  единиц. Прибыль от производства единицы продукции  $j$ -го вида составляет  $c_j$  единиц. Значения параметров приведены в табл. 2.3.

Математическая модель задачи оптимизации производственного плана по величине прибыли примет вид

$$\begin{aligned} \max Z &= 9x_1 + 6x_2 + 7x_3; \\ 6x_1 + 2x_2 + 5x_3 &\leq 180; \\ 2x_2 + 4x_3 &\leq 60; \\ 5x_1 + 2x_2 + 5x_3 &\leq 120; \\ x_j &\geq 0 \quad (j = \overline{1,3}). \end{aligned} \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\}$$

Таблица 2.3

Вид ресурса, $i$	Расход $i$ -го ресурса на единицу $j$ -ой продукции, $a_{ij}$			Запас ресурса, $b_i$
	$\Pi_1$	$\Pi_2$	$\Pi_3$	
Труд	6	2	5	180
Сырье		2	4	60
Оборудование	5	2	5	120
Прибыль, $c_j$	9	6	7	

Преобразовывая модель к канонической форме и предпочтительному виду, получим:

$$\begin{aligned} \max Z &= 9x_1 + 6x_2 + 7x_3 + 0x_4 + 0x_5 + 0x_6; \\ 6x_1 + 2x_2 + 5x_3 + x_4 &= 180; \\ 2x_2 + 4x_3 + x_5 &= 60; \\ 5x_1 + 2x_2 + 5x_3 + x_6 &= 120; \\ x_j &\geq 0 \quad (j = \overline{1,6}). \end{aligned} \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\}$$

Занесем условия задачи в симплексную таблицу (табл. 2.3). Значения базисных переменных  $x_4, x_5$  и  $x_6$  будут равны правым частям системы ограничений. Значение целевой функции для опорного плана вычисляется как скалярное произведение  $\Delta_0 = \mathbf{c}_B \mathbf{b}$ , где  $\mathbf{c}_B$  – вектор коэффициентов целевой функции при базисных переменных, а  $\mathbf{b}$  – вектор значений базисных переменных. Оценки свободных переменных вычисляются по формуле  $\Delta_j = \mathbf{c}_B \mathbf{A}_j - c_j$  ( $j = \overline{1,3}$ ), где  $\mathbf{A}_j$  вектор – столбец коэффициентов при переменной  $x_j$ .

Таблица 2.4

Номер итерации	$BП$	$c_B$	$b$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	Симплексные отношения
				9	6	7	0	0	0	
0	$x_4$	0	180	6	2	5	1	0	0	$180/6 = 30$
	$x_5$	0	60	0	2	4	0	1	0	—
	$x_6$	0	120	5	2	5	0	0	1	$120/5 = 24$
	Оценки		$\Delta_0$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$	
1			0	-9	-6	-7	0	0	0	
	$x_4$	0	36	0	-2/5	-1	1	0	-6/5	—
	$x_5$	0	60	0	2	4	0	1	0	$60/2 = 30$
	$x_1$	9	24	1	2/5	1	0	0	1/5	$24/(2/5) = 60$
2	Оценки		$\Delta_0$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$	
			216	0	-2,4	2	0	0	1,8	
	$x_4$	0	48	1	0	-1/5	1	1/5	-6/5	—
	$x_2$	6	30	0	1	2	0	1/2	0	—
3	$x_1$	9	12	0	0	1/5	0	-1/5	1/5	—
	Оценки		$\Delta_0$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$	
			288	0	0	6,8	0	1,2	1,8	

Для ввода в базис (при нулевой итерации) выберем переменную с максимальной по абсолютной величине отрицательной оценкой. Это переменная  $x_1$ . Соответствующий ей столбец называют разрешающим. В таблице 2.4 он выделен затенением. Вычисленные симплексные отношения показывают максимально возможные значения для этой переменной, если учитывать только одно, соответствующее строке, ограничение. Для этой переменной второй вид ресурса не требуется  $a_{12} = 0$  и симплексное отношение во второй строке не вычисляется. Если бы получился отрицательный результат, то его также заменили бы прочерком.

Выберем в качестве разрешающей строки ту, для которой симплексное отношение минимально. В данном случае строка, соответствующая третьему ограничению, ограничивает максимальное значение переменной  $x_1$  и, следовательно, будет разрешающей. Она выделена затенением, а элемент  $a_{13}$ , называемый разрешающим, — еще и жирным шрифтом.

Перейдем к нехудшему опорному плану, вводя в базис переменную  $x_1$  (разрешающий столбец) вместо  $x_6$  (разрешающая строка). По правилам симплексного преобразования элементы разрешающей строки делятся на разрешающий элемент, который становится равным единице. Все остальные элементы разрешающего столбца для новой итерации равны нулю. Для продолжения пересчета элементов таблицы, включая значения базисных переменных и оценки, используется формула прямоугольников. Старое значение элемента умножаем на разрешающий элемент (первая диагональ) и отнимаем произведение элементов, образующих вторую диагональ прямоугольника. Результат делим на разрешающий элемент и записываем в таблицу для новой итерации. Например, новое значение переменной  $x_4$

вычисляется как  $(180 \times 5 - 120 \times 6)/5 = 180 - 24 \times 6 = 36$ , новое значение коэффициента  $a_{12}$  будет

$$\frac{5 \times 2 - 6 \times 2}{5} = -\frac{2}{5},$$

величина новой оценки  $\Delta_2$  для переменной  $x_2$  составит

$$\frac{5 \times (-6) - (-9) \times 2}{5} = \frac{-30 + 18}{5} = \frac{-12}{5} = -2,4.$$

Прежде чем переходить к следующей итерации, следует для контроля правильности расчетов, вычислить оценки, используя приведенные выше формулы для  $\Delta_0$  и  $\Delta_j$ ,  $j = \{1, 6\}$ . В этих формулах участвуют все используемые числовые значения, поэтому случайные ошибки при ручном расчете будут своевременно обнаружены. При программной реализации на этой основе можно построить алгоритмы для коррекции погрешностей, возникающих из-за ограниченной точности вычислений, которые могут накапливаться при большом числе итераций.

Для рассматриваемой задачи на первой итерации только переменная  $x_2$  имеет отрицательную оценку и, следовательно, должна быть включена в базис. Разрешающей строкой, как это видно из значений симплексных отношений, приведенных в таблице, будет вторая, поэтому переменную  $x_5$  исключаем из базиса. После выполнения второй итерации получаем, что все оценки переменных положительны, а, следовательно, оптимальный по прибыли план найден.

Это вектор  $\mathbf{x}^* = (x_1, x_2, x_3, x_4, x_5, x_6) = (12, 30, 0, 48, 0, 0)$ , которому соответствует прибыль  $Z^* = Z(\mathbf{x}^*) = 288$ . В этот оптимальный план вошла дополнительная переменная  $x_4 = 48$ , означающая, что такое количество единиц первого ресурса не используется в оптимальном плане. Переменные  $x_5$  и  $x_6$  равны нулю (в базис они не входят), следовательно, второй и третий ресурсы использованы полностью.

Таким образом, при применении симплексного метода не только находится оптимальный план, но и полезная для анализа задачи дополнительная информация, например, об объемах ресурсов, не используемых в этом плане. Вполне определенный экономический смысл имеют и оценки переменных, полученные при последней итерации. Его выяснение отложим до рассмотрения теории двойственности.

Современные компьютерные программы могут предусматривать и получение дополнительных отчетов. Полученный оптимальный план не предусматривает производство третьего вида продукции. Предположим, что по тем или иным причинам ее производство нужно обеспечить. Введение дополнительного ограничения на минимальный объем производства этой продукции приведет к снижению прибыли, значит в реальной действительности производство будет искать способы обойти это ограничение. Чтобы получить желаемый результат экономическими средствами, достаточно повысить прибыльность этой продукции (или снизить прибыльность других видов продукции). Например, при использовании *Excel* можно получить отчет об устойчивости решения по отношению к коэффициентам целевой функции. Очевидно, что для третьего вида продукции должно существовать такое минимальное значение величины прибыли, при котором эта продукция войдет в оптимальное решение. Удобнее получить это значение из дополнительного отчета, чем выполнять перебор вариантов.

## 2.14. Альтернативные варианты при использовании симплекс-метода

При использовании симплекс-метода возможны следующие четыре особых случая:

1. Вырожденность.
2. Альтернативные оптимальные решения.
3. Неограниченные решения.
4. Отсутствие допустимых решений.

Выясним причины возникновения таких ситуаций и способы их интерпретации в реальных задачах.

**Вырожденность.** Если хоть одна базисная переменная нулевая, то базисное решение называют вырожденным. Такая ситуация вполне может возникнуть в процессе решения задачи. Тогда может случиться, что переход к новому базису не обеспечивает улучшения значения функционала. Как правило, при последующих итерациях вырожденность исчезает. Дело в том, что на практике ее возникновение объясняется наличием в исходной постановке задачи избыточных ограничений. В этом случае, когда искусственная переменная, соответствующая одному из таких ограничений, выводится из базиса, функционал может и не улучшаться. Такая ситуация показана на рис. 2.12.

Из двух ограничений ресурсного типа второе является избыточным. При графическом решении (рис. 2.12) это очевидно, но из данных симплексной таблицы такой вывод сделать не удается. При исключении из базиса дополнительной переменной, соответствующей второму ограничению, значение функционала не возрастет. В итоге получим точку A, как оптимальное решение, которое не перестает быть верным от того, что оно вырожденное.

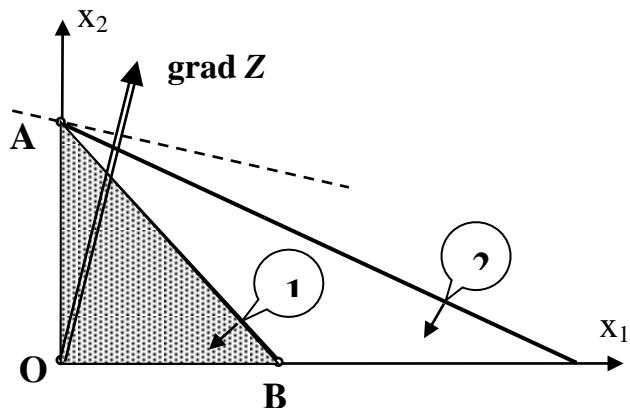


Рис. 2.12

Можно вообразить ситуацию, когда последовательность итераций при вырожденном базисе приведет к повторению уже имевшей место итерации, т. е. возникнет зацикливание. Искусственные примеры такого рода существуют, однако в реальных задачах зацикливание столь мало вероятно, что в большинстве программ, реализующих симплекс-метод, средства защиты от зацикливания не предусматриваются, так как они значительно замедляют процесс вычислений. Во-первых, нужно контролировать возникновение зацикливания, во-вторых, реализовать алгоритм для изменения порядка выбора разрешающих строк и столбцов.

Если подобные затруднения все же возникнут при компьютерном расчете, то можно попробовать произвести незначительные изменения отдельных коэффициентов модели (в пределах точности их определения) и повторить расчет. Если ранее имело место зацикливание, то порядок выбора разрешающих элементов, скорее всего, изменится.

*Альтернативные оптимальные решения.* Согласно основной теореме линейного программирования, если целевая функция достигает экстремального значения более чем в одной угловой точке, то она принимает это же значение в любой точке, являющейся их выпуклой линейной комбинацией. Графическая иллюстрация для случая двух переменных приведена на рис. 2.13. Максимальное значение целевая функция получает в угловых точках  $A$  и  $B$ . Предположим, что при использовании симплекс-метода получена точка  $A$  как оптимальный план. Тогда небазисной переменной  $x_1$  будет соответствовать нулевая оценка, так как при ее введении в базис с последующим переходом к точке  $B$  значение функционала не изменится.

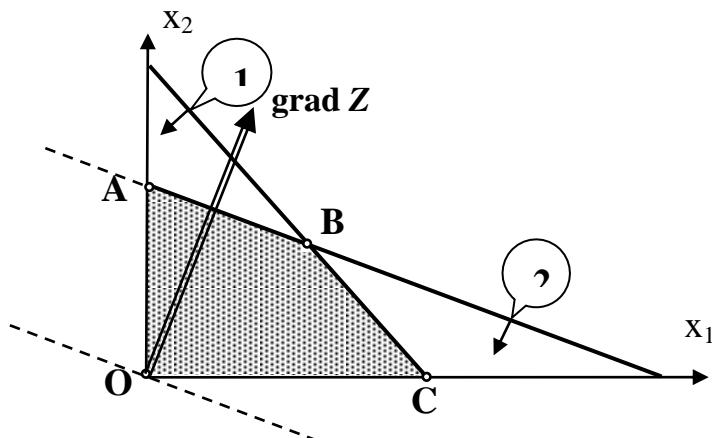


Рис. 2.13

Таким образом, признаком существования альтернативных планов является наличие нулевых оценок для небазисных переменных. На практике существование альтернативных решений можно использовать для учета дополнительных соображений при выборе плана действий. Если рассмотренный пример интерпретировать как задачу об оптимальном производственном плане, то лучше выбрать точку  $B$ , чтобы меньше зависеть от изменений рыночной конъюнктуры.

Если все оценки небазисных переменных строго больше нуля, то найденный оптимальный план является единственным.

С практической точки зрения, малые положительные значения оценок не очень существенно отличаются от нулевых хотя бы потому, что большая часть используемых данных определена с некоторой погрешностью. Следовательно, есть основания выполнить расчеты дополнительных вариантов, чтобы оценить реальный эффект от введения таких переменных в принимаемый план.

*Неограниченные решения.* Неограниченное возрастание переменных без нарушения ограничений – это явный признак ошибки. Например, в задаче о производственном плане каждый ресурс ограничен. Мало вероятно, что при записи модели такие ограничения пропущены, но при их вводе возможна ошибка в знаках. При использовании компьютерных программ, если не указать требование неотрицательности переменных, придавая формально одной из переменных отрицательное значение, программа может получать ресурсы для неограниченного роста других переменных.

Если ошибку не удается быстро обнаружить, то можно ввести дополнительные ограничения, например, на минимальные и максимальные значения переменных, задавая их с большим допуском по отношению к реально возможным значениям. Ошибка

выявится при анализе ограничений, в которых участвуют переменные, значения которых вышли за пределы допустимого.

*Отсутствие допустимых решений.* Такая ситуация возникает при несовместности ограничений задачи, что можно считать ошибкой в ее постановке или в используемых данных. На рис 2.14 показаны ограничения по минимальным объемам производства каждого из двух видов продукции. Видно, что третье ограничение, определяющее максимально допустимое использование некоторого ресурса, позволяет удовлетворить любое из этих требований, но не оба одновременно.

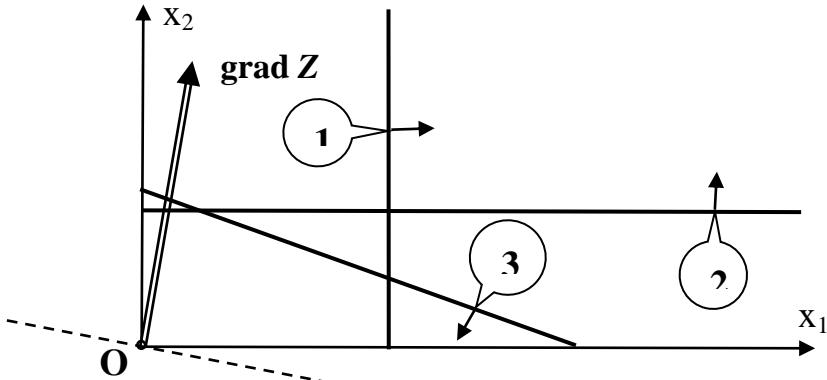


Рис. 2.14

Такие противоречия, естественно, возникают в условиях плановой экономики при разных целевых установках центрального планирующего органа и руководителей производства. Первый, как правило, – стремится к увеличению объема плановых заданий, – последним же предпочтительнее такой план, который легче выполнить и получить поощрения за перевыполнение. Когда с этой целью искажаются фактические данные, то может случиться, что ретроспективный расчет плана по отчетным данным покажет, что допустимые решения отсутствуют.

Рыночная экономика также не свободна от подобных проблем. Данные могут целенаправленно искажаться, например, для того, чтобы повлиять на текущий курс акций. В любом случае оптимизационная модель становится эффективным средством контроля и анализа всей совокупности статистической отчетности.

### 3.1. Прямая и двойственная задачи ЛП.

Сформулируем задачу оптимального использования ресурсов (прямую задачу ЛП). Возможен выпуск  $n$  различных видов продукции  $\Pi_j$ , ( $j = \overline{1, n}$ ). Используемые для этого ресурсы (виды сырья, группы оборудования, рабочая сила различной квалификации, производственные площади, финансовые средства и т. д.) общим числом  $m$ , ограничены величинами  $b_i$ , ( $i = \overline{1, m}$ ). Пусть  $a_{ij}$  — норма расхода сырья  $i$ -го вида на единицу  $j$ -й ( $j = \overline{1, n}$ ) продукции,  $c_j$  — цена реализации единицы  $j$ -й продукции (реализация обеспечена).

Построение модели начинают с введения переменных. Объемы выпуска каждого вида продукции обозначим через  $x_j$  ( $j = \overline{1, n}$ ).

Требуется найти план выпуска продукции, обеспечивающий максимальную прибыль  $Z$ . Поскольку прибыль от производства единицы продукции каждого вида  $c_j$ , то общая прибыль  $Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$ .

Информацию удобно представить в табличной форме.

Вид ресурса, $i$	Расход $i$ -го ресурса на единицу $j$ -й продукции, $a_{ij}$				Запас ресурса, $b_i$
	1	2	...	$n$	
1	$a_{11}$	$a_{12}$	...	$a_{1n}$	$b_1$
2	$a_{21}$	$a_{22}$	...	$a_{2n}$	$b_2$
...	...	...	...	...	...
$m$	$a_{m1}$	$a_{m2}$	...	$a_{mn}$	$b_m$
Прибыль, $c_j$	$c_1$	$c_2$	...	$c_n$	

Сформулируем ограничения на ресурсы:

$$\left\{ \begin{array}{l} \text{Необходимый объем ресурсов} \\ \text{для производства всей продукции} \end{array} \right\} \leq \left\{ \begin{array}{l} \text{Располагаемые} \\ \text{ресурсы} \end{array} \right\}.$$

Математическая модель задачи примет вид:

$$\begin{aligned} \max Z &= c_1x_1 + c_2x_2 + \dots + c_nx_n; \\ &a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1; \\ &a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2; \\ &\dots \dots \dots \dots \dots \dots \dots \dots \dots \\ &a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m; \\ &x_1 \geq 0, \quad x_2 \geq 0, \quad \dots, \quad x_n \geq 0. \end{aligned} \quad \left. \right\}$$

Дополнительно введенные условия неотрицательности определяются экономическим смыслом переменных.

Сформулируем двойственную задачу. На предприятии решили рационально использовать отходы основного производства. В плановом периоде появились ресурсы — отходы сырья  $m$  видов в объемах  $b_i$  единиц ( $i = \overline{1, m}$ ). Из них, учитывая специализацию предприятия, можно наладить выпуск  $n$  видов неосновной продукции либо реализовать

эти ресурсы сторонней организации. Какие обстоятельства необходимо учесть при продаже? Необходимо установить оценки (цены) на ресурсы  $y_1, y_2, \dots, y_m$ . Обозначим затраты покупателя через  $f$ . Оценки должны быть установлены исходя из следующих требований, отражающих несовпадающие интересы предприятия и организации-покупателя:

1) Общую стоимость ресурсов (сырья) покупающая организация стремится минимизировать. Интересы покупателя будут учтены, если:

$$\min f = b_1 y_1 + b_2 y_2 + \dots + b_m y_m.$$

2) Чтобы не снижать доход, выручка от продажи ресурсов должна быть не меньше, чем прибыль от их использования в собственном производстве (предприятие согласно уступить ресурсы только по таким ценам, при которых оно получит за них выручку, не меньшую той, что могло бы получить, организовав собственное производство).

Так как затраты сырья на производство единицы первого вида продукции определяются первым столбцом матрицы коэффициентов исходной задачи, то предприятие откажется от выпуска каждой единицы продукции первого вида, если выполняется условие

$$a_{11}y_1 + a_{21}y_2 + \dots + a_{m1}y_m \geq c_1,$$

где  $c_1$  – прибыль от производства единицы первого вида продукции (т.е. выручка за сырье, идущее на единицу продукции первого вида превышает ее цену).

Аналогичные рассуждения легко провести в отношении выпуска продукции каждого вида. Поэтому требование предприятия, реализующего отходы сырья, можно формализовать в виде следующей модели:

$$\begin{aligned} \min f &= b_1 y_1 + b_2 y_2 + \dots + b_m y_m; \\ a_{11}y_1 + a_{21}y_2 + \dots + a_{m1}y_m &\geq c_1; \\ a_{12}y_1 + a_{22}y_2 + \dots + a_{m2}y_m &\geq c_2; \\ &\dots \\ a_{1n}y_1 + a_{2n}y_2 + \dots + a_{mn}y_m &\geq c_n; \\ y_1 \geq 0, y_2 \geq 0, \dots, y_m &\geq 0. \end{aligned} \quad \left. \right\}$$

(по смыслу задачи оценки должны быть неотрицательными).

Значение расчета цен сырья состоит в оценке их использования и влияния на объем и эффективность производства. Переменные  $y_i$  ( $i = \overline{1, m}$ ) называют **двойственными оценками**, или объективно обусловленными оценками. В зарубежной литературе их иногда называют **теневыми ценами**. Эти цены определяются эффективностью собственного производства. Если в качестве примера представить себе некоторый регион, богатый лесом, но не располагающий возможностями для его переработки, то можно предполагать, что продаваться этот лес будет дешево. При развитии собственной лесопереработки цены на продажу леса сразу возрастут.

Пара взаимно двойственных симметричных задач имеет вид:

**прямая задача**

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

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad (i = \overline{1, m}),$$

$$x_j \geq 0, \quad (j = \overline{1, n});$$

**двойственная задача**

$$\min f = \sum_{i=1}^m b_i y_i,$$

$$\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad (j = \overline{1, n}),$$

$$y_i \geq 0, \quad (i = \overline{1, m}).$$

Правила перехода можно сформулировать так:

1. Если прямая задача – на максимум, то двойственная – на минимум, и наоборот.
2. Каждому из  $m$  ограничений прямой задачи соответствует переменная двойственной задачи.
3. Каждой из  $n$  переменных прямой задачи соответствует ограничение двойственной.
4. Коэффициенты  $c_j$  целевой функции прямой задачи являются свободными членами ограничений двойственной.
5. Свободные члены  $b_i$  ограничений прямой задачи являются коэффициентами целевой функции двойственной.
6. Матрица коэффициентов ограничений двойственной задачи получается транспонированием матрицы коэффициентов прямой задачи, и наоборот.
7. Для пары взаимно-двойственных симметричных задач все переменные неотрицательны; задаче на максимум соответствуют неравенства типа  $\leq$ , а задаче на минимум – вида  $\geq$ .

Правила построения двойственных задач для случая, когда прямая задача представлена в общей форме:

<i>Задача максимизации</i>	<i>Задача минимизации</i>
<u>Ограничения</u>	<u>Переменные</u>
$\leq$	$\geq 0$
$=$	Любого знака
<u>Переменные</u>	<u>Ограничения</u>
$\geq 0$	$\geq$
Любого знака	$=$
<i>прямая задача</i>	
$\max Z = \sum_{j=1}^n c_j x_j;$	$\min f = \sum_{i=1}^m b_i y_i;$
$\sum_{j=1}^n a_{ij} x_j \leq b_i, (i = \overline{1, m_1}),$	$\sum_{i=1}^m a_{ij} y_i \geq c_j, (j = \overline{1, n_1}),$
$\sum_{j=1}^n a_{ij} x_j = b_i, (i = \overline{m_1 + 1, m}),$	$\sum_{i=1}^m a_{ij} y_i = c_j, (j = \overline{n_1 + 1, n}),$
$x_j \geq 0, (j = \overline{1, n_1}; n_1 \leq n),$	$y_i \geq 0, (i = \overline{1, m_1}; m_1 \leq m),$
$x_j - \forall \text{знака}$	$y_i - \forall \text{знака}$
$(j = \overline{n_1 + 1, n}),$	$(i = \overline{m_1 + 1, m}).$

### 3.2. Соотношения между решениями прямой и двойственной задач

Тесная взаимосвязь прямой и двойственной задач позволяет сформулировать ряд общих закономерностей.

**Теорема (Основное неравенство теории двойственности):**

Для любых допустимых планов  $X = (x_1, x_2, \dots, x_n)$  и  $Y = (y_1, y_2, \dots, y_m)$  прямой и двойственной задач справедливо неравенство  $Z(X) \leq f(Y)$ , т. е.

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y_i.$$

Экономическая интерпретация: для любого допустимого плана производства  $x$  и любого допустимого вектора оценок ресурсов  $y$  общая созданная стоимость не превосходит суммарной оценки ресурсов.

**Теорема (Критерий оптимальности Канторовича):**

Если для некоторых допустимых планов  $X^*$  и  $Y^*$  пары двойственных задач выполняется равенство  $Z(X^*) = f(Y^*)$ , то  $X^*$  и  $Y^*$  являются оптимальными планами соответствующих задач.

Экономическое содержание теоремы состоит в том, что план производства  $X$  и вектор оценок ресурсов  $Y$  являются оптимальными, если цена всей произведенной продукции и суммарная оценка ресурсов совпадают.

**Теорема (Малая теорема двойственности):**

Для существования оптимального плана любой из пары двойственных задач необходимо и достаточно существование допустимого плана для каждой из них.

Между решениями прямой и двойственной задач существует следующее соответствие: основным переменным прямой задачи соответствуют дополнительные переменные двойственной и наоборот.

### 3.3. Основные теоремы двойственности и их экономическое содержание

#### Теорема (первая теорема двойственности).

Если одна из двойственных задач имеет оптимальное решение, то и другая имеет оптимальное решение, причем экстремальные значения целевых функций равны. Если одна из двойственных задач неразрешима вследствие неограниченности целевой функции на множестве допустимых планов, то система ограничений другой задачи противоречива (т.е. не имеет допустимых планов).

Экономическая интерпретация теоремы: если разрешима задача определения оптимального плана производства, максимизирующего выпуск продукции, то разрешима и задача определения оценок ресурсов. Чтобы план производства и вектор оценок ресурсов были оптимальны, необходимо и достаточно, чтобы доход и суммарная оценка ресурсов совпадали (полное отсутствие непроизводительных затрат). Несовпадение говорит о том, что система находится в нестабильном состоянии, поскольку стоимость потребляемых ресурсов превышает получаемый доход. Таким образом, двойственные оценки выступают как инструмент балансировки затрат и результатов.

Второе утверждение теоремы можно интерпретировать так: предположим, что ресурсов недостаточно для обеспечения заданного плана производства, тогда для его выполнения пришлось бы приобретать ресурсы по любой (формально неограниченной) цене. Совпадение значений целевых функций для планов двойственных задач достаточно для того, чтобы эти планы были оптимальными. Двойственные оценки гарантируют рентабельность оптимального плана и убыточность всякого другого, отличного от оптимального. Они позволяют сопоставить и сбалансировать затраты и результаты системы.

#### Теорема (о дополняющей нежесткости).

Для того чтобы планы  $X^*$  и  $Y^*$  пары двойственных задач были оптимальными, необходимо и достаточно выполнение условий:

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

Эти условия означают, что, если после оптимизации какая-либо переменная одной из задач положительна, то соответствующее ограничение двойственной задачи должно обращаться в строгое равенство.

Величина  $\sum_{i=1}^m a_{ij} y_i$  представляет суммарную стоимость всех ресурсов, используемых для производства единицы продукции, которая и должна равняться доходу от ее реализации  $c_j$ , если планы оптимальны. Разности  $\sum_{i=1}^m a_{ij} y_i - c_j$  часто называют приведенной стоимостью (приведенными издержками)  $j$ -го вида деятельности.

Величина  $b_i - \sum_{j=1}^n a_{ij} x_j$  – это неиспользованное количество  $i$ -го ресурса, которое для оптимального плана производства может быть положительным только для недефицитных (используемых не полностью ресурсов), которым соответствует нулевое значение двойственной оценки  $y_i^*$ .

Таким образом, если

$$\sum_{j=1}^n a_{ij}x_j^* < b_i, \text{ то } y_i^* = 0, \text{ если } y_i^* > 0, \text{ то } \sum_{j=1}^n a_{ij}x_j^* = b_i.$$

### **Теорема (об оценках).**

*Двойственные оценки показывают приращение целевой функции, вызванное малым изменением свободного члена соответствующего ограничения ЗЛП. Значения переменных  $y_i$  в оптимальном решении двойственной задачи представляют собой оценки влияния свободных членов системы ограничений исходной задачи на оптимальное значение ее целевой функции. Двойственные оценки численно равны изменению целевой функции при изменении соответствующего свободного члена ограничений на единицу:*

$$\frac{\partial Z(X^*)}{\partial b_i} = y_i, \quad (i = 1, m).$$

Экономический смысл теоремы – единичный прирост ресурса (при оптимальном его использовании) позволяет увеличить доход на величину, равную теневой цене этого ресурса.

При изменении  $i$ -го ресурса оптимальный доход является линейной функцией от его приращения, причем коэффициентом служит  $y_i$  —  $i$ -я компонента оптимального решения двойственной задачи.

Если  $y_i$  мало, то значительному увеличению  $i$ -го ресурса будет соответствовать небольшое увеличение оптимального дохода и ценность ресурса невелика.

Если  $y_i = 0$ , то при увеличении  $i$ -го ресурса оптимальный доход остается неизменным и ценность этого ресурса равна нулю. В самом деле, сырье, запасы которого превышают потребности в нем, не представляет ценности для производства и его оценку можно принять за нуль.

Если  $y_i$  велико, то незначительному увеличению  $i$ -го ресурса будет соответствовать существенное увеличение оптимального дохода и ценность ресурса высока. Уменьшение ресурса ведет к существенному сокращению выпуска продукции.

Переменную  $y_i$  считают некоторой характеристикой ценности  $i$ -го ресурса. В частности, при увеличении  $i$ -го ресурса на единицу ( $\Delta b_i = 1$ ) оптимальный доход возрастает на  $y_i$ , что позволяет рассматривать  $y_i$  как "условную цену", оценку единицы  $i$ -го ресурса, объективно обусловленную оценку.

Так как  $y_i$  представляет частную производную от оптимального дохода по  $i$ -му ресурсу, то  $y_i$  характеризует скорость изменения оптимального дохода при изменении  $i$ -го ресурса.

Экономическая интерпретация двойственных оценок не всегда возможна.

### 3.4. Применение двойственных оценок в послеоптимизационном анализе

С помощью  $y_i$  можно определить степень влияния ограничений на значение целевой функции. Предельные значения (нижняя и верхняя границы) ограничений ресурсов, для которых  $y_i$  остаются неизменными, определяются по формулам:

$$b_i^H = \min\left(\frac{x_j}{d_{ij}}\right), b_i^B = \max\left(\frac{x_j}{d_{ij}}\right),$$

где  $x_j$  — значение переменной в оптимальном решении;  $d_{ij}$  — элементы матрицы, обратной к матрице базиса оптимального решения, для которой  $A = (a_{ij})$ .

Если в план включаются новые виды продукции, то их оценка находится по формуле

$$\Delta_j = \sum_{i=1}^m a_{ij} y_{onmi} - c_j.$$

Если  $\Delta_j < 0$ , то новый вид продукции улучшает план. При  $\Delta_j > 0$  нецелесообразно включать новый вид продукции.

Рассмотрим проблему взаимозаменяемости ресурсов. Дефицитным называется ресурс, который при реализации оптимального плана полностью используется. Если в ЗЛП имеются два или больше дефицитных ресурсов, то представляет интерес, каким количеством одного дефицитного ресурса можно заменить другой дефицитный ресурс, чтобы целевая функция принимала то же самое максимальное значение.

Введем понятие коэффициента взаимозаменяемости. Пусть  $b_i$  и  $b_k$  — объемы дефицитных ресурсов  $i$ -го и  $k$ -го вида. Обозначим отношение  $\Delta b_k / \Delta b_i$  через  $\eta_{ki}$ , т.е.

$$\eta_{ki} = \frac{\Delta b_k}{\Delta b_i}.$$

Это отношение называется **коэффициентом взаимозаменяемости**. При  $\Delta b_i = 1$  имеем  $\eta_{ki} = \Delta b_k$ , т.е. коэффициент взаимозаменяемости показывает, насколько нужно увеличить ресурс  $k$ -го вида, чтобы компенсировать уменьшение ресурса  $i$ -го вида на единицу. Заменив в соотношении  $\eta_{ki} = \frac{\Delta b_k}{\Delta b_i}$  приращения на дифференциалы, получим

$$\frac{\Delta z}{\Delta b_i} \approx \frac{\partial z(x^*)}{\partial b_i} = y_i^*, \quad \frac{\Delta z}{\Delta b_k} \approx \frac{\partial z(x^*)}{\partial b_k} = y_k^*,$$

$$\eta_{ki} = \frac{\Delta z / y_k^*}{\Delta z / y_i^*} = \frac{y_i^*}{y_k^*}.$$

Если ресурс  $i$ -го вида уменьшить на  $\Delta b_i$ , то, увеличив ресурс  $k$ -го вида на  $\Delta b_k^* = (\frac{y_i^*}{y_k^*}) \Delta b_i$ , можно компенсировать уменьшение целевой функции, при этом сам оптимальный план изменится.

### 3.5. Примеры

**Пример 1.** Пусть фирма может выпускать 3 вида продукции  $\Pi_j$ , где  $(j = \overline{1,3})$ . При ее изготовлении используются три вида ресурсов (труд, сырье и оборудование), расход которых  $b_i$ , где  $(i = \overline{1,3})$ , ограничен величинами  $b_1 = 180$ ,  $b_2 = 60$  и  $b_3 = 120$ . Затраты ресурса  $i$ -го вида на единицу продукции  $j$ -го вида составляют  $a_{ij}$  единиц. Прибыль от производства единицы продукции  $j$ -го вида составляет  $c_j$  единиц. Значения параметров приведены в табл. 3.1

Таблица 3.1

Вид ресурса, $i$	Расход $i$ -го ресурса на единицу $j$ -ой продукции, $a_{ij}$			Запас ресурса, $b_i$
	$\Pi_1$	$\Pi_2$	$\Pi_3$	
Труд	6	2	5	180
Сырье		2	4	60
Оборудование	5	2	5	120
Прибыль, $c_j$	9	6	7	

Математическая модель задачи оптимизации производственного плана по величине

$$\begin{aligned} \max Z &= 9x_1 + 6x_2 + 7x_3; \\ 6x_1 + 2x_2 + 5x_3 &\leq 180; \\ 2x_2 + 4x_3 &\leq 60; \\ 5x_1 + 2x_2 + 5x_3 &\leq 120; \\ x_j &\geq 0 \quad (j = \overline{1,3}). \end{aligned} \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\}$$

прибыли примет вид

Преобразовывая модель к канонической форме и предпочтительному виду, получим:

$$\begin{aligned} \max Z &= 9x_1 + 6x_2 + 7x_3 + 0x_4 + 0x_5 + 0x_6; \\ 6x_1 + 2x_2 + 5x_3 + x_4 &= 180; \\ 2x_2 + 4x_3 + x_5 &= 60; \\ 5x_1 + 2x_2 + 5x_3 + x_6 &= 120; \\ x_j &\geq 0 \quad (j = \overline{1,6}). \end{aligned} \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\}$$

Занесем условия задачи в симплексную таблицу (табл. 3.2). Значения базисных переменных  $x_4$ ,  $x_5$  и  $x_6$  будут равны правым частям системы ограничений. Значение целевой функции для опорного плана вычисляется как скалярное произведение  $\Delta_0 = \mathbf{c}_B \mathbf{b}$ , где  $\mathbf{c}_B$  – вектор коэффициентов целевой функции при базисных переменных, а  $\mathbf{b}$  – вектор значений базисных переменных. Оценки свободных переменных вычисляются по формуле  $\Delta_j = \mathbf{c}_B \mathbf{A}_j - c_j$  ( $j = \overline{1,3}$ ), где  $\mathbf{A}_j$  вектор – столбец коэффициентов при переменной  $x_j$ .

Таблица 3.2.

Номер итерации	БП	$\mathbf{c}_B$	$\mathbf{b}$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	Симплексные отношения
				9	6	7	0	0	0	

0	$x_4$	0	180	6	2	5	1	0	0	$180/6 = 30$
	$x_5$	0	60	0	2	4	0	1	0	—
	$x_6$	0	120	5	2	5	0	0	1	$120/5 = 24$
	Оценки		$\Delta_0$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$	
1			0	-9	-6	-7	0	0	0	
$x_4$	0	36	0	-2/5	-1	1	0	-6/5	—	
$x_5$	0	60	0	2	4	0	1	0	$60/2 = 30$	
$x_1$	9	24	1	2/5	1	0	0	1/5	$24/(2/5) = 60$	
Оценки		$\Delta_0$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$		
2		216	0	-2,4	2	0	0	1,8		
$x_4$	0	48	1	0	-1/5	1	1/5	-6/5	—	
$x_2$	6	30	0	1	2	0	1/2	0	—	
$x_1$	9	12	0	0	1/5	0	-1/5	1/5	—	
Оценки		$\Delta_0$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$		
2		288	0	0	6,8	0	1,2	1,8		

После выполнения второй итерации получаем, что все оценки переменных положительны, а, следовательно, оптимальный по прибыли план найден.

Это вектор  $\mathbf{x}^* = (x_1, x_2, x_3, x_4, x_5, x_6) = (12, 30, 0, 48, 0, 0)$ , которому соответствует прибыль  $Z^* = Z(\mathbf{x}^*) = 288$ . В этот оптимальный план вошла дополнительная переменная  $x_4 = 48$ , означающая, что такое количество единиц первого ресурса не используется в оптимальном плане. Переменные  $x_5$  и  $x_6$  равны нулю (в базис они не входят), следовательно, второй и третий ресурсы использованы полностью.

Таким образом, при применении симплексного метода не только находится оптимальный план, но и полезная для анализа задачи дополнительная информация, например, об объемах ресурсов, не используемых в этом плане.

Запишем модель двойственной задачи.

Модель прямой задачи

$$\begin{aligned} \max Z &= 9x_1 + 6x_2 + 7x_3; \\ 6x_1 + 2x_2 + 5x_3 &\leq 180; \\ 2x_2 + 4x_3 &\leq 60; \\ 5x_1 + 2x_2 + 5x_3 &\leq 120; \\ x_j &\geq 0 \quad (j = \overline{1,3}). \end{aligned} \quad \left. \right\}$$

Модель двойственной задачи, согласно приведенным правилам, примет вид:

$$\begin{aligned} \min f &= 180y_1 + 60y_2 + 120y_3; \\ 6y_1 &\quad + 5y_3 \geq 9; \\ 2y_1 + 2y_2 + 2y_3 &\geq 6; \\ 5y_1 + 4y_2 + 5y_3 &\geq 7; \\ y_i &\geq 0 \quad (i = \overline{1,3}). \end{aligned} \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\}$$

Если записать исходное и конечное состояния симплексной таблицы, то видно соответствие между переменными прямой и двойственной задач.

Номер итерации	БП	<b>c<sub>Б</sub></b>	<b>b</b>	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
				9	6	7	0	0	0
0	$x_4$	0	180	6	2	5	1	0	0
	$x_5$	0	60	0	2	4	0	1	0
	$x_6$	0	120	5	2	5	0	0	1
	Оценки		$\Delta_0$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$
2			0	-9	-6	-7	0	0	0
	$x_4$	0	48	1	0	-1/5	1	1/5	-6/5
	$x_2$	6	30	0	1	2	0	1/2	0
	$x_1$	9	12	0	0	1/5	0	-1/5	1/5
	Оценки		$\Delta_0$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$
Соответствующие переменные двойственной задачи				$y_4$	$y_5$	$y_6$	$y_1$	$y_2$	$y_3$

Содержательный смысл оценок переменных прямой задачи в том, что они показывают, на сколько уменьшится целевая функция, если соответствующая переменная увеличится на единицу. Значит оценки переменных  $x_4, x_5, x_6$  это и есть оптимальные значения переменных двойственной задачи, т. е. двойственные оценки.

Например, если на последней итерации (при получении оптимального решения) оценка  $\Delta_6 = 1,8$ , то эта величина есть отклонение целевой функции от оптимального значения при присвоении переменной  $x_6$  единичного значения. Но это равносильно уменьшению третьего ресурса на единицу (достаточно посмотреть на третью строку нулевой итерации), т. е. это цена третьего ресурса.

Следовательно, значения  $y_1^* = 0, y_2^* = 1,2, y_3^* = 1,8$  – это оценки соответствующих ресурсов. Первый ресурс недефицитный (ранее было отмечено, что величину этого ресурса можно уменьшить на 48 единиц без изменения прибыли), а наиболее дефицитным является третий ресурс.

Если подставить найденные цены ресурсов в математическую модель двойственной задачи, то целевая функция должна получить значение 288, как и при решении прямой задачи. Действительно, имеем:

$$180 \cdot 0 + 60 \cdot 1,2 + 120 \cdot 1,8 = 72 + 216 = 288.$$

Оценки переменных  $x_1, x_2, x_3$  – это значения свободных переменных двойственной задачи  $y_4^* = 0, y_5^* = 0, y_6^* = 6,8$ , которые можно истолковать так: если вместо продажи

части ресурсов по оптимальным двойственным ценам привести по единице каждого из возможных видов продукции, то производство третьего вида продукции повлечет указанный убыток, а производство первых двух видов прибыль не изменит.

Иными словами, при производстве первых двух видов продукции получаемый доход равен стоимости используемых ресурсов.

**Пример 2.** Исходя из специализации и технологических возможностей предприятия может выпускать четыре вида продукции. Сбыт любого количества обеспечен. Для изготовления этой продукции используются трудовые ресурсы, полуфабрикаты и станочное оборудование. Общий объем ресурсов (в расчете на трудовую неделю), расход каждого ресурса на единицу выпускаемой продукции и прибыль, полученная за единицу продукции, приведены в табл. Требуется определить план выпуска, доставляющий предприятию максимум прибыли.

Таблица 3.3

Вид ресурса, $i$		Выпускаемая продукция				Запас ресурса
		$\Pi_1$	$\Pi_2$	$\Pi_3$	$\Pi_4$	
$P_1$	Трудовые ресурсы, чел.-ч.	4	2	2	8	4800
$P_2$	Полуфабрикаты, кг	2	10	6	0	2400
$P_3$	Станочное оборудование, станко-ч.	1	0	2	1	1500
Цена единицы продукции, ден. ед.		65	70	60	120	

Математическая модель прямой задачи:

$$\begin{aligned} \max Z &= 65x_1 + 70x_2 + 60x_3 + 120x_4; \\ 4x_1 + 2x_2 + 2x_3 + 8x_4 &\leq 4800, \\ 2x_1 + 10x_2 + 6x_3 &\leq 2400, \\ x_1 + 2x_3 + x_4 &\leq 1500, \\ x_j &\geq 0, (j = \overline{1, 4}). \end{aligned}$$

Z	65	70	60	120		max
$y_i$	4	2	2	8	$\leq$	4800
	2	10	6	0		2400
	1	0	2	1		1500

Составим двойственную задачу. Транспонируем таблицу:

f	4800	2400	1500		min
	4	2	1	$\geq$	65
	2	10	0		70
	2	6	2		60
	8	0	1		120

Математическая модель двойственной задачи:

$$\min f = 4800y_1 + 2400y_2 + 1500y_3;$$

$$\left. \begin{array}{l} 4y_1 + 2y_2 + y_3 \geq 65, \\ 2y_1 + 10y_2 \geq 70, \\ 2y_1 + 6y_2 + 2y_3 \geq 60, \\ 8y_1 + y_3 \geq 120, \end{array} \right\}$$

$$y_j \geq 0, (j=1,3).$$

Симплексным методом решили прямую задачу:  $x^* = (0; 0; 400; 500; 0; 0; 200)$ ,  $Z^* = Z(x^*) = 84000$ .

Номер итерации	БП	$\mathbf{c}_B$	$\mathbf{b}$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
				65	70	60	120	0	0	0
0	$x_5$	0	4800	4	2	2	<b>8</b>	1	0	0
	$x_6$	0	2400	2	10	46	0	0	1	0
	$x_7$	0	1500	1	0	2	1	0	0	1
	Оценки		$\Delta_0$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$	$\Delta_7$
1			0	-65	-70	-60	-120	0	0	
	$x_4$	120	600	1/2	1/4	1/4	1	1/8	0	0
	$x_6$	0	2400	2	0	6	0	0	1	0
	$x_7$	0	900	1/2	-1/4	7/4	0	-1/8	0	1
2	Оценки		$\Delta_0$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$	
			72000	-5	-40	-30	0	15	0	0
	$x_4$	120	500	5/12	-1/6	0	1	1/8	-1/24	0
	$x_3$	60	400	1/3	5/3	1	0	0	1/6	0
2	$x_7$	0	200	-	-19,6	0	0	-1/8	-	1
	Оценки		1/12						7/24	
			$\Delta_0$	$\Delta_1$	$\Delta_2$	$\Delta_3$	$\Delta_4$	$\Delta_5$	$\Delta_6$	
			84000	5	10	0	0	15	5	0
Соответствующие переменные двойственной задачи				$y_4$	$y_5$	$y_6$	$y_7$	$y_1$	$y_2$	$y_3$

Учитывая это соответствие, выписываем из индексной строки последней (2-й) итерации компоненты искомого оптимального плана  $y^* = (15; 5; 0; 5; 10; 0; 0)$  — двойственные оценки.

В соответствии с теоремой  $\min f = \max Z = 84000$ . Запишем это равенство в развернутой форме:  $48000 \cdot 15 + 2400 \cdot 5 + 1500 \cdot 0 = 65 \cdot 0 + 70 \cdot 0 + 60 \cdot 400 + 120 \cdot 500$

Учитывая, что компоненты  $y_1^* = 15, y_2^* = 5, y_3^* = 0$  представляют собой оценки ресурсов  $P_1, P_2, P_3$ , заключаем: при оптимальном плане оценка ресурсов, затраченных на выпуск продукции, совпадает с оценкой произведенной продукции. В этом состоит экономическое содержание теоремы.

Таким образом, оптимальность плана означает точное воплощение в оценке произведенной по этому плану продукции оценки всех израсходованных ресурсов, т. е. полное отсутствие непроизводительных затрат.

Найден оптимальный план  $x^* = (0; 0; 400; 500; 0; 0; 200)$  выпуска продукции. При этом плане третье ограничение прямой задачи выполняется как строгое неравенство:  $0 + 2400 + 500 = 1300 < 1500$ . Это означает, что расход ресурса  $P_3$  меньше его запаса, т. е. ресурс  $P_3$  избыточный. Именно поэтому в оптимальном плане  $y^* = (15; 5; 0; 5; 10; 0; 0)$  двойственной задачи оценка  $y_3^*$  этого ресурса равна нулю.

А вот оценки  $y_1^*$  и  $y_2^*$  ресурсов  $P_1$  и  $P_2$  выражаются положительными числами 15 и 5, что свидетельствует о дефицитности этих ресурсов: они при оптимальном плане используются полностью. В самом деле, ограничения по этим ресурсам выполняются как строгие равенства:  $40 + 20 + 2400 + 8500 = 4800$ ,  $20 + 100 + 6400 = 2400$ .

Поскольку  $15 > 5$ , ресурс  $P_1$  считается более дефицитным, чем ресурс  $P_2$ . Это мы подтвердим более убедительно позднее.

На основе теоремы 2 (о дополняющей нежесткости) нетрудно объяснить, почему не вошла в оптимальный план продукция  $\Pi_1$  и  $\Pi_2$ : первое и второе ограничения двойственной задачи выполняются как строгие неравенства:  $4 \cdot 15 + 2 \cdot 5 + 0 > 65$ ,  $2 \cdot 15 + 10 \cdot 5 > 70$ . Это означает, что оценки ресурсов, расходуемых на изготовление единицы продукции  $\Pi_1$  и  $\Pi_2$ , превышают оценки единицы этой продукции. Понятно, что такую продукцию выпускать предприятию невыгодно. Что же касается продукции  $\Pi_3$  и  $\Pi_4$  ( $x_3^* > 0, x_4^* > 0$ ), то выпуск ее оправдан, поскольку оценка израсходованных ресурсов совпадает с оценкой произведенной продукции:  $2 \cdot 15 + 6 \cdot 5 + 2 \cdot 0 = 60$ ,  $8 \cdot 15 + 0 = 120$ .

Установлено, что ресурсы  $P_1$  и  $P_2$  являются дефицитными. В связи с этим на основе теоремы 3 можно утверждать, что на каждую единицу ресурса  $P_i$ , введенную дополнительно в производство, будет получена дополнительная выручка  $\Delta_i Z$ , численно равная  $y_i^*$ . В самом деле, при  $\Delta b_1 = 1$  получаем  $\Delta_1 Z = y_1^* \Delta b_1 = 15 \cdot 1 = 15$ . По тем же причинам каждая дополнительная единица ресурса  $P_2$  обеспечит прирост  $\Delta_2 Z$  выручки, равный 5 ден. ед. Теперь становится понятно, почему ресурс  $P_1$  считается более дефицитным по сравнению с ресурсом  $P_2$ : он может содействовать получению большей выручки.

Что же касается избыточного ресурса  $P_3$ , то увеличение его запаса не приведет к росту выручки, поскольку  $\Delta_3 Z = y_3^* \Delta b_3 = 0$ .  $\Delta b_3 = 0$ . Из этих рассуждений следует, что оценки ресурсов позволяют совершенствовать план выпуска продукции.

Выясним экономический смысл оценок  $y_4^*, y_5^*, y_6^*, y_7^*$  продукции  $\Pi_1, \Pi_2, \Pi_3, \Pi_4$ . По оптимальному плану  $x^* = (0; 0; 400; 500; 0; 0; 200)$  выпускать следует продукцию  $\Pi_3$  и  $\Pi_4$ . Оценки  $y_6^*$  и  $y_7^*$  этих видов продукции равны нулю. Что это означает практически, станет ясно, если представить оценки в развернутой записи:

$$\begin{aligned} y_6^* &= (2y_1^* + 6y_2^* + 2y_3^*) - 60 = \\ &= (2 \cdot 15 + 6 \cdot 5 + 2 \cdot 0) - 60 = 0, y_7^* = (8y_1^* + y_3^*) - 120 = (8 \cdot 15 + 0) - 120 = 0. \end{aligned}$$

Таким образом, нулевая оценка показывает, что эта продукция является неубыточной, поскольку оценка ресурсов, расходуемых на выпуск единицы такой продукции, совпадает с оценкой единицы изготовленной продукции.

Что же касается продукции  $\Pi_1$  и  $\Pi_2$ , являющейся, как установлено ранее, убыточной, а потому и не вошедшей в оптимальный план, то для ее оценок  $y_4^* = 5$  и  $y_5^* = 10$  получаем:

$$y_4^* = (y_1^* + 2y_2^* + y_3^*) - 65 = 70 - 65 = 5, y_5^* = (2y_1^* + 10y_2^*) - 70 = 80 - 70 = 10. \quad \text{Отсюда видно, что оценка убыточной продукции показывает, насколько будет снижать каждая изготовленная единица такой продукции достигнутый оптимальный уровень выручки.}$$

Выясним состав двойственной оценки. Для этого рассмотрим, например, первый

ресурс (его запас  $b_1 = 4800$ ). Он дефицитен. Увеличение запаса этого ресурса на единицу приведет к дополнительному выпуску продукции, что увеличит выручку на  $y_1^* = 15$  ден. ед. За счет чего? Возьмем соответствующий столбец  $A_5 = (1/8; 0; -1/8)^T$  табл. Элементы его характеризуют изменение объемов выпуска продукции и остатка ресурса при увеличении первого ресурса на единицу, т. е. если заменить  $b_1$  на  $b_1' = b_1 + 1 = 4800 + 1 = 4801$ , то выпуск продукции  $\Pi_4 x_4 = 500$  заменится на  $x_4^* = 500 + 1/8 = 500,125$ ; выпуск продукции  $\Pi_3 x_3^* = 400$  — на  $x_3^* = 400 + 0 = 400$ . Резерв же третьего ресурса сократится до  $x_7^* = 200 - 1/8 = 199,875$ . При этом выручка возрастет на  $120 \cdot 1/8 + 60 \cdot 0 - 0 \cdot (-1/8) = 15$  ден. ед., что соответствует двойственной оценке первого ресурса. Аналогично при увеличении второго ресурса на единицу выручка возрастет на  $120 \cdot (-1/24) + 60 \cdot 1/16 + 0 \cdot (-7/24) = 5$  ден. ед., что соответствует двойственной оценке второго ресурса. Полученные равенства и показывают, какие составляющие образуют двойственные оценки.

Найдем коэффициент взаимозаменяемости ресурсов. В примере дефицитны трудовые ресурсы и полуфабрикаты. Если бы трудовые ресурсы уменьшили на единицу, то связанное с этим падение выручки (на 15 ден. ед.) можно было бы компенсировать увеличением полуфабрикатов на

$$\Delta b_2 = \frac{y_1^*}{y_2} \Delta b_1 = \frac{15}{5} \cdot 1 = 3.$$

Следовательно, обеспечив полуфабрикаты в объеме  $b_2' = b_2 + \Delta b_2 = 2400 + 3 = 2403$  (кг), можно получить с трудовыми ресурсами  $b_1' = b_1 - \Delta b_1 = 4800 - 1 = 4799$  (чел.-ч) ту же выручку, что и при начальных ресурсах. В табл. представлены значения коэффициентов взаимозаменяемости для примера. Знак  $\infty$  означает, что заменить уменьшение на единицу одного ресурса никаким увеличением другого невозможно.

$i \backslash k$	1	2	3
1	1	$1/3$	0
2	3	1	0
3	$\infty$	$\infty$	1

Проанализируем целесообразность расширения ассортимента выпускаемой продукции и установление цены на новую продукцию. Пусть в условиях примера изучается вопрос о целесообразности выпуска продукции  $\Pi_5$  с характеристиками, представленными в табл.

Чтобы выпуск продукции  $\Pi_5$  был оправдан, оценка ресурсов, затрачиваемых на изготовление единицы продукции  $\Pi_5$ , должна быть не менее цены  $c_5 = 95$ .

Вид ресурса, $i$		$\Pi_1$
$P_1$	Трудовые ресурсы, чел.-ч.	3
$P_2$	Полуфабрикаты, кг	6
$P_3$	Станочное оборудование, станко-ч.	8
Цена единицы продукции, ден. ед.		95

Находим оценку затраченных ресурсов:  $3 \cdot 15 + 6 \cdot 5 + 8 \cdot 0 = 75$ . Поскольку  $75 < 95$ , выпускать продукцию  $\Pi_5$  целесообразно: каждая единица этой продукции принесет предприятию прибыль, равную  $95 - 75 = 20$  ден. ед.

#### 4.1. Постановка и математическая модель транспортной задачи

Транспортные задачи (модели) – специальный класс задач линейного программирования. Они часто используются для оптимизации объемов перевозок из пунктов отправления в пункты назначения при минимальных суммарных затратах. При этом должны быть учтены как ограниченные возможности поставщиков по отправке грузов, так и заданные потребности получателей. Предполагается, что тарифы за перевозку единицы груза от любого поставщика к любому получателю известны и что стоимость перевозки по выбранному маршруту пропорциональна объему груза.

Модели транспортного типа часто используются для решения задач управления запасами, движением капиталов, назначением персонала, составления расписаний и др. Разумеется, эти задачи можно решать симплекс-методом, но для них существует более простой и эффективный алгоритм, использующий таблицы со специальной структурой. Таким образом достигается наглядность описания ситуации, а объем вычислительной работы сокращается настолько, что при небольшой размерности задачи ее можно решить даже без вычислительной техники.

Для формулировки транспортной задачи введем обозначения:

$m$  – число поставщиков (складов, пунктов отправления);

$n$  – число получателей (пунктов назначения);

$a_i$  – объем грузов в пункте отправления ( $i = \overline{1, m}$ );

$b_j$  – объем спроса в пункте получения ( $j = \overline{1, n}$ );

$c_{ij}$  – стоимость перевозки единицы груза из пункта  $i$  в пункт  $j$ ;

$x_{ij}$  – количество груза, перевозимого из пункта  $i$  в пункт  $j$ .

Чтобы при минимальных затратах учесть как запросы получателей, так и возможности поставщиков, сформулируем следующую задачу линейного программирования:

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \quad \text{– минимум суммарных затрат,}$$

$$\sum_{j=1}^n x_{ij} \leq a_i, \quad i = \overline{1, m} \quad \text{– возможности поставщиков (не более),}$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = \overline{1, n} \quad \text{– запросы получателей (точно),}$$

$$x_{ij} \geq 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Очевидно, что решение такой задачи можно найти, если суммарные возможности поставщиков  $\sum_{i=1}^m a_i = A$  достаточны, чтобы обеспечить все запросы получателей, равные

$$\sum_{j=1}^n b_j = B, \quad \text{т. е. } A \geq B. \quad \text{Если наоборот } A < B, \text{ то задача формально противоречива, и}$$

надо корректировать постановку задачи, принимая решение о том, какой или какие получатели недополучат заказанный груз. Например, можно записать модель так:

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min ,$$

$$\sum_{j=1}^n x_{ij} = a_i, \quad i = \overline{1, m} \quad \text{– возможности поставщиков (точно),}$$

$$\sum_{i=1}^m x_{ij} \leq b_j, \quad j = \overline{1, n} \quad \text{– запросы получателей (по возможности),}$$

$$x_{ij} \geq 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Модель транспортной задачи называют открытой, если  $A \neq B$ , и закрытой, когда  $A = B$ . Случай, когда спрос и предложение точно совпадают, может показаться на практике маловероятным, но, во-первых, открытые модели просто преобразуются в закрытую. Для этого достаточно при  $A > B$  ввести фиктивного получателя с объемом спроса  $A - B$ , а при  $A < B$  ввести фиктивного поставщика с объемом поставки  $B - A$ . Если тарифы фиктивных перевозок считать нулевыми, то результаты интерпретируются так: поставки к фиктивному получателю – грузы, оставшиеся у поставщика; поставки от фиктивного поставщика – недовыполнение заказа получателя.

Во-вторых, только к закрытой модели применим специальный алгоритм решения транспортной задачи. Поэтому далее будем рассматривать закрытую модель вида:

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min ,$$

$$\sum_{j=1}^n x_{ij} = a_i, \quad i = \overline{1, m} \quad \text{– все возможности поставщиков использованы,}$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = \overline{1, n} \quad \text{– запросы получателей полностью выполнены,}$$

$$x_{ij} \geq 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Очевидное утверждение о том, что закрытая транспортная задача всегда имеет решение, можно доказать, если принять:

$$x_{ij} = \frac{a_i b_j}{A}, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Действительно, суммируя по всем столбцам и учитывая, что  $A = B$ , получим:

$$\sum_{j=1}^n x_{ij} = \sum_{j=1}^n \frac{a_i b_j}{A} = \frac{a_i}{A} \sum_{j=1}^n b_j = \frac{B}{A} a_i = a_i, \quad i = \overline{1, m} .$$

Суммируя по строкам, получим ограничения-равенства для получателей.

Такой план является допустимым, но отнюдь не оптимальным. Для выполнения оптимизации условия задачи удобно представить в виде транспортной таблицы. При ручном расчете в каждую клетку табл. 4.1 заносят как тариф, так и объем перевозки.

Таблица 4.1

Поставщики	Получатели	Предложение
------------	------------	-------------

	1	2	...	$j$	...	$n$	поставщиков
1	$c_{11}$ $x_{11}$	$c_{12}$ $x_{12}$	...	$c_{1j}$ $x_{1j}$	...	$c_{1n}$ $x_{1n}$	$a_1$
2	$c_{21}$ $x_{21}$	$c_{22}$ $x_{22}$	...	$c_{2j}$ $x_{2j}$	...	$c_{2n}$ $x_{2n}$	$a_2$
...	...	...	...	...	...	...	...
$i$	$c_{i1}$ $x_{i1}$	$c_{i2}$ $x_{i2}$	...	$c_{ij}$ $x_{ij}$	...	$c_{in}$ $x_{in}$	$a_i$
...	...	...	...	...	...	...	...
$m$	$c_{m1}$ $x_{ml}$	$c_{m2}$ $x_{m2}$	...	$c_{mj}$ $x_{mj}$	...	$c_{mn}$ $x_{mn}$	$a_m$
Спрос получателей	$b_1$	$b_2$	...	$b_j$	...	$b_n$	$\sum_{i=1}^m a_i$ $\sum_{j=1}^n b_i$

## 4.2. Построение начального опорного плана

Закрытая транспортная модель содержит  $m+n$  ограничений в виде равенств. Поскольку суммы ограничений для поставщиков и для получателей равны, то одно (любое) из этих равенств избыточно.

Все остальные равенства линейно независимы, удаление любого эквивалентно исключению из модели соответствующего получателя или поставщика. Поскольку независимых равенств  $m+n-1$ , то базисный опорный план должен содержать такое же число независимых переменных.

Специальная структура транспортной модели (каждая переменная входит точно в два уравнения, все коэффициенты при переменных равны единице) позволяет построить опорный план без использования искусственных переменных, которые вводились при применении симплекс-метода. Для этого есть целый ряд методов, которые отличаются по степени приближения к оптимальному решению, что неизбежно влечет их усложнение. Хороший компромисс обеспечивает *метод минимального элемента*, который использует следующую последовательность действий:

1. Выберем в транспортной таблице ячейку с минимальной стоимостью перевозок. Если таких ячеек несколько, то выберем любую из них.

2. Переменной в выбранной ячейке присвоим максимально возможное значение, допускаемое ограничениями на спрос и предложение.

3. Корректируем значения спроса и предложения, уменьшая их на величину значения занесенной переменной. При этом одно из этих значений обязательно будет равно нулю.

4. Вычеркнем строку (или столбец) с обнуленной величиной ограничения, чтобы не присваивать значения остальным переменным. Если одновременно обнуляются значения и спроса и предложения, то вычеркивается только что-то одно.

5. Если осталось больше одной ячейки, то возвращаемся к первому шагу. Иначе заносим в оставшуюся ячейку еще нераспределенный груз, для которого спрос и предложение всегда будут совпадать. Построение опорного плана закончено.

Рассмотрим применение этого метода на примере (табл. 4.2). Пусть возможности поставщиков составляют 50, 60 и 70 ед. груза, а спрос получателей задан как 90, 30 и 40 ед. Поскольку возможные поставки превышают спрос на 20 ед., введем фиктивного получателя с таким объемом спроса. Эти данные вместе со значениями стоимостей перевозок занесем в таблицу. Номерами ее строк и столбцов будем считать порядковые номера поставщиков и получателей.

Таблица 4.2

Предложение поставщиков	Спрос получателей			
	90	30	40	20
50	8	9	2 ~	0
60	5	3	1 <b>40</b>	0
70	7	6	4 ~	0

Для фиктивного получателя транспортные затраты нулевые, так как груз остается у поставщика. Поэтому при выборе маршрута с минимальной стоимостью не будем принимать его во внимание. Тогда наименьшую стоимость имеет ячейка (2, 3). С учетом ограничений для второй строки и третьего столбца записываем в нее значение объема перевозки 40 ед. и корректируем предложение и спрос. Для третьего получателя спрос удовлетворен, следовательно, третий столбец вычеркиваем. Следующей ячейкой с

минимальной стоимостью в незаполненной части таблицы будет ячейка (2, 2), и в нее занесем значение объема перевозок равное 20. Последовательность действий и конечный результат представлены в табл. 4.3.

Таблица 4.3

Предложение поставщиков	Спрос получателей				
	90	30	40	20	
50	8	9	2	0	
30	↑	~	~	20 →	
60	5	3	1	0	
~	20	40 •	~	~	
70	7	6	4	0	
60	10 ↓	~	~	~	

### 4.3. Метод потенциалов

Этот метод основан на соотношениях теории двойственности и состоит в том, что каждому поставщику ставится с соответствие числа (потенциалы)  $u_i$ , а каждому получателю –  $v_j$ , удовлетворяющие условиям:

$$u_i + v_j = c_{ij},$$

в тех клетках таблицы, которые вошли в опорный план. Таких клеток  $m+n$ , а потенциалов  $m+n-1$ . Потенциалы можно интерпретировать как выплаты за единицу груза, получаемые транспортной организацией как при погрузке, так и после доставки груза. Величины потенциалов могут быть любого знака, но выплаты за перевозку по тому опорному плану, для которого они определены, будут точно соответствовать заданным тарифам.

Для незаполненных клеток значение разности  $c_{ij} - (u_i + v_j)$  может оказаться любого знака. Если  $c_{ij} - (u_i + v_j) > 0$ , то включение такой клетки в базис только увеличит суммарные затраты. Следовательно, если во всех незаполненных клетках тарифы превышают суммы потенциалов, то план оптимален. Рассматриваемые разности являются оценками переменных, поэтому, если план еще не является оптимальным, то нужно выбрать для включения в базис ту клетку, для которой абсолютное значение отрицательной оценки  $c_{ij} - (u_i + v_j) < 0$  является наибольшим, и занести в нее максимально возможную величину перевозимого груза. При этом балансы по строкам и столбцам необходимо сохранить, а все переменные должны оставаться неотрицательными.

Выполнение описанной процедуры рассмотрим на примере (табл. 4.4). Сначала требуется определить 7 неизвестных потенциалов, имея 6 уравнений. Любому из них можно присвоить произвольное значение (например  $u_1 = 0$ ) и затем, считая, что в заполненных клетках  $u_i + v_j = c_{ij}$ , последовательно (порядок вычислений показан стрелками) найдем остальные потенциалы.

Таблица 4.4

Предложение поставщиков	Спрос получателей				$u_i$	$v_j$
	90	30	40	20		
50	30	8	9	2	0	0
60		5	3	1	0	-4
70	60	20	40			
	60	10	7	4	0	-1
	8	7	5	0		

Перспективной для ввода в базис является только переменная (1, 3), оценка которой равна  $c_{13} - u_1 + v_3 = 2 - 0 - 5 = -3$ . Занесем ее в правый нижний угол клетки и построим замкнутый цикл, который начинается и заканчивается в этой ячейке и состоит из последовательности вертикальных и горизонтальных отрезков (диагональные недопустимы), соединяющих заполненные клетки. Поскольку вводимая в базис переменная всегда может быть выражена через базисные переменные, то такой цикл

всегда существует и будет единственным. Его можно обходить как по часовой стрелке, так и против, что несущественно. Важно то, что если вводимую в базис переменную пометить знаком  $\oplus$  и изменять знак в каждой угловой точке цикла, то в любой строке и любом столбце каждому знаку  $\oplus$  будет соответствовать знак  $\ominus$  (табл. 4.5).

Таблица 4.5

Предложение поставщиков	Спрос получателей				0	0
	90	30	40	20		
50	8	9	2	-3	20	0
60	5	3	1	0	0	-4
70	7	6	4	0	0	-1
	8	7	5	0	$u_i$	$v_j$

Следовательно, если всем участвующим в цикле переменным дать некоторое приращение  $\Delta$  с соответствующими знаками, то все балансовые соотношения по строкам и столбцам будут сохранены. Чтобы объемы перевозок оставались неотрицательными, это приращение  $\Delta$  нужно взять равным минимальному по абсолютной величине значениюю переменной из числа помеченных знаком  $\Theta$ . В примере это ячейка (3, 2), поэтому  $\Delta = 10$ .

После изменения грузопотоков получим новое базисное решение, которое приведено в табл. 4.6. В него введена переменная  $x_{13}$  и исключена переменная  $x_{32}$ . После этой итерации суммарные затраты на перевозки изменятся на  $\Delta(c_{13} - (u_1 + v_3)) = 10(2 - 0 - 5) = 10(-3) = -30$ . Вычисление новых значений потенциалов производится аналогично.

Таблица 4.6

Предложение поставщиков	Спрос получателей				0	0
	90	30	40	20		
50	8	9	2	-3	20	0
60	5	3	1	0	0	-1
70	7	6	4	0	0	-1
	8	7	5	0	$u_i$	$v_j$

Новой вводимой в базис переменной будет  $x_{22}$ , имеющая оценку  $c_{22} - (u_2 + v_2) = 5 - (-1) - 8 = -2$ . Построенный для нее цикл позволяет изменить значения входящих в него переменных на  $\Delta = 20$ . Результаты приведены в табл. 4.7.

Таблица 4.7

Предложение поставщиков	Спрос получателей				0	0
	90	30	40	20		
	8	9	2	0	0	0
	8	7	5	0	0	0

<b>50</b>			<b>30</b>	<b>20</b>	
<b>60</b>	5	3	1	0	-1
<b>70</b>	<b>20</b>	<b>30</b>	<b>10</b>		
<b>70</b>	7	6	4	0	1
	<b>70</b>				
	6	4	2	0	$u_i$
					$v_j$

После вычисления потенциалов отрицательная оценка получится для переменной  $x_{34}$ , действительно  $c_{34} - (u_3 + v_4) = 0 - 1 - 0 = -1$  и  $\Delta = 10$ . Получим табл. 4.8.

Таблица 4.8

Предложение поставщиков	Спрос получателей				$u_i$
	<b>90</b>	<b>30</b>	<b>40</b>	<b>20</b>	
<b>50</b>	8	9	40	2	0
<b>60</b>	5	3	1	0	-2
<b>70</b>	<b>30</b>	<b>30</b>	<b>10</b>		
	60	7	6	4	0
					$v_j$
	7	5	2	0	

Теперь для небазисных переменных все оценки  $c_{ij} - (u_i + v_j) > 0$  неотрицательны, поэтому решение оптимально. Так как среди этих оценок нет нулевых то решение единственное.

Чтобы исключить ошибки при выполнении расчетов, целесообразно:

1. Проверять для каждой новой таблицы выполнение балансовых соотношений по строкам и столбцам.
2. После вычисления потенциалов убедиться, что их суммы для заполненных клеток действительно равны тарифам.
3. При записи новой таблицы необходимо сразу заносить в нее тарифы, чтобы лучше различать отдельные клетки. Величины спроса и предложения переписывать не обязательно.
4. Если балансовые соотношения выполняются, а потенциалы вычислить не удается, то это значит, что базис вырожденный и переменная с нулевым значением пропущена. Поскольку базис должен содержать  $m + n - 1$  переменную, то недостающую нулевую переменную следует ввести. Можно использовать любую клетку, для которой нельзя построить цикл (базисные переменные должны быть независимыми), но из числа допустимых вариантов предпочтительнее тот, где меньше величина тарифа.

#### 4.4. Транспортная модель с промежуточными пунктами

Часто кроме исходных и конечных пунктов перевозок используются промежуточные пункты для временного хранения грузов. При этом часть продукции может быть реализована непосредственно в этих пунктах, возможно также, что через отдельные конечные пункты осуществляются перевозки транзитом в другие конечные пункты. Такую модель можно преобразовать в обычную транспортную модель, если ввести буфер достаточно большой, чтобы вместить объем всего предложения или спроса. Будем называть транзитными все пункты, в которые груз может как ввозиться, так и вывозиться. Для каждого транзитного пункта (таким пунктам будут соответствовать одновременно и строка и столбец транспортной модели) увеличим и объем спроса, и объем предложения на величину буфера. Очевидно, что тариф перевозки из такого пункта в него же может быть только нулевым, а полученная переменная, которая может достигать величины буфера, выполняет только техническую роль. При таком построении модели транзитные перевозки будут использоваться только тогда, когда это необходимо или выгодно. Для исключения несуществующих маршрутов перевозок будем полагать, что тарифы для них равны достаточно большому числу  $M$ . Пример такой задачи приведен на рис. 4.1.

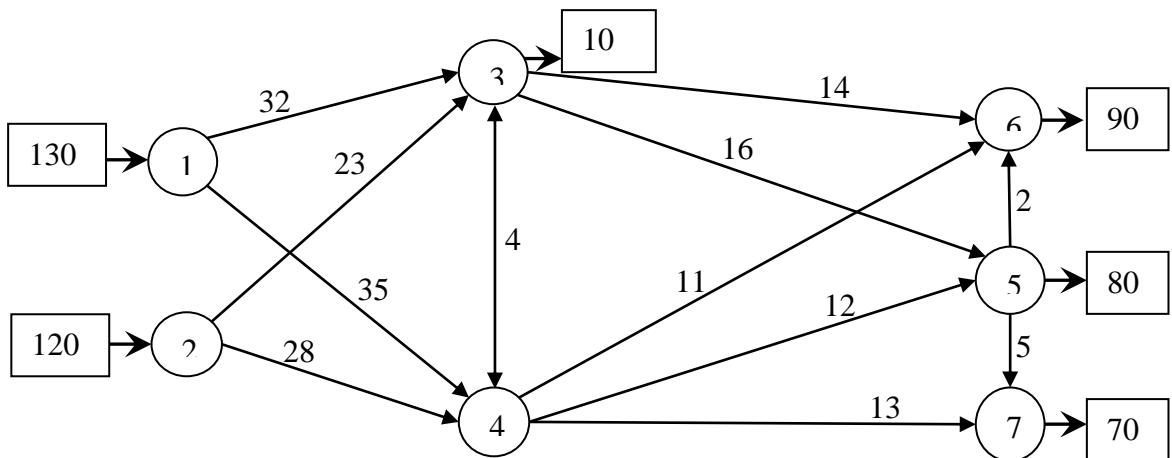


Рис. 4.1

Пункты ① и ② – это грузоотправители с объемами поставки 130 и 120 единиц, пункты ③ и ④ – транзитные (перевалочные) центры, причем в пункте ③ реализуется 10 единиц груза. Объемы реализации в пунктах ⑤ ⑥ и ⑦, а также затраты на транспортировку единицы груза по всем возможным маршрутам показаны. Заметим, что между пунктами ③ и ④ возможны перевозки в любом направлении, а через пункт ⑤ допустим транзит в пункты ⑥ и ⑦. Транспортную модель этой задачи представим в виде табл. 4.9.

Таблица 4.9

	③	④	⑤	⑥	⑦	
①	32	35	$M$	$M$	$M$	130
②	23	28	$M$	$M$	$M$	120
③	0	4	16	14	$M$	$B$
④	4	0	12	11	13	$B$
⑤	$M$	$M$	0	2	5	$B$
	$B+10$	$B$	$B+80$	90	70	

Для расчета примем объем буфера  $B = 250$ , что равно всему объему предложения (или спроса). Поскольку через пункты ③, ④ и ⑤ может осуществляться транзит, то, когда эти пункты рассматриваются как поставщики, им соответствует объем предложения, равный

буферу. Для этих же пунктов объем спроса увеличен на величину буфера. В результате баланс спроса и предложения сохраняется. Перемещению груза из пункта в него же соответствует нулевой тариф, а для исключения недопустимых маршрутов возьмем число  $M$  при численном расчете, например,  $M = 99$ . Если ни один из таких маршрутов не войдет в оптимальное решение, что всегда возможно при корректной постановке задачи, значит это число выбрано достаточно большим. Решение (полученное с использованием *Excel*) приведено на рис. 4.2. Показаны только используемые маршруты и оптимальные объемы перевозок.

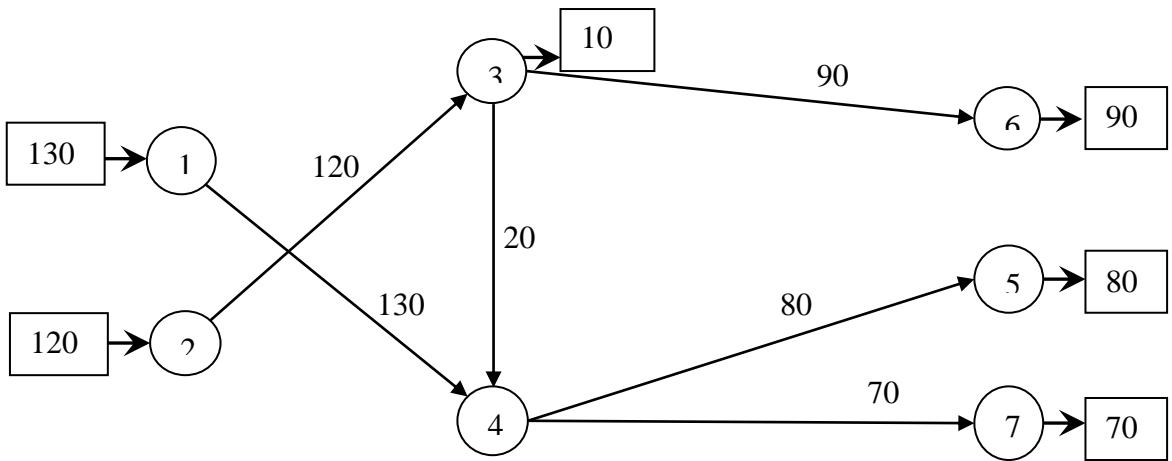


Рис. 4.2

Можно учесть и другие дополнительные условия. Например, пусть объем поставки груза от  $i$ -го поставщика к  $j$ -му получателю не должен превышать  $d_{ij}$ . Для этого  $j$ -й столбец заменим на два столбца и занесем в первый из них спрос  $d_{ij}$ , а во второй – оставшийся объем спроса. Тарифы в обоих столбцах одни и те же, но в  $i$ -ой строке в первом из двух введенных столбцов тариф используется фактический, а во второй заносится достаточно большое число  $M$ . При ручном расчете – это единственный разумный вариант решения задачи. При использовании пакета прикладных программ (*Excel*, *MathCad*) достаточно просто добавить условие, что  $x_{ij} \leq d_{ij}$ . Однако использование описанных выше табличной и графической форм представления задачи остается удобным и полезным.

#### 4.5. Задачи и модели оптимального размещения и концентрации производства

Размещение производительных сил – одна из важнейших задач экономики. Оптимальное размещение и концентрация производства важны не только с экономической точки зрения, но также с социальной, экономической, демографической.

Задача оптимального размещения и концентрации производства состоит в том, чтобы, исходя из условий наибольшей экономической эффективности, найти план размещения предприятий по производству определенный видов продукции в пределах некоторого региона.

Кроме того, необходимо определить оптимальный план поставок сырья и полуфабрикатов этим предприятиям. В результате решения такой задачи устанавливают, следует ли строить (расширять, модифицировать) рассматриваемые предприятия или нет. Если же нужно строить, то каковы должны быть оптимальные мощности этих предприятий.

Рассмотрим следующую экономическую ситуацию:

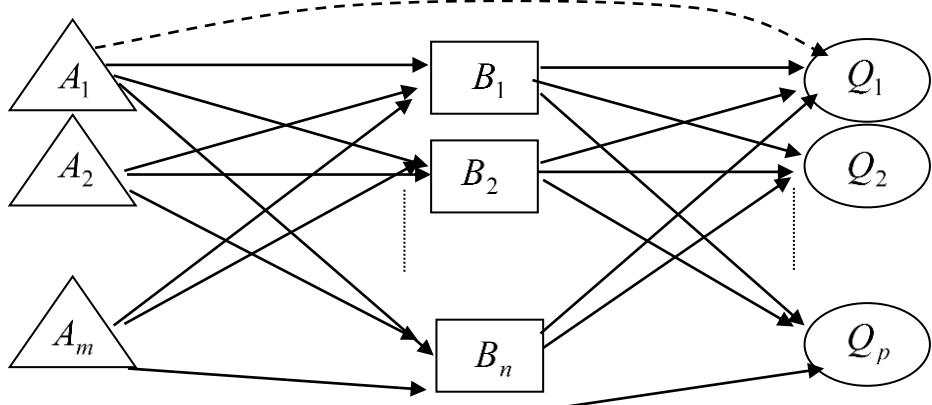
имеется экономическая система, состоящая из трех типов предприятий:

$A_i$  ( $i = \overline{1, m}$ ) – предприятия–заготовители и поставщики сырья;

$B_j$  ( $j = \overline{1, n}$ ) – предприятия–потребители сырья, они же поставщики каких-то полуфабрикатов;

$Q_k$  ( $k = \overline{1, p}$ ) – предприятия–потребители промежуточной продукции и сырья.

Схема перевозок в этом случае имеет вид:



Возможно включение не трех, а большего числа предприятий, а также создание комплексных предприятий, производящих промежуточную и конечную продукцию.

Среди рассматриваемых предприятий могут быть как действующие, не подлежащие реконструкции или планируемые к реконструкции, так и новые, которые предполагается построить.

Каждое из предприятий может выпускать несколько различных видов продукции и соответственно потреблять различную продукцию. Между поставщиками и потребителями могут существовать коммуникации (связи), пропускная способность которых может быть ограничена и не ограничена, и такие, между которыми связи нет.

Если предприятие существует и не подлежит реконструкции, то его мощности известны.

Если предполагается предприятие расширить, то задаются мощности после расширения. На мощность предприятия могут накладываться как односторонние так и двусторонние ограничения.

Кроме того, для таких задач, должны быть определены значения некоторых неуправляемых параметров, характеризующих экономические факторы. Например, затраты каждого из предприятий на изготовление единицы продукции каждого вида;

затраты на перевозку единицы груза, удельные капиталовложения в строительство, реконструкцию, расширение производства.

Решением задачи является оптимальный размер производства по отношению к выбранному критерию.

Приведенная задача является многоэтапной и многопродуктовой. Она решается достаточно сложно.

Если рассматривать лишь два звена производства, например,  $A_i$  ( $i = \overline{1, m}$ ) и  $B_j$  ( $j = \overline{1, n}$ ), то получаем одноэтапную задачу.

Если осуществляются перевозки однородного продукта, то имеем однопродуктовую задачу.

Условия, которым удовлетворяет модель, вытекают из экономического содержания задачи и обеспечивают адекватность модели реальной задаче. При этом важно, чтобы выполнялось условие начальной несбалансированности предложения и спроса, для того чтобы можно было решать задачу на размещение производства. Если имеет место баланс, то задача сводится лишь к оптимальному прикреплению поставщиков к потребителю (обычная ТЗ).

#### **4.6. Одноэтапная однопродуктовая модель с двусторонними ограничениями мощностей.**

Постановка задачи:

Имеется  $B_j$  ( $j = \overline{1, n}$ ) потребителей и спрос каждого потребителя известен и равен  $b_j$  ( $j = \overline{1, n}$ );  $Q_k$  ( $k = \overline{1, p}$ ) – поставщики.

Будем предполагать, что

1) часть из них действующие и реконструкции не подлежат (таких предприятий –  $s$  ).

2) вторая часть предприятий – действующие, но предполагаемые к реконструкции –  $h$  предприятий.

3) третья часть – новые предприятия, предполагаемые к постройке – их  $p - (s + h)$  предприятий.

Ограничения:

Для первых  $s$  предприятий мощности  $q_k$ ,  $k = \overline{1, s}$  заданы.

Для второй группы предприятий задаются два возможных предела объема производства:

$\underline{q}_k$ ,  $k = \overline{s+1, s+h}$  – нижний предел соответствует достигнутой мощности;

$\bar{q}_k$ ,  $k = \overline{s+1, s+h}$  – верхний предел соответствует максимальной мощности, которую предприятие может иметь после реконструкции.

Для третьей группы предприятий задается только верхний предел мощностей  $\bar{q}_k$ ,  $k = \overline{(s+h)+1, p}$ .

Кроме того, известны

$c_{kj}$  – удельные приведенные затраты при перевозке груза от  $k$ -го поставщика к  $j$ -му потребителю.

Может быть  $c_{kj} = c_k + t_{kj} + E_H K_k$ , где

$c_k$  – себестоимость единицы продукции на  $k$  предприятии;

$t_{kj}$  – затраты на перевозку единицы продукции от  $k$  к  $j$ -му предприятию;

$K_k$  – удельные капиталовложения в строительство  $k$ -го предприятия

$E_H$  – нормативный коэффициент сравнительной экономической эффективности капиталовложений.

Требуется найти оптимальные мощности предприятий и оптимальный план перевозок груза, который минимизирует расходы.

Математическая модель задачи имеет вид:

целевая функция

$$z = \sum_{k=1}^p \sum_{j=1}^n c_{kj} x_{kj} \rightarrow \min$$

где  $x_{kj}$  – количество груза, перевозимого от  $k$  поставщика  $j$ -му потребителю.

Ограничения:

1. Условие несбалансированности

$$\sum_{k=1}^s q_k + \sum_{k=s+1}^p \bar{q}_k > \sum_{j=1}^n b_j .$$

Если имеет место баланс, то задача сводится лишь к оптимальному прикреплению поставщиков к потребителю (обычной ТЗ).

2. Спрос потребителей должен быть удовлетворен

$$\sum_{k=1}^p x_{kj} = b_j , j = \overline{1, n} .$$

3. Сумма поставок от каждого действующего и нереконструируемого предприятия должна быть равна мощности этого предприятия

$$x_k = \sum_{j=1}^n x_{kj} = q_k , k = \overline{1, s} .$$

При введении фиктивного потребителя для действующих нереконструируемых предприятий поставки ему не должны планироваться. Ограничение пропускных способностей и отсутствие соответствующих связей математически записывается с помощью очень большой стоимости перевозки  $M$ .

4. Сумма поставок от каждого из реконструируемых предприятий по всем потребителям должно быть меньше верхнего предела, но больше нижнего предела

$$\underline{q}_k \leq x_k = \sum_{j=1}^n x_{kj} \leq \bar{q}_k , k = \overline{s+1, s+h} .$$

Для этих предприятий существуют дополнительные условия: часть продукции, соответствующая мощности предприятия до реконструкции **не может** поставляться фиктивному потребителю.

5. Сумма поставок от каждого предприятия, планируемого к новому строительству, должна быть не более верхнего предела мощностей:

$$x_k = \sum_{j=1}^n x_{kj} \leq \bar{q}_k , k = \overline{(s+h)+1, p}$$

6.  $x_{kj} \geq 0 , k = \overline{1, p} , j = \overline{1, n}$ , т.к.  $x_{kj}$  – количество груза.

Получена математическая модель открытой ТЗ имеющей двусторонние ограничения.

### Решение

1. Преобразуем эту задачу к закрытой ТЗ.

Должно быть выполнено условие баланса. Для этого вводим фиктивного потребителя  $B_{n+1}$  со спросом

$$b_{n+1} = \sum_{k=1}^s q_k + \sum_{k=s+1}^p \bar{q}_k - \sum_{j=1}^n b_j$$

2. Следует избавиться от двусторонних ограничений.

Каждое предприятие  $Q_k$ , для которого мощности ограничены с двух сторон, разбиваем условно на два предприятия:

$Q_k^{[1]}$  и  $Q_k^{[2]}$ , мощности которых  $\underline{q}_k$  и  $\bar{q}_k - \underline{q}_k$  соответственно,  
 $k = \overline{s+1, s+h}$

С экономической точки зрения предприятие  $Q_k^{[1]}$  можно рассматривать как обычное предприятие  $Q_k$  с мощностью  $\underline{q}_k$  до реконструкции, а предприятие  $Q_k^{[2]}$  как ту часть предприятия, которая будет достраиваться или реконструироваться.

Поставки  $x_{kj} = x_{kj}^{[1]} + x_{kj}^{[2]}$ .

Поставщики		Потребители и их спрос						
Действительные и условные	Мощности	$B_1$	$B_2$	...	$B_j$	...	$B_n$	$B_{n+1}$
		$b_1$	$b_2$		$b_j$		$b_n$	$b_{n+1}$
$Q_1$	$q_1$							M
$\vdots$								
$Q_s$	$q_s$							M
$Q_{s+1}$	$Q_{s+1}^{[1]}$	$\underline{q}_{s+1}$						M
	$Q_{s+1}^{[2]}$	$\bar{q}_{s+1} - \underline{q}_{s+1}$						0
$\vdots$								
$Q_{s+h}$	$Q_{s+h}^{[1]}$	$\underline{q}_{s+h}$						M
	$Q_{s+h}^{[2]}$	$\bar{q}_{s+h} - \underline{q}_{s+h}$						0
$Q_{s+h+1}$		$\bar{q}_{s+h+1}$						0
$\vdots$								
$Q_p$		$\bar{q}_p$						0

Поставки фиктивному потребителю  $c_{k,n+1} = M$ , для  $Q_k^{[1]}$  и  $c_{k,n+1} = 0$  для  $Q_k^{[2]}$ .

Соответствующим образом изменяться ограничения и целевая функция.

Так как для действующих предприятий поставки фиктивному потребителю запрещены, то удельные приведенные затраты для такого потребителя полагаем равными  $M$ .

0 – при поставках фиктивному потребителю от реконструируемых и строящихся предприятий.

Распределительная таблица для решения данной задачи (с двусторонними ограничениями на мощности) представлена ниже.

Замечание. При решении многопродуктовой задачи дополнительно вводится запрет на перевозку, если рассматриваемый вид продукции не пригоден потребителю.

#### 4.7. Двухэтапная транспортная задача.

В различных отраслях народного хозяйства (материально-техническое снабжение, торговля) грузы могут доставляться через промежуточные пункты. Допустим, имеется  $m$  ( $i = 1, m$ ) пунктов производства,  $n$  ( $j = 1, n$ ) пунктов потребления и  $p$  ( $k = 1, p$ ) — промежуточных баз. Как и в обычной транспортной задаче, обозначим через  $a_i$ ,  $b_j$  соответственно объемы поставок и потребления. Пусть  $d_k$  — мощность  $k$ -ой базы,  $c_{ik}$  и  $c_{kj}$  — соответственно стоимость перевозки единицы продукции от поставщиков на базы и с баз к потребителям. Тогда модель задачи примет вид

$$z = \sum_{i=1}^m \sum_{k=1}^p c_{ik} x_{ik} + \sum_{k=1}^p \sum_{j=1}^n c_{kj} x_{kj} \rightarrow \min,$$

при ограничениях:

$$\sum_{k=1}^p x_{ik} \leq a_i, i = \overline{1, m};$$

$$\sum_{i=1}^m x_{ik} \leq d_k, k = \overline{1, p};$$

$$\sum_{k=1}^p x_{kj} \geq b_j, j = \overline{1, n};$$

$$x_{ik} \geq 0, x_{kj} \geq 0.$$

Если суммарная пропускная мощность баз равна суммарной мощности поставщиков и суммарному спросу потребителей, т. е.

$$\sum_{k=1}^p d_k = \sum_{i=1}^m a_i = \sum_{j=1}^n b_j,$$

то пропускные емкости баз будут использованы полностью и, следовательно, схема перевозок с баз к потребителям не зависит от схемы перевозок от поставщиков на базы. В таких условиях задачу можно решать по частям. Оптимальный план можно составить объединением плана поставок от поставщиков к базам и плана поставок с баз к потребителям. Однако оптимальный план двухэтапной транспортной задачи, вообще говоря, отличен от плана, полученного объединением оптимальных планов решения транспортной задачи для каждого этапа в отдельности.

Двухэтапную транспортную задачу легко свести к классической транспортной задаче. Для этого базы будем считать одновременно поставщиками и потребителями. Для каждой базы в расширенной матрице (поставщики+базы) — (потребители+базы) отведем строку и столбец. Тогда матрица тарифов будет состоять из четырех блоков (табл.).

В первом — левом верхнем блоке будем отражать связи поставщиков с базами, в четвертом — связи баз с потребителями. Второй — правый верхний блок показывает связи поставщиков с потребителями. Поскольку по условию задачи непосредственные перевозки от поставщиков к потребителям запрещены, то в этом блоке все тарифы считаются равными  $M$  (где  $M$  — большое число). Третий — левый нижний блок образуется по строкам и столбцам базами, имеет форму квадрата. Так как перевозки между базами запрещаются, то соответствующие показатели с также считаются равными  $M$ . В клетках третьего квадрата, в которых отражаются связи базы с самой собой, тарифы равны нулю. Поставки в этих клетках показывают величину неиспользованной мощности базы. Диагональ из нулевых тарифов, отражающая связи базы с самой собой, называется **фиктивной**.

Таблица 4.10

	$d_1$	$\dots$	$d_p$	$b_1$	$\dots$	$b_n$
$a_1$	<b>I</b> $(c_{ik})_{m \times p}$			<b>II</b> M		
$\vdots$						
$a_m$						
$d_1$	0	<b>III</b>	M	<b>IV</b> $(c_{kj})_{p \times n}$		
$\vdots$	$\vdots$	$\dots$	$\vdots$			
$d_p$	M	$\dots$	0			

Решение двухэтапной транспортной задачи имеет некоторые особенности. Основная из них — некоторое изменение нахождения базисного решения. Вначале необходимо распределить поставки в одном из блоков (первом или четвертом). Затем заполняется фиктивная диагональ и только потом распределяются поставки в другом блоке (четвертом или первом). Вторая особенность заключается в том, что если цикл пересчета проходит через фиктивную диагональ, то он обязательно проходит через нее дважды; одна вершина цикла, находящаяся на диагонали, будет всегда положительной, а другая — отрицательной.

#### 4.8. Транспортные задачи с ограничениями или запрещениями перевозок.

При нахождении решения ряда конкретных транспортных задач часто бывает необходимо учитывать дополнительные ограничения, которые не встречаются при рассмотрении простых вариантов транспортных задач. Остановимся подробнее на некоторых возможных усложнениях в постановках транспортных задач.

1. При некоторых реальных условиях перевозки груза из определенного пункта отправления  $A_i$  ( $i = \overline{1, m}$ ) в пункт назначения  $B_j$  ( $j = \overline{1, n}$ ) не могут быть осуществлены. Для определения оптимальных планов таких задач предполагают, что тариф перевозки единицы груза из пункта  $A_i$  в пункт  $B_j$  является сколь угодно большой величиной  $M$ , и при этом условии известными методами находят решение новой транспортной задачи. При таком предположении исключается возможность при оптимальном плане транспортной задачи перевозить груз из пункта  $A_i$  в пункт  $B_j$ . Такой подход к нахождению решения транспортной задачи называют **запрещением перевозок** или **блокированием** соответствующей клетки таблицы данных задачи.

2. В отдельных транспортных задачах дополнительным условием является обеспечение перевозки по соответствующим маршрутам определенного количества груза. Пусть, например, из пункта отправления  $A_i$  в пункт назначения  $B_j$  требуется обязательно перевезти  $d_{ij}$  единиц груза. Тогда в клетку таблицы данных транспортной задачи, находящуюся на пересечении строки  $A_i$  и столбца  $B_j$ , записывают указанное число  $d_{ij}$  и в дальнейшем эту клетку считают свободной со сколь угодно большим тарифом перевозок  $M$ . Для полученной таким образом новой транспортной задачи находят оптимальный план, который определяет оптимальный план исходной задачи.

3. Пусть при решении транспортной задачи требуется ограничить перевозки от поставщика с номером  $l$  к потребителю с номером  $k$ . Возможны ограничения двух типов: 1)  $x_{lk} \geq a$ ; 2)  $x_{lk} \leq b$ , где  $a$  и  $b$  — постоянные величины.

1) Если  $x_{lk} \geq a$ , то необходимо прежде, чем решать задачу, сократить (уменьшить) запасы  $l$ -го поставщика и запросы  $k$ -го потребителя на величину  $a$  (зарезервировать перевозку  $x_{lk} = a$ ). После решения задачи в оптимальном решении следует увеличить объем перевозки  $x_{lk}$  на величину  $a$ .

Если  $x_{lk} \leq b$  то необходимо вместо  $k$ -го потребителя с запросами  $b_k$  ввести двух других потребителей. Один из них с номером  $k'$  должен иметь запросы  $b_{k'} = b$ , а другой с номером  $n+1$  — запросы  $b_{n+1} = b_k - b$ . Стоимости перевозок для этих потребителей остаются прежними, за исключением стоимости  $c_{l(n+1)}$ , которая принимается равной сколь угодно большому числу  $M$ . После получения оптимального решения величины грузов, перевозимых к  $(n+1)$ -му потребителю, прибавляются к величинам перевозок  $k$ -го потребителя. Так как  $c_{l(n+1)} = M$  самая большая стоимость перевозки, то в оптимальном решении клетка с номером  $(l, n+1)$  останется пустой,  $x_{l(n+1)} = 0$  и объем перевозки  $x_{lk}$  не превзойдет  $b$ .

## 4.9. Оптимальные назначения

Алгоритм и методы решения транспортной задачи могут быть использованы при решении экономических задач:

- сокращение производства с учетом суммарных расходов на изготовление и транспортировку продукции. Может оказаться экономически более выгодным доставлять сырье из отдаленного источника, но при меньшей себестоимости.

- увеличение производительности автомобильного транспорта за счет минимизации порожнего пробега;

- оптимальное распределение посевных площадей между сельхоз культурами;

- оптимальные назначения:

Пусть имеются  $m$  лиц  $A_i$  ( $i = \overline{1, m}$ ), которые могут выполнять  $B_j$  ( $j = \overline{1, m}$ ) различных работ. Известна производительность  $c_{ij}$   $i$ -го лица при выполнении  $j$ -й работы. Необходимо определить кого и на какую работу следует назначить, чтобы добиться максимальной суммарной производительности при условии, что каждое лицо может быть назначено только на одну работу.

Математическая модель данной задачи имеет вид:

$$f = \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} \rightarrow \max$$

при ограничениях

$$\begin{cases} \sum_{j=1}^m x_{ij} = 1, & i = \overline{1, m}, \\ \sum_{i=1}^m x_{ij} = 1, & j = \overline{1, m}, \end{cases} \quad x_{ij} \geq 0, (i = \overline{1, m}, j = \overline{1, m}).$$

Умножая функцию  $f$  на  $-1$ , приводим задачу к транспортной, в которой объем запасов каждого поставщика и объем потребностей каждого потребителя равны единице.

Приведенная задача допускает следующее обобщение: имеется  $m$  групп лиц  $A_i$  ( $i = \overline{1, m}$ ) по  $a_i$  человек в каждой и  $n$  категорий работ  $B_j$  ( $j = \overline{1, n}$ ) по  $b_j$  единиц в каждой. Известна производительность  $c_{ij}$  лица  $i$ -й группы при выполнении  $j$ -ой категории работ. Необходимо определить сколько лиц, из какой группы и на какую категорию работ назначить, чтобы добиться максимальной суммарной производительности.

Математическая модель данной задачи имеет вид:

$$f = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max$$

при ограничениях

$$\begin{cases} \sum_{j=1}^n x_{ij} = a_i, & i = \overline{1, m}, \\ \sum_{i=1}^m x_{ij} = b_j, & j = \overline{1, n}, \end{cases} \quad x_{ij} \geq 0, (i = \overline{1, m}, j = \overline{1, n}).$$

В задаче общее число работ равно общему числу лиц, т.е.  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ . Если

$\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$ , то вводят фиктивную группу лиц, содержащую  $\sum_{j=1}^n b_j - \sum_{i=1}^m a_i$  человек;

если  $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$ , то используют фиктивную категорию работ, состоящую из

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

#### 4.10. Примеры

**Пример 1 (транспортная задача).** Четыре леспромхоза заготавливают пиломатериалы в объемах: 1 леспромхоз – 300 тыс. м<sup>3</sup>; 2 – 400 тыс. м<sup>3</sup>; 3 – 300 тыс. м<sup>3</sup>; 4 – 400 тыс. м<sup>3</sup>, всего 1400 тыс. м<sup>3</sup>. Эти пиломатериалы используют четыре лесопильных завода в объемах: 1 завод – 200 тыс. м<sup>3</sup>; 2 – 350 тыс. м<sup>3</sup>; 3 – 400 тыс. м<sup>3</sup>; 4 – 450 тыс. м<sup>3</sup>; всего 1400 тыс. м<sup>3</sup>. Матрица стоимости перевозок имеет следующий вид:

$$C = \begin{pmatrix} 2 & 3 & 5 & 2 \\ 4 & 2 & 1 & 3 \\ 3 & 4 & 3 & 5 \\ 1 & 2 & 2 & 4 \end{pmatrix}.$$

Составить оптимальный план перевозок, чтобы все пиломатериалы были вывезены, все потребности были удовлетворены и чтобы транспортные затраты были минимальными.

Все исходные данные запишем в виде таблицы транспортной задачи.

В правом верхнем углу клетки  $(i, j)$  дается цена перевозки единицы груза  $C_{ij}$ , которую назовем стоимостью клетки. Первоначальный план строим методом “минимальной стоимости”. В матрице транспортной задачи выбираем наименьший элемент. Он равен единице. Ему соответствуют клетки (4, 1) и (2, 3). Выбираем любую, например (4, 1), и записываем в ней число  $\min(200, 400) = 200$ ; исключаем из рассмотрения первый столбец. В клетку (2, 3) записываем  $\min(400, 400) = 400$  и исключаем из рассмотрения второй столбец и вторую строку. Минимальная цена для незаполненных клеток равна 2. Таких клеток будет две: (1, 4) и (4, 2). Записываем в клетку (1, 4)  $\min(300, 450) = 300$ , а в клетку (4, 2) –  $\min(300, 200) = 200$ . Остались незаполненными два столбца: второй и четвертый. Записываем оставшиеся нераспределенные объемы в клетку (3, 2) – 150 тыс. м<sup>3</sup> и в клетку (3, 4) – 150 тыс. м<sup>3</sup>.

Так как количество загруженных клеток  $6 < 4 + 4 - 1 = 7$ , то план вырожденный. Дополним его до невырожденного фиктивной клеткой (1, 3). Понятно, что эта клетка не составляет с другими занятymi клетками цикл. Таким образом, мы получили начальный невырожденный план.

Таблица 4.11

Леспромхозы $A_i$	Лесопильные заводы				Запасы $a_i$	Потен-циалы $U_i$
	$B_1$	$B_2$	$B_3$	$B_4$		
$A_1$	2	3	0 - 5	300 + 2	300	$U_1 = 0$
$A_2$	4	2	400 1		400	$U_2 = -4$
$A_3$	3	150	4 + 3	150 - 5	300	$U_3 = 3$

$A_4$	200	1	200	2	2	4	400	$U_4 = 1$
Потребности $b_j$	200		350		400	450	1400	
Потенциалы $V_j$	$V_1=0$		$V_2=1$		$V_3=5$	$V_4=2$		

Теперь найдем потенциалы. По загруженным клеткам записываем систему уравнений

$$\begin{aligned} U_1 + V_3 &= 5; & U_3 + V_4 &= 5; \\ U_3 + V_2 &= 4; & U_2 + V_3 &= 1; \\ U_4 + V_2 &= 2; & U_4 + V_1 &= 1. \\ U_1 + V_4 &= 2; \end{aligned}$$

Решая эту систему, при условии  $U_1=0$  находим потенциалы  $U_i$ ,  $V_j$  и записываем их в соответствующие столбец и строку таблицы.

Проверим на оптимальность полученный план:

$$\begin{aligned} \Delta_{11} &= C_{12} - (U_1 + V_1) = 2 - (0 + 0) = 2 > 0; \\ \Delta_{12} &= C_{12} - (U_1 + V_2) = 3 - (0 + 1) = 2 > 0; \\ \Delta_{21} &= C_{21} - (U_2 + V_1) = 4 - (-4 + 0) = 8 > 0; \\ \Delta_{22} &= C_{22} - (U_2 + V_2) = 2 - (-4 + 1) = 5 > 0; \\ \Delta_{24} &= C_{24} - (U_2 + V_4) = 3 - (-4 + 2) + 2 = 5 > 0; \\ \Delta_{31} &= C_{31} - (U_3 + V_1) = 3 - (3 + 0) = 0 \geq 0; \\ \Delta_{33} &= C_{33} - (U_3 + V_3) = 3 - (3 + 5) = -5 < 0; \\ \Delta_{43} &= C_{43} - (U_4 + V_3) = 3 - (1 + 5) = -3 < 0; \\ \Delta_{44} &= C_{44} - (U_4 + V_4) = 4 - (1 + 2) = 1 > 0. \end{aligned}$$

План не является оптимальным, потому что  $\Delta_{33} < 0$  и  $\Delta_{43} < 0$ .

$\min(\Delta_{33}; \Delta_{34}) = \min(-5; -3) = -5$  достигается в клетке (3, 3). Для клетки (3, 3) строим цикл с вершинами в клетках (3, 3), (1, 3), (1, 4), (3, 4). Расставим знаки "+" в клетках (3, 3) и (1, 4) и "-" в клетках (1, 3) и (3, 4).

Найдем минимальную перевозку  $\Theta$  среди клеток с отрицательными знаками и  $\Theta = \min(x_{13}, x_{34}) = \min(0, 150) = 0$ . Значение  $\Theta$  запишем в перспективную клетку (3, 3), прибавим к перевозкам в клетку (1, 4) и отнимаем от перевозок клеток (1, 3) и (3, 4). Таким образом, мы получили новую таблицу планирования.

Отметим, что поскольку в цикл с отрицательными вершинами вошла клетка с фиктивными перевозками, то новая матрица планирования практически не изменилась, за исключением того, что клеткой с фиктивными перевозками стала клетка (3, 3) вместо клетки (1, 3).

Новый план проверяем на оптимальность. Находим потенциалы из системы уравнений:

$$\begin{aligned}
U_1 + V_4 &= 2; & U_3 + V_4 &= 5; \\
U_2 + V_3 &= 1; & U_4 + V_1 &= 1; \\
U_3 + V_2 &= 4; & U_4 + V_2 &= 2; \\
U_3 + V_3 &= 3; & U_1 &= 0
\end{aligned}$$

и записываем в табл. 4.12

Таблица 4.12

Леспромхозы $A_i$	Лесопильные заводы				Запасы $a_i$	Потенциалы $U_i$	
	$B_1$	$B_2$	$B_3$	$B_4$			
$A_1$	2	3	5	300	2	300	$U_1 = 0$
$A_2$	4	2	400	1	3	400	$U_2 = 1$
$A_3$	3	150	4	0	3	150	$U_3 = 3$
$A_4$	200	1	200	2	2	4	$U_4 = 1$
Потребности $b_j$	200	350	400	450		1400	
Потенциалы $V_j$	$V_1 = 0$	$V_2 = 1$	$V_3 = 0$	$V_4 = 2$			

Вычисляем  $\Delta_{ij}$  для незагруженных клеток.

$$\Delta_{12} = 3 - (1+0) = 2 > 0; \quad \Delta_{31} = 3 - (3+0) = 0 \geq 0;$$

$$\Delta_{13} = 5 - (0+0) = 5 > 0; \quad \Delta_{43} = 3 - (1+0) = 2 > 0;$$

$$\Delta_{21} = 4 - (1+0) = 3 > 0; \quad \Delta_{44} = 4 - (1+2) = 1 > 0.$$

$$\Delta_{24} = 3 - (1+2) = 0 \geq 0;$$

Поскольку все  $\Delta_{ij} \geq 0$ , то план оптимален.

Матрица оптимальных перевозок имеет вид

$$X_{onm} = \begin{pmatrix} 0 & 0 & 0 & 300 \\ 0 & 0 & 400 & 0 \\ 0 & 150 & 0 & 150 \\ 200 & 200 & 0 & 0 \end{pmatrix}.$$

Транспортные затраты составляют

$$Z = 200 \cdot 1 + 150 \cdot 4 + 200 \cdot 2 + 400 \cdot 1 + 300 \cdot 2 + 150 \cdot 5 = 2950 \text{ ден. ед.}$$

**Пример 2 (двуэтапная транспортная задача).** В некотором районе имеется два маслодельных завода  $A_1, A_2$ . Сливочное масло поступает сначала в холодильники  $D_1, D_2, D_3$ , а из них — в пункты потребления  $B_1, B_2, B_3, B_4$ . Возможности маслодельных заводов, мощность холодильников, запросы потребителей и соответствующие тарифы представлены в табл. 4.13

Таблица 4.13

Поставки производства $a_i$ , т	Объем потребления $b_j$ , т	Пропускная способность ППП $d_k$ , т	Тариф $c_{ik}$	Тариф $c_{kj}$
350	150	240	20, 23, 16	12, 19, 20, 13
450	250	500	15, 10, 24	10, 15, 14, 12
	175			
	225	260		16, 21, 25, 11

**Решение.** Так как  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j = 800$ , то можно решать двухэтапную ТЗ задачу (табл. 4.14). (При  $\sum_{i=1}^m a_i \neq \sum_{j=1}^n b_j$ , следует ввести фиктивного поставщика или фиктивного потребителя.)

Таблица 4.14

	240	500	260	150	250	175	225
350	20	23	16	M	M	M	M
	<b>75</b>	<b>50</b>	<b>225</b>				
450	15	10	24	M	M	M	M
		<b>450</b>					
240	0	M	M	12	19	20	13
		<b>165</b>			<b>75</b>		
500	M	0	M	150	10	15	14
					<b>175</b>	<b>175</b>	12
260	M	M	0	16	21	25	11
			<b>35</b>				<b>225</b>

Начальный опорный план, приведенный в табл., найден способом минимального элемента, причем распределение начато с четвертого блока. Затем заполнялась фиктивная диагональ и, наконец, первый блок.

Оптимальный план (не единственный) представлен в табл.:  $z_{\min} = 10600 + 10425 = 21025$  руб. Интересно отметить, что если решать задачу в два этапа, т. е. находить сначала оптимальный план прикрепления поставщиков к холодильникам, а затем задачу оптимального прикрепления холодильников к потребителям, то суммарные потери составят  $z_{\min} = 10460 + 14810 = 25350$  руб., т. е. возрастают, хотя на первом этапе издержки уменьшаются.

Таблица 4.15

	240	500	260	150	250	175	225
350	20	23	16	M	M	M	M
	<b>125</b>		<b>225</b>				
450	15	10	24	M	M	M	M
		<b>450</b>					

240	0	M	M	12	19	20	13
500	M	0	M	10	15	14	12
260	M	M	0	16	21	25	11

**115      50      35      250      175      225**

**Пример 3.** (одноэтапная многопродуктовая задача размещения и концентрации производства)

В состав леспромхоза входят два лесопункта, каждый из которых осуществляет отгрузку круглых лесоматериалов. Поскольку леспромхоз производит несколько видов сортиментов, то задача является многопродуктовой. Кроме отгрузки сортиментов на сторону, в составе первого лесопункта организовано производство пиломатериалов и технологической щепы из дров. При решении задачи следует учитывать, что в перерабатывающих производствах может быть использовано сырье обоих лесопунктов.

В задаче известны: минимально и максимально возможные объемы производства отдельных сортиментов в разрезе лесопунктов; направления использования заготовленных сортиментов и потребности в них по каждому из этих направлений; текущие затраты на заготовку и транспортировку сырья.

В результате решения задачи требуется определить оптимальные объемы производства сортиментов конкретного вида в разрезе лесопунктов, а также оптимальные размеры каждого из лесопунктов.

Исходные данные, необходимые для решения задачи, представлены в табл.

Таблица 4.16

Мощности лесопунктов по выпуску отдельных сортиментов, тыс. м<sup>3</sup>

Лесопункт 1			Лесопункт 2		
Пиловочник	Тех. сырье	Дрова	Пиловочник	Тех. сырье	Дрова
50/60	10/15	35/50	23/30	4/10	15/25

Везде: числитель – нижний предел производства; знаменатель – верхний предел производства.

Таблица 4.17

Потребности в сортиментах по направлениям использования, тыс. м<sup>3</sup>

Переработано внутри предприятия (на первом лесопункте)		Поставлено на сторону			Использовано дров на топливные нужды
пиловочника	дров	пиловочника	дров	тех. сырья	
14	20	60	20	15	15

Таблица 4.18

Затраты на производство и транспортировку сырья, ден. ед./тыс. м<sup>3</sup>

Затраты на производство		Затраты на транспортировку сырья	
Пиловочника	Тех. сырья	Дров	Затраты на перевозку сырья

Лесопункт № 1	Лесопункт № 2	Лесопункт № 1	Лесопункт № 2	Лесопункт № 1	Лесопункт № 2	
25	23	6	10	3	7	4

Решение данной задачи начинается с представления исходных данных в виде таблицы, в которой отражены все допустимые перевозки и их стоимость.

Составим рабочую таблицу. Данная задача является транспортной многопродуктовой задачей с двусторонними ограничениями на мощности поставщиков. Поэтому построение транспортной таблицы начинается с определения количества поставщиков и их мощностей. Фактически в задаче имеется три поставщика: поставщики пиловочника, дров и технического сырья, но так как каждый из них имеет двусторонние ограничения на мощности, то будем рассматривать каждого из поставщиков как двух. Первый из них имеет мощность, равную нижней границе мощности исходного поставщика, а второй – мощность, равную разнице между нижней и верхней границей. Так, например, для пиловочника имеем  $L_1^{11}$  – первый поставщик с мощностью 50 ед.,  $L_1^{21}$  – второй поставщик с мощностью  $60 - 50 = 10$  ед. (данные взяты из таблицы). Аналогично производится подсчет мощностей остальных поставщиков (для первого и второго лесопунктов). Таким образом, имеем 12 поставщиков с соответствующими мощностями.

Далее нужно определить потребителей и их потребности. Используя условие задачи, получим 6 потребителей с соответствующими потребностями.

Получили открытую транспортную задачу. Для решения этой задачи нужно ввести фиктивных потребителей, так как имеющиеся мощности производителей больше потребностей потребителей ( $190 > 144$ ). Введем трех фиктивных потребителей (по количеству используемых сортиментов):  $B_4^1$ ,  $B_4^2$ ,  $B_4^3$ . Определим их потребности. Для пиловочника имеем: мощности =  $50 + 10 + 23 + 7 = 90$ , а потребности =  $14 + 60 = 76$ . Тогда  $90 - 76 = 14$ . Таким образом, мощности фиктивного поставщика по пиловочнику  $b_4^1 = 16$ . Аналогично рассчитываются мощности остальных фиктивных потребителей (по дровам и техническому сырью).

В транспортной задаче тарифы полагаются равными некоторой большой величине  $M$  в тех случаях, когда вид продукции, производимый в пункте производства, непригоден в данном пункте потребления. В остальных случаях данные берутся из таблицы. Затраты на «поставку» сырья от любого из «поставщиков» фиктивному «потребителю» принимаются равными нулю не для всех пунктов производства, а лишь для тех, по которым объем производства принимается равным разности между установленными верхними и нижними границами объема производства. В этом случае поставки фиктивному потребителю будут характеризовать излишки производственных мощностей.

Транспортная таблица составлена. Далее задачу можно решать по этой таблице, а можно воспользоваться тем фактом, что при переработке сортименты не перемешиваются и поэтому исходную задачу можно разбить на три транспортные задачи (по количеству сортиментов) меньшего размера.

Лесопункты	Наим. продукта	Мощности	Потребности								
			В1 (внутри предприятия)		В2 – поставки за пределы предприятия			B3 (топл.)	В4 – фиктивные		
			$B_1^1$ - пил	$B_1^2$ - др	$B_2^1$ - пил	$B_2^2$ - др	$B_2^3$ - тех. с	$B_3$	$B_4^1$ - пил	$B_4^2$ - др	$B_4^3$ - тех. с
			$b_1^1 = 14$	$b_1^2 = 20$	$b_2^1 = 60$	$b_2^2 = 20$	$b_2^3 = 15$	$b_3 = 15$	$b_4^1 = 16$	$b_4^2 = 20$	$b_4^3 = 10$
Л1	Л <sub>1</sub> <sup>11</sup> - пил	50	25	M	25	M	M	M	M	M	M
	Л <sub>1</sub> <sup>12</sup> - др	35	M	3	M	3	M	3	M	M	M
	Л <sub>1</sub> <sup>13</sup> - тех. с	10	M	M	M	M	6	M	M	M	M
	Л <sub>1</sub> <sup>21</sup> - пил	$60 - 50 = 10$	25	M	25	M	M	M	0	M	M
	Л <sub>1</sub> <sup>22</sup> - др	$50 - 35 = 15$	M	3	M	3	M	3	M	0	M
	Л <sub>1</sub> <sup>23</sup> - тех. с	$15 - 10 = 5$	M	M	M	M	6	M	M	M	0
Л2	Л <sub>2</sub> <sup>11</sup> - пил	23	M	M	23+4	M	M	M	M	M	M
	Л <sub>2</sub> <sup>12</sup> - др	15	M	M	M	7+4	M	7+4	M	M	M
	Л <sub>2</sub> <sup>13</sup> - тех. с	4	M	M	M	M	10+4	M	M	M	M
	Л <sub>2</sub> <sup>21</sup> - пил	$30 - 23 = 7$	M	M	23+4	M	M	M	0	M	M
	Л <sub>2</sub> <sup>22</sup> - др	$25 - 15 = 10$	M	M	M	7+4	M	7+4	M	0	M
	Л <sub>2</sub> <sup>23</sup> - тех. с	$10 - 4 = 6$	M	M	M	M	10+4	M	M	M	0

Таблица 4.19. Пиловочник

Лесопункты		Мощности	<i>отребности</i>		
			$b_1^1 = 14$	$b_2^1 = 60$	$b_4^1 = 14$
Л1	$\Pi_1^{11}$	50	25	25	M
	$\Pi_1^{21}$	10	25	25	0
Л2	$\Pi_2^{11}$	23	M	27	M
	$\Pi_2^{21}$	7	M	27	0

Таблица 4.20. Дрова

Лесопункты		Мощности	<i>отребности</i>			
			$b_1^2 = 20$	$b_2^2 = 20$	$b_3^2 = 15$	$b_4^2 = 20$
Л1	$\Pi_1^{11}$	35	3	3	3	M
	$\Pi_1^{21}$	15	3	3	3	0
Л2	$\Pi_2^{11}$	15	M	11	11	M
	$\Pi_2^{21}$	10	M	11	11	0

Таблица 4.21. Тех. сырье

Лесопункты		Мощности	<i>отребности</i>	
			$b_2^3 = 15$	$b_4^3 = 10$
Л1	$\Pi_1^{11}$	10	6	M
	$\Pi_1^{21}$	5	6	0
Л2	$\Pi_2^{11}$	4	14	M
	$\Pi_2^{21}$	6	14	0

Получим три транспортные задачи закрытого типа (проверить, что эти задачи закрытого типа). Рассмотрим решение задачи. На рисунке введены: условие рассматриваемой задачи и матрица тарифов. Далее нужно ввести зависимости рассматриваемой задачи.

Решая задачи, представленные в таблицах, получаем

$$X_{\text{пил}}^{\text{опт}} = \begin{pmatrix} 13 & 37 & 0 \\ 1 & 0 & 9 \\ 0 & 23 & 0 \\ 0 & 0 & 7 \end{pmatrix}, \quad X_{\text{др}}^{\text{опт}} = \begin{pmatrix} 15 & 5 & 15 & 0 \\ 5 & 0 & 0 & 10 \\ 0 & 15 & 0 & 0 \\ 0 & 0 & 0 & 10 \end{pmatrix}, \quad X_{\text{тех. с}}^{\text{опт}} = \begin{pmatrix} 10 & 0 \\ 1 & 4 \\ 4 & 0 \\ 0 & 6 \end{pmatrix}.$$

Значения целевых функций  $F(x_{ij}^k)$ :  $F(X_{\text{пил}}^{\text{опт}}) = 1896$ ,  $F(X_{\text{др}}^{\text{опт}}) = 285$ ,  $F(X_{\text{тех. с}}^{\text{опт}}) = 122$ , что дает в сумме 2303 и совпадает с решением многопродуктовой задачи.

Объединяя объемы перевозок по каждому лесопункту, получаем

$$X_{\text{пил}}^{\text{опт}} = \begin{pmatrix} 14 & 37 & 9 \\ 0 & 23 & 7 \end{pmatrix}, \quad X_{\text{др}}^{\text{опт}} = \begin{pmatrix} 20 & 5 & 15 & 10 \\ 0 & 15 & 0 & 10 \end{pmatrix}, \quad X_{\text{тех. с}}^{\text{опт}} = \begin{pmatrix} 11 & 4 \\ 4 & 6 \end{pmatrix}.$$

Полученное решение говорит о том, что мощность первого лесопункта по пиловочнику должна быть равна  $14 + 37 = 51$  тыс. м<sup>3</sup>, из них 14 тыс. м<sup>3</sup> перерабатывается внутри предприятия, а 37 тыс. м<sup>3</sup> поставляется на сторону. Резервы производственной мощности для первого лесопункта составляют 9 тыс. м<sup>3</sup>.

Аналогичные выводы следует сделать по второму лесопункту и по остальным сортиментам: по тех. сырью и дровам.

**Пример 4 (транспортная задача с ограничениями перевозок).** Решить транспортную задачу, исходные данные которой приведены в табл., при дополнительных условиях: объем перевозки груза от 1-го поставщика 2-му потребителю должен быть не менее 100 единиц ( $x_{12} \geq 100$ ), а от 3-го 1-му не более 200 единиц ( $x_{31} \leq 200$ ).

Таблица 4.22

$a_i \backslash b_j$	500	400	300
200	1	5	6
300	2	6	7
500	3	7	8

**Решение.** Для того чтобы в оптимальном решении объем перевозки  $x_{12}$  был не менее 100 единиц, при решении задачи будем предполагать, что запасы 1-го поставщика  $a_1$  и запросы 2-го потребителя  $b_2$ , меньше фактических на 100 единиц. После получения оптимального решения объем перевозки  $x_{12}$ , увеличим на 100 единиц.

Для того чтобы удовлетворить требованию  $x_{31} \leq 200$ , вместо 1-го потребителя введем двух других. Один из них под прежним первым номером имеет запросы  $b_1 = 200$  единиц и прежние стоимости перевозок единиц груза. Другому присвоим четвертый номер. Его запросы равны  $b_4 = 500 - 200 = 300$  единиц и стоимости перевозок единиц груза те же, что и у 1-го потребителя, за исключением  $c_{34}$ , которую примем равной сколь угодно большому числу  $M$ , т.е.  $c_{34} = M$ . После нахождения оптимального решения задачи объемы перевозок для 4-го потребителя необходимо прибавить к

соответствующим объемам перевозок для 1-го потребителя.

В результате указанных преобразований таблица исходных данных задачи будет иметь вид, представленный в табл. 4.23

Таблица 4.23

$a_i \backslash b_j$	200	300	300	300
100	1	5	6	1
300	2	6	7	2
500	3	7	8	M

Далее проверяем выполнение необходимого и достаточного условия существования решения задачи. Находим суммарные запасы поставщиков и запросы потребителей:

$$a_1 + a_2 + a_3 = 100 + 300 + 500 = 900,$$

$$b_1 + b_2 + b_3 + b_4 = 200 + 300 + 300 + 300 = 1100.$$

Имеем открытую транспортную задачу. Вводим фиктивного поставщика с запасами  $a_4 = 1100 - 900 = 200$  (табл. 4.24).

Таблица 4.24

$a_i \backslash b_j$	200	300	300	300
100	1	5	6	1
300	2	6	7	2
500	3	7	8	M
200	0	0	0	0

Полученную задачу решаем методом потенциалов.

Последняя таблица имеет следующий вид:

Таблица 4.25

$a_i \backslash b_j$	200	300	300	300
100	1 <b>100</b>	5 0	6 0	1 0
300	2 <b>0</b>	6 0	7 0	2 <b>300</b>
500	3 <b>100</b>	7 <b>300</b>	8 <b>100</b>	M
200	0	0	0 <b>200</b>	0

В табл. записаны оценки для свободных клеток. Решение оптимальное, так как все оценки неположительные. Запишем оптимальное решение исходной задачи. Для этого увеличим объем перевозки  $x_{12}$  на 100 единиц и объединим объемы перевозок 4-го потребителя с объемами перевозок 1-го потребителя.

Получим

$$X^* = \begin{pmatrix} 100 & 100 & 0 \\ 300 & 0 & 0 \\ 100 & 300 & 100 \end{pmatrix}.$$

Вычислим значение целевой функции на этом решении:

$Z(X^*) = 100 \cdot 1 + 100 \cdot 5 + 300 \cdot 2 + 100 \cdot 3 + 300 \cdot 7 + 100 \cdot 8 = 4400$ . Заметим, что так как имеются нулевые оценки свободных клеток, то найденное решение не является единственным. Загружая любую из свободных клеток с нулевой оценкой, получим другое оптимальное решение, причем значение целевой функции на новом решении останется тем же.

Ответ:  $Z_{\min} = 4400$  при  $X^* = \begin{pmatrix} 100 & 100 & 0 \\ 300 & 0 & 0 \\ 100 & 300 & 100 \end{pmatrix}$ .

## 5.1. Введение

В процессе целенаправленной человеческой деятельности возникают ситуации, в которых интересы отдельных лиц (участников, групп, сторон) либо прямо противоположны (антагонистичны), либо, не будучи непримиримыми, все же не совпадают. Примерами таких ситуаций являются спортивные игры, арбитражные споры, военные учения, борьба между блоками избирателей за своих кандидатов, в международных отношениях -отстаивание интересов своего государства и т. д. Здесь каждый из участников сознательно стремится добиться наилучшего результата за счет другого участника. Подобного рода ситуации встречаются и в различных сферах производственной деятельности.

Для приведенных ситуаций (они называются конфликтными) характерно, что эффективность решений, принимаемых в ходе конфликта каждой из сторон, существенно зависит от действий другой стороны. При этом ни одна из сторон не может полностью контролировать положение, так как и той и другой стороне решения приходится принимать в условиях неопределенности. Так, при определении объема выпуска продукции на одном предприятии нельзя не учитывать размеров выпуска аналогичной продукции на других предприятиях. В реальных условиях нередко возникают ситуации, в которых антагонизм отсутствует, но существуют противоположные тенденции. Например, для нормального функционирования производства, с одной стороны, необходимо наличие запасов разнообразных ресурсов, но с другой, - стремление к чрезвычайному увеличению этих запасов вызывает дополнительные затраты по их содержанию и хранению. В приведенных примерах конфликтные ситуации возникают в результате сознательной деятельности людей. Однако на практике встречаются неопределенности, которые порождаются не сознательным противодействием другой стороны, а недостаточной информированностью об условиях проведения планируемой операции.

Раздел математики, изучающий конфликтные ситуации на основе их математических моделей, называется теорией игр. Таким образом, **теория игр** - это математическая теория конфликтных ситуаций, разрабатывающая рекомендации по наиболее рациональному образу действий каждого из участников в ходе конфликтной ситуации, т. е. таких действий, которые обеспечивали бы ему наилучший результат. Игровую схему можно придать многим ситуациям в экономике. Здесь выигрышем могут быть эффективность использования дефицитных ресурсов, производственных фондов, величина прибыли, себестоимость и т. д.

Возникновение теории игр относится к 1944 г., когда вышла монография Дж. фон Неймана и О. Моргенштерна «Теория игр и экономическое поведение», содержащая в основном экономические примеры. В настоящее время теория игр бурно развивается.

Необходимо подчеркнуть, что методы и рекомендации теории игр разрабатываются применительно к таким специфическим конфликтным ситуациям, которые обладают свойством **многократной повторяемости**. Если конфликтная ситуация реализуется однократно или ограниченное число раз, то рекомендации теории игр теряют смысл.

Чтобы проанализировать конфликтную ситуацию по ее математической модели, ситуацию необходимо упростить, учитывая лишь важнейшие факторы, существенно влияющие на ход конфликта. Игра – это упрощенная математическая модель конфликтной ситуации, отличающаяся от реального конфликта тем, что ведется по определенным правилам. Поэтому можно сказать, что **игра** – это совокупность правил, определяющих возможные действия (чистые стратегии) участников игры. Суть игры в том, что каждый из участников принимает такие решения в развивающейся конфликтной ситуации, которые, как он полагает, могут обеспечить ему наилучший исход. **Исход игры** – это значение некоторой функции, которая называется **функцией выигрыша (платежной функцией)**. Она может задаваться либо аналитическим выражением, либо таблично (матрицей). Далее

будем рассматривать только такие игры, в которых выигрыш выражается количественно: стоимостью, баллами и т. д.

Величина выигрыша зависит от стратегии, применяемой игроком. **Стратегия** – это совокупность правил, однозначно определяющих последовательность действий игрока в каждой конкретной ситуации, складывающейся в процессе игры. **Оптимальной стратегией** называется стратегия, которая обеспечивает игроку наилучший результат. Причем предполагается, что противник столь же разумен, как и игрок, и делает все, чтобы добиться своей цели.

В реальных конфликтах можно прогнозировать образ мышления и вероятное поведение противника, что является предметом теории рефлексивных игр, которая стала развиваться в последние годы. Предположение о разумности противника позволяет найти наиболее осторожную стратегию, без учета которой вряд ли стоит принимать окончательное решение.

Всякая игра состоит из отдельных партий. **Партией** называют каждый вариант реализации игры определенным образом. В свою очередь, в партии игроки совершают конкретные ходы. **Ход** – это выбор и реализация игроком одного из допустимых вариантов поведения. Ходы бывают **личные** и **случайные**. При личном ходе игрок самостоятельно и осознанно выбирает и реализует ту или иную чистую стратегию. Например, в шахматах каждый ход является личным. При случайном ходе выбор чистой стратегии производится с использованием какого-либо механизма случайного выбора, например, с применением таблицы случайных чисел.

Классифицировать игры можно по разным признакам. Различают, например, игры по количеству игроков. В игре может участвовать любое конечное число игроков. Если при этом игроки объединяются, например, в две группы, преследующие противоположные цели, то имеет место игра двух "лиц" (парная игра). В зависимости от количества стратегий в игре они делятся на **конечные** или **бесконечные**. Примером игры с бесконечным числом стратегий может быть ситуация «продавец – покупатель», когда как цена, так и количество товара могут быть названы любыми. В зависимости от взаимоотношений участников различают игры **бескоалиционные** (участники не имеют права заключать соглашения), или **некооперативные**, и **коалиционные**, или кооперативные. Примером может служить ситуация, когда несколько производителей одного товара могут воздействовать на его рыночную цену.

По характеру выигрышной игры делятся на игры **с нулевой суммой** и **ненулевой суммой**. В первых – общий капитал игроков не меняется, а лишь перераспределяется в ходе игры, в связи с чем сумма выигрышной равна нулю (проигрыш принимается как отрицательный выигрыш). Так как имеется прямой конфликт интересов, то их также называют **антагонистическими**. Однако если возникает новая стоимость, которую можно разделить между игроками, то это игра с ненулевой суммой, т.е. сумма выигрышной отлична от нуля. Например, при проведении лотереи часть взноса участников идет организатору лотереи.

По виду функции выигрыша игры делятся на матричные, биматричные, непрерывные, выпуклые, сепарабельные и др. В **матричных** играх (при двух участниках) выигрыши первого игрока задаются матрицей, в **биматричных** выигрыши каждого игрока задаются своей матрицей. Другие типы таких игр различаются видом аналитического выражения платежной функции. По количеству ходов игры делятся на **одноходовые** (выигрыш распределяется после одного хода каждого игрока) и **многоходовые** (выигрыш распределяется после нескольких ходов). Многоходовые игры в свою очередь делятся на позиционные, стохастические, дифференциальные и др. В зависимости от объема имеющейся информации различают игры с **полней** и **неполней** информацией.

Игры, в которых участники стремятся добиться для себя наилучшего результата, сознательно выбирая допустимые правилами игры способы действий, называют иногда

*стратегическими*. Неопределенность может быть обусловлена как сознательным противодействием противника, так и неизвестными объективными обстоятельствами. В экономической практике нередко приходится формализовать (моделировать) ситуации, придавая им игровую схему, в которых один из участников безразличен к результату игры. Такие игры называют *играми с природой*, понимая под термином "природа" всю совокупность внешних обстоятельств, в которых сознательному игроку (его называют иногда "статистиком", а соответствующую игру – *статистической*) приходится принимать решение. Например, выбор агрономической службой сельскохозяйственного предприятия участков для посева той или иной культуры в надежде получить в предстоящем году наилучший урожай; определение объема выпуска сезонной продукции в ожидании наиболее выгодного для ее реализации уровня спроса; формирование пакета ценных бумаг в расчете на высокие дивиденды и т. д. В таких играх в качестве второго игрока выступает: в первом примере – в буквальном смысле природа; во втором – уровень спроса; в третьем – размеры ожидаемой прибыли. В играх с природой степень неопределенности для сознательного игрока (статистика) возрастает: если в стратегических играх каждый из участников постоянно ожидает наихудшего для себя ответного действия партнера, то в статистических играх "природа", являясь безразличной в отношении выигрыша инстанцией, может предпринимать и такие ответные действия (будем говорить: реализовать такие состояния), которые ей совершенно невыгодны, а выгодны сознательному игроку (статистику).

## 5.2. Матричные игры с нулевой суммой

Чтобы описать матричную игру с нулевой суммой, необходимо перечислить все возможные стратегии каждой из сторон и определить результат игры для каждой пары таких стратегий. Обозначим возможные стратегии первого игрока **A**:

$A_1, A_2, \dots, A_m$ . Стратегии второго игрока **B**, который является противником (оппонентом) игрока **A**, обычно обозначают  $B_1, B_2, \dots, B_n$ . Если игрок **A** применил стратегию  $A_i$ , а оппонент – стратегию  $B_j$ , то результатом такого

решения будет выигрыш игрока **A** равный  $a_{ij}$ . Для оппонента величина  $a_{ij}$  будет проигрышем. Если игрок **A** проигрывает, тогда его выигрыш  $a_{ij}$  будет отрицательной величиной. Таким образом, игра с нулевой суммой полностью описывается платежной матрицей игры

Игроки	Стратегии игрока <b>B</b>					
	<b>A</b>	<b>B</b>	$B_1$	...	$B_j$	...
Стратегии игрока <b>A</b>	$A_1$	$a_{11}$	...	$a_{1j}$	...	$a_{1n}$
	...		.....			
	$A_i$	$a_{i1}$	...	$a_{ij}$	...	$a_{in}$
	...		.....			
	$A_m$	$a_{m1}$	...	$a_{mj}$	...	$a_{mn}$

Если игра состоит только из личных ходов, то выбор пары чистых стратегий ( $A_i; B_j$ ) единственным образом определяет исход (результат) игры. Если же в игре используются случайные ходы, то исход игры определяется средним значением выигрыша (математическим ожиданием). Платежная матрица является табличной записью функции выигрыша. В теории матричных игр всегда предполагается, что в платежной матрице записаны выигрыши игрока **A**. Напомним, что выигрыши могут выражаться и отрицательными числами. Это означает, что в подобном случае фактически выигрывает игрок **B**.

### 5.3. Решение матричных игр в чистых стратегиях

Целью участников любой матричной игры является выбор наиболее выгодных стратегий, доставляющих игроку **A** максимальный выигрыш, а игроку **B** минимальный проигрыш. Стратегию игрока **A** называют оптимальной, если при ее применении выигрыш игрока **A** не уменьшается, какими бы стратегиями не пользовался игрок **B**. Оптимальной для игрока **B** называют стратегию, при использовании которой проигрыш игрока **B** не увеличивается, какие бы стратегии не применял игрок **A**.

При поиске оптимальных стратегий игроки опираются на основной принцип теории игр – **принцип осторожности**, в соответствии с которым каждый игрок, считая партнера по игре весьма разумным противником, выбирает свои действия в предположении о том, что соперник ни в коем случае не упустит ни единой возможности использовать любую его ошибку, любой промах в своих интересах. Поэтому игроки должны быть предельно внимательны при выборе каждой своей чистой стратегии.

Предположим, что игроку **A** надлежит сделать свой выбор. Анализируя платежную матрицу, он для каждой чистой стратегии  $A_i$  ( $i = \overline{1, m}$ ) сначала найдет минимальное значение  $\alpha$  ожидаемого выигрыша:  $\alpha_i = \min_j \alpha_{ij}$ ,  $i = \overline{1, m}$ , а затем из всех  $\alpha_i$  выберет наибольшее  $\alpha = \max_i \alpha_i$ , а затем выберет соответствующую ему чистую стратегию  $A_i$ .

Это и будет наиболее предпочтительная (гарантирующая, "перестраховочная") в данных условиях стратегия игрока **A**. Ее называют **максминной**, поскольку она отвечает величине

$$\alpha = \max_i \min_j \alpha_{ij}.$$

Число  $\alpha$  называется **нижней чистой ценой игры (максимином)**. Оно показывает, какой минимальный выигрыш может получить игрок **A**, правильно применяя свои чистые стратегии при любых действиях игрока **B**.

В свою очередь, игрок **B**, стремясь минимизировать проигрыш, при выборе наиболее предпочтительной стратегии, используя принцип осторожности выберет стратегию согласно условию:

$$\beta = \min_j \max_i \alpha_{ij}.$$

Ее называют **минимаксной**. Число  $\beta$  определяемое по формуле, называется **верхней чистой ценой игры (минимаксом)**. Она показывает, какой максимальный проигрыш может быть у игрока **B** при правильном выборе им своих чистых стратегий независимо от действий игрока **A**.

Таким образом, правильно используя чистые стратегии, игрок **A** обеспечит себе выигрыш не меньше  $\alpha$ , а игрок **B** в результате правильного применения своих чистых стратегий не позволит игроку **A** выиграть больше, чем  $\beta$ .

Различают стратегии **чистые** и **смешанные**. Чистая стратегия  $A_i$  ( $i = \overline{1, m}$ ) первого игрока (чистая стратегия  $B_j$   $j = \overline{1, n}$  второго игрока) – это возможный ход первого (второго) игрока, выбранный им с вероятностью, равной 1.

Если первый игрок имеет  $m$  чистых стратегий, а второй –  $n$  чистых стратегий, то для любой пары стратегий первого и второго игроков чистые стратегии можно представить в виде единичных векторов.

Например, для пары стратегий  $A_1$ ,  $B_2$  чистые стратегии первого и второго игроков запишутся в виде:  $p_1 = (1; 0; \dots; 0)$ ,  $q_2 = (0; 1; 0; \dots; 0)$ .

Для пары стратегий  $A_i$ ,  $B_j$  чистые стратегии можно записать в виде:

$$p_i = (0; \dots; 0; 1; 0; \dots; 0),$$

*i -е место*

$$q_j = (0; \dots; 0; 1; 0; \dots; 0),$$

*j -е место*

**Теорема 1.** В матричной игре нижняя чистая цена игры не превосходит верхней чистой цены игры, т. е.  $\alpha \leq \beta$ .

Если для чистых стратегий  $A_i, B_j$  игроков  $A$  и  $B$  соответственно имеет место равенство  $\alpha = \beta$ , то

- пару чистых стратегий  $(A_i, B_j)$  называют **седловой точкой матричной игры**,
- элемент  $a_{ij}$ , матрицы, стоящий на пересечении  $i$ -й строки и  $j$ -го столбца, — **седловым элементом платежной матрицы**,
- а число  $v = \alpha = \beta$  — **чистой ценой игры**.

Например, если игра задана следующей платежной матрицей

	<b>B</b> <sub>1</sub>	<b>B</b> <sub>2</sub>	<b>B</b> <sub>3</sub>	<b>B</b> <sub>4</sub>	<b>B</b> <sub>5</sub>	<b>B</b> <sub>6</sub>
<b>A</b> <sub>1</sub>	4	5	5	6	6	4
<b>A</b> <sub>2</sub>	7	9	3	-3	4	7
<b>A</b> <sub>3</sub>	5	3	3	1	4	5
<b>A</b> <sub>4</sub>	3	4	3	6	-6	3

то для определения нижней и верхней чистой цены игры следует определить, какой выигрыш гарантирует игроку **A** каждая из стратегий при самых неблагоприятных действиях оппонента. Это означает, что в каждой строке нужно найти минимальное значение  $a_i = \min_j a_{ij}, i = \overline{1, 4}$ .

Поскольку оппонент может провести такой же анализ для выбора одной из стратегий  $B_j, j = \overline{1, n}$ , то в каждом столбце он будет искать максимально возможные значения выигрыша игрока **A**:  $b_j = \max_i a_{ij}, j = \overline{1, 6}$ . Дополнив платежную матрицу этими значениями, получим:

	<b>B</b> <sub>1</sub>	<b>B</b> <sub>2</sub>	<b>B</b> <sub>3</sub>	<b>B</b> <sub>4</sub>	<b>B</b> <sub>5</sub>	<b>B</b> <sub>6</sub>	$a_i$
<b>A</b> <sub>1</sub>	4	5	5	6	6	4	<b>4</b>
<b>A</b> <sub>2</sub>	7	9	3	-3	4	7	-3
<b>A</b> <sub>3</sub>	5	3	3	1	4	5	1
<b>A</b> <sub>4</sub>	3	4	3	6	-6	3	-6
$b_j$	7	9	<b>5</b>	6	6	7	

Из этой таблицы видно, что выбор стратегии  $A_1$  гарантирует игроку **A** выигрыш не менее 4 ед. при любой стратегии оппонента. Таким образом,  $A_1$  — максиминая стратегия. Соответствующее ей значение 4 ед. — есть нижняя цена игры.

Для оппонента стратегия  $B_3$  минимизирует максимально возможный проигрыш и называется **минимаксной**. Используя ее, оппонент не может проиграть больше верхней цены игры  $\beta = \min_j \max_i a_{ij} = 5$ . Для рассматриваемого примера  $4 \leq v \leq 5$ .

Для платежной матрицы

	<b>B</b> <sub>1</sub>	<b>B</b> <sub>2</sub>	<b>B</b> <sub>3</sub>	<b>B</b> <sub>4</sub>	<b>B</b> <sub>5</sub>	$a_i$
<b>A</b> <sub>1</sub>	3	2	1	4	5	1
<b>A</b> <sub>2</sub>	4	3	<b>2</b>	5	6	2
<b>A</b> <sub>3</sub>	7	4	-3	4	-1	-3
$b_j$	7	4	<b>2</b>	5	6	

$A_2$  - максиминная стратегия,  $B_3$  - минимаксная стратегия, следовательно,  $\alpha = \beta = 2$  и  $v = 2$ . Особенность этого примера в том, что если оппонент придерживается стратегии  $B_3$ , то игроку **A** невыгодно использовать какую-либо стратегию кроме  $A_2$ . Но если игрок **A** использует стратегию  $A_2$ , то оппоненту придется использовать  $B_3$ . Причина состоит в том, что выигрыш  $a_{23} = 2$  одновременно является минимальным для максиминной стратегии  $A_2$  и максимальным для минимаксной стратегии  $B_3$ . Это игра с седловой точкой  $(A_2; B_3)$ , стратегии сторон  $A_2, B_3$ , соответствующие этой точке, являются **оптимальными чистыми стратегиями**. Чистая цена игры равна  $v = 2$ .

Про игры с седловой точкой говорят, что они решаются в чистых стратегиях, так как последние полностью определяют рациональное поведение сторон. Платежная матрица может иметь несколько седловых точек. Так, в следующей платежной матрице

	<b>B</b> <sub>1</sub>	<b>B</b> <sub>2</sub>	<b>B</b> <sub>3</sub>	<b>B</b> <sub>4</sub>	<b>B</b> <sub>5</sub>	$a_i$
<b>A</b> <sub>1</sub>	7	2	<b>2</b>	4	<b>2</b>	2
<b>A</b> <sub>2</sub>	4	3	<b>2</b>	5	<b>2</b>	2
<b>A</b> <sub>3</sub>	3	4	-3	4	-1	-3
$b_j$	7	4	<b>2</b>	5	<b>2</b>	

есть две максиминные стратегии у игрока **A**:  $A_1$  и  $A_2$ , а у оппонента - две минимаксные -  $B_3$  и  $B_5$ . Каждой из четырех седловых точек соответствует одна и та же чистая цена игры  $v = 2$ .

В развернутой форме признак матричной игры с седловой точкой запишется в виде

$$\max_i \min_j a_{ij} = \min_j \max_i a_{ij} = a_{i_0 j_0}.$$

Седловой элемент является наименьшим в строке и наибольшим в столбце. Поэтому, если игрок **B** отклонится от своей минимаксной стратегии, его проигрыш может только увеличиться. Аналогично, отклонение игрока **A** от своей максиминной стратегии ведет к уменьшению его выигрыша. Таким образом, наиболее предпочтительные стратегии в игре с седловой точкой обладают свойством устойчивости, создают ситуацию равновесия. Отсюда следует, что если в матрице игры существует седловой элемент, то

наилучшими для игроков являются наиболее предпочтительные чистые стратегии. Итак, если матричная игра имеет седловую точку, то она решается в чистых стратегиях: чистые стратегии, образующие седловую точку, и будут оптимальными, а решением игры считается тройка  $\{A_i, B_j, v\}$ .

#### 5.4. Решение матричной игры в смешанных стратегиях

Если игра имеет седловую точку, то оптимальными для игроков будут соответственно максиминная и минимаксная стратегии, а чистой ценой игры – седловой элемент платежной матрицы.

Проанализируем возможное поведение сторон на примере платежной матрицы без седловой точки.

	<b>B</b> <sub>1</sub>	<b>B</b> <sub>2</sub>	<b>B</b> <sub>3</sub>	<b>B</b> <sub>4</sub>	<b>B</b> <sub>5</sub>	<b>B</b> <sub>6</sub>	<i>a</i> <sub><i>i</i></sub>
<b>A</b> <sub>1</sub>	4	5	5	6	6	4	<b>4</b>
<b>A</b> <sub>2</sub>	7	9	3	-3	4	7	-3
<b>A</b> <sub>3</sub>	5	3	3	1	4	5	1
<b>A</b> <sub>4</sub>	3	4	3	6	-6	3	-6
<i>b</i> <sub><i>j</i></sub>	7	9	<b>5</b>	6	6	7	

Если игра проводится однократно и нет обоснованных предположений о действиях другой стороны, то игроку **A** логично использовать максиминную стратегию **A**<sub>1</sub>, а оппоненту – минимаксную стратегию **B**<sub>3</sub>. Но при многократном повторении игры оппонент, убедившись в предсказуемости поведения, может уменьшить свой проигрыш с 5 до 4, применив стратегию **B**<sub>1</sub>. Игрок **A** может в ответ увеличить свой выигрыш до 7, применив стратегию **A**<sub>2</sub>, которой оппонент может противопоставить **B**<sub>4</sub>, и т. д.

Следовательно, надо определиться, во-первых, какие стратегии и как часто надо использовать, во-вторых, в каком порядке их чередовать. При этом оппонента нельзя недооценивать, значит, любой фиксированный порядок неприемлем, так как может быть разгадан. Выход в том, что каждой доступной стратегии можно приписать определенную вероятность ее использования, а сам выбор доверить случаю, т.е. использовать смешанные стратегии.

**Смешанной стратегией** первого **A** (второго **B**) игрока называется вектор

$$p = (p_1, p_2, \dots, p_m), \text{ где } p_i \geq 0, (i = \overline{1, m}) \text{ и } \sum_{i=1}^m p_i = 1$$

$$(q = (q_1, q_2, \dots, q_n), \text{ где } q_j \geq 0, (j = \overline{1, n}) \text{ и } \sum_{j=1}^n q_j = 1),$$

*p<sub>i</sub>* и *q<sub>j</sub>* — вероятности, с которыми игроки **A** и **B** выбирают свои чистые стратегии **A<sub>i</sub>** и **B<sub>j</sub>** в ходе игры.

Механизм случайного выбора чистых стратегий, которым пользуется игрок **A**, обеспечивает ему бесконечное множество смешанных стратегий. Любая его чистая стратегия **A<sub>i</sub>** может рассматриваться как частный случай смешанной стратегии, *i*-я компонента которой равна 1, а остальные равны 0. Аналогично для игрока **B**.

Поскольку игроки выбирают свои чистые стратегии случайно и независимо друг от друга, игра имеет случайный характер и случайной становится величина выигрыша (проигрыша).

В таком случае средняя величина выигрыша (проигрыша) – математическое ожидание – является функцией смешанных стратегий *p,q*:

$$f(p, q) = \sum_{i=1}^m \sum_{j=1}^n a_{ij} p_i q_j$$

Функция  $f(p; q)$  называется **платежной функцией** игры с матрицей  $[a_{ij}]_{m \times n}$ .

Стратегии  $p = (p_1, p_2, \dots, p_m)$ ,  $q = (q_1, q_2, \dots, q_n)$  называются **оптимальными**, если для произвольных стратегий  $p = (p_1, p_2, \dots, p_m)$ ,  $q = (q_1, q_2, \dots, q_n)$  выполняется условие

$$f(p, q^*) \leq f(p^*, q^*) \leq f(p^*, q).$$

$(p^*; q^*)$  – является седловой точкой функции  $f(p; q)$ .

Использование в игре оптимальных смешанных стратегий обеспечивает первому игроку выигрыш, не меньший, чем при использовании им любой другой стратегии  $p$ , второму игроку – проигрыш, не больший, чем при использовании им любой другой стратегии  $q$ .

Значение платежной функции при оптимальных стратегиях определяет цену игры  $v$ , т. е.  $f(p^*; q^*) = v$ .

Совокупность оптимальных стратегий и цены игры составляет **решение игры**.

Оказывается, если использовать смешанные стратегии, то для любой матричной игры можно найти оптимальные стратегии и цену игры. В этом состоит смысл следующей теоремы, которая в теории игр считается основной.

**Теорема.** В смешанных стратегиях любая конечная матричная игра имеет седловую точку.

Выше дано определение оптимальных смешанных стратегий. Предположим, что задана матричная игра и указаны некоторые смешанные стратегии  $p^*$  и  $q^*$  игроков и число  $v$ . Как проверить, будет ли набор  $\{p; q; v\}$  решением игры? Для ответа на этот вопрос требуется установить справедливость неравенства для любых смешанных стратегий. Но поскольку различных смешанных стратегий существует бесконечное множество, то непосредственная проверка неравенства невозможна. Ответить на поставленный вопрос позволяет следующая теорема, которая имеет непосредственно практические приложения.

**Теорема.** Для того чтобы смешанные стратегии  $p^*$  и  $q^*$  были оптимальными для игроков **A** и **B** в игре с матрицей  $[a_{ij}]_{m \times n}$  с ценой  $v$ , необходимо и достаточно выполнение неравенств

$$\begin{aligned} \sum_{i=1}^m a_{ij} p_i^* &\geq v, \quad i = \overline{1, m}, \\ \sum_{j=1}^n a_{ij} q_j^* &\leq v, \quad j = \overline{1, n}. \end{aligned}$$

Итак, чтобы проверить, является ли набор  $\{p; q; v\}$  решением матричной игры, достаточно проверить, удовлетворяют ли  $p$  и  $q$  неравенствам теоремы и уравнениям

$$\sum_{i=1}^m p_i = 1, \quad \sum_{j=1}^n q_j = 1$$

в которых все  $p$  и  $q$  неотрицательны.

Итак, теорема утверждает, что если игрок **A** применяет оптимальную смешанную стратегию  $p^*$ , а игрок **B** – любую чистую стратегию, то выигрыш игрока **A** будет не меньше цены игры  $v$ ; если игрок **B** использует оптимальную смешанную стратегию  $q^*$ , а игрок **A** – любую чистую стратегию, то проигрыш игрока **B** не превысит цены игры.

Чистые стратегии игрока, входящие в его оптимальную смешанную стратегию с

вероятностями, отличными от нуля, называются *активными стратегиями* игрока.

**Теорема.** *Если один из игроков придерживается своей оптимальной смешанной стратегии, то его выигрыш остается неизменным и равным цене игры независимо от того, какую стратегию применяет другой игрок, если только тот не выходит за пределы своих активных стратегий.*

Можно доказать, что число активных стратегий игроков не превышает наименьшего из чисел  $m$  и  $n$ .

Решение игры можно существенно упростить, если своевременно выявить имеющееся в платежной матрице доминирование одних стратегий над другими, ибо это позволит предварительно сократить размерность матрицы.

В платежной матрице сравниваем элементы строк  $s$  и  $t$ , а именно:  $a_{sj}$  с элементами  $a_{tj}$  для  $j = \overline{1, n}$ . Если  $a_{sj} \geq a_{tj}$  ( $j = \overline{1, n}$ ) то выигрыш игрока **A** при стратегии  $A_s$  будет больше, чем при стратегии  $A_t$ , такую бы чистую стратегию не применил игрок **B**. В этом случае стратегия  $A_s$  доминирует над стратегией  $A_t$ . Стратегию  $A_s$  называют *доминирующей*, а стратегию  $A_t$  – *доминируемой*.

Поскольку игрок **B** заинтересован в минимизации проигрыша, доминирующими будет столбец с наименьшими элементами. Например, сравниваем элементы  $r$ -го и  $l$ -го столбцов. Если все элементы  $a_{ir} \geq a_{il}$  ( $i = \overline{1, m}$ ). То игроку **B** свой выбор выгодно сделать по  $l$ -му столбцу. В этом случае стратегия  $B_l$  игрока **B** доминирует над стратегией  $B_r$ . Стратегия  $B_l$  называется *доминирующей*, а стратегия  $B_r$  – *доминируемой*.

Если в матричной игре имеем строки (столбцы) с одними и теми же элементами, то такие строки (столбцы), а соответственно и стратегии игроков **A** и **B** называются *дублирующими*.

В матричной игре доминируемые и дублирующие строки (столбцы) можно опускать, что не влияет на решение игры, но позволяет уменьшить размерность платежной матрицы.

Упрощение платежных матриц за счет исключения заведомо невыгодных игрокам чистых стратегий оправдано ввиду справедливости следующей теоремы.

**Теорема.** *Пусть  $I$  — игра, в матрице которой  $k$ -я стратегия игрока **A** доминирует над  $s$ -й, а  $I'$  — игра, матрица которой получена из матрицы игры  $I$  исключением  $s$ -й строки. Тогда: а) цена игры  $I'$  равна цене игры  $I$ ; б) оптимальная смешанная стратегия  $q^* = (q_1; \dots; q_n)$  игрока **B** в игре  $I'$  является также его оптимальной смешанной стратегией и в игре  $I$ ; в) если  $p^* = (p_1; \dots; p_{s-1}; p_s + 1; \dots; p_m)$ -оптимальная смешанная стратегия игрока **A** в игре  $I'$ , то его смешанная стратегия  $p = (p_1; \dots; p_{s-1}; 0; p_s + 1; \dots; p_m)$  является оптимальной в игре  $I$ .*

Если стратегия  $A_s$  доминирует над стратегией  $A_t$ , то вероятность применения последней в оптимальной смешанной стратегии  $p^*$  игрока **A** равна нулю, а поэтому  $t$ -ю строку из платежной матрицы можно исключить. Если  $l$ -я стратегия  $B_l$  игрока **B** доминирует над  $r$ -й стратегией  $B_r$ , то  $r$ -й столбец из платежной матрицы можно исключить.

В случае необходимости платежную матрицу можно подвергать и другим преобразованиям, не меняющим вероятности активных стратегий игроков. Это вытекает из следующей теоремы.

**Теорема.** *Оптимальные смешанные стратегии  $p^*$  и  $q^*$  соответственно игроков **A** и **B** в матричной игре  $[a_{ij}]_{m \times n}$  с ценой  $v$  будут оптимальными и в матричной игре*

$[ba_{ij} + c]_{m \times n}$  с ценой  $v' = bv + c$ , где  $b > 0$ .

Можно упростить платежную матрицу, прибавляя, например, ко всем элементам достаточно большое положительное число, в результате чего можно получить новую матрицу с положительными (неотрицательными) элементами. Умножая элементы на подходящий положительный множитель (отличный от нуля), можно уменьшить (увеличить) элементы новой матрицы, что облегчает дальнейшие вычисления. При этом вероятности активных стратегий не меняются.

Так, разделив элементы матрицы

$$\begin{pmatrix} 400 & -300 \\ 200 & 600 \end{pmatrix}$$

на 100 (умножив на 0,01), а затем прибавив к элементам новой матрицы число 3, придем к матрице

$$\begin{pmatrix} 7 & 0 \\ 5 & 9 \end{pmatrix}.$$

Работать с этой матрицей конечно проще, чем с исходной.

Поскольку оптимальные смешанные стратегии игроков в результате рассмотренных упрощений платежной матрицы не меняются, то все получаемые в процессе преобразований матрицы называют стратегически эквивалентными.

## 5.5. Непосредственное решение матричной игры.

Если игра имеет седловую точку, то ее решение лежит в области чистых стратегий: оптимальными будут максиминная и минимаксная стратегии, а ценой игры – седловой элемент платежной матрицы.

Рассмотренная принципиальная схема определения оптимальных смешанных стратегий для игр без седловых точек, вытекающая из теоремы, предусматривает решение системы  $m + n$  линейных неравенств и линейных уравнений при условии неотрицательности всех неизвестных  $p$  и  $q$ . Однако этот путь нельзя признать рациональным, так как он связан с большим объемом вычислений, заметно увеличивающимся с ростом числа чистых стратегий участников игры.

Оказывается, решение любой конечной матричной игры может быть сведено к решению двух взаимно двойственных задач линейного программирования.

Исключение доминируемых стратегий может привести к потере некоторых решений; если же исключаются только строго доминируемые стратегии, то множество решений игры не изменяется.

### Решение игр 2×2. Сведение МИ к системе линейных алгебраических уравнений

Игра 2×2 является наиболее простым случаем конечных матричных игр. В этой игре каждый из игроков обладает только двумя стратегиями. Если игра 2×2 имеет седловую точку, то ее решение очевидно.

Рассмотрим игру

	$B_1$	$B_2$
$A_1$	$a_{11}$	$a_{12}$
$A_2$	$a_{21}$	$a_{22}$

Предположим, что игра не имеет седловой точки, т.е.  $\alpha \neq \beta$ . Требуется найти оптимальные смешанные стратегии игроков  $p^* = (p_1^*; p_2^*)$  и  $q^* = (q_1^*; q_2^*)$ , а также цену игры  $v$ .

Очевидно, что в игре 2×2, не имеющей седловой точки, обе стратегии игроков являются активными. Поэтому если игрок  $A$ , будет применять свою оптимальную смешанную стратегию, то, независимо от действий игрока  $B$ , выигрыш его будет равен цене игры  $v$ .

Игрок  $A$  будет применять стратегию  $A_1$  с вероятностью  $p_1$  и стратегию  $A_2$  с вероятностью  $p_2$ . Если игрок  $B$  отвечает своей стратегией  $B_1$ , то выигрыш игрока  $A$  определяется из уравнения

$$a_{11}p_1 + a_{21}p_2 = v.$$

Если же игрок  $B$  будет применять стратегию  $B_2$ , то выигрыш игрока  $A$  не изменится и определяется равенством

$$a_{12}p_1 + a_{22}p_2 = v.$$

Учитывая условие  $p_1 + p_2 = 1$ , получим систему трех уравнений с тремя неизвестными

$$\begin{cases} a_{11}p_1 + a_{21}p_2 = v, \\ a_{12}p_1 + a_{22}p_2 = v, \\ p_1 + p_2 = 1. \end{cases}$$

Решив эту систему, найдем оптимальное решение для игрока  $A$ :  $p^* = (p_1^*; p_2^*)$  и цену игры  $v$ . Аналогично определяется оптимальная стратегия игрока  $B$  из системы уравнений:

$$\begin{cases} a_{11}q_1 + a_{21}q_2 = v, \\ a_{12}q_1 + a_{22}q_2 = v, \\ q_1 + q_2 = 1. \end{cases}$$

Таким образом, матричная игра сведена к системе линейных уравнений.

### Графическое решение игр вида $(2 \times n)$ и $(m \times 2)$

Графический метод применим к играм, в которых хотя бы один игрок имеет только две стратегии. Рассмотрим игру  $(2 \times n)$ .

		Второй игрок			
		$y_1$	$y_2$	...	$y_n$
Первый игрок	$x_1$	$a_{11}$	$a_{12}$	...	$a_{1n}$
	$x_2 = 1 - x_1$	$a_{21}$	$a_{22}$	...	$a_{2n}$

Предполагаем, что игра не имеет седловой точки. Обозначим:  $x_1$  – вероятность применения первым игроком 1-й стратегии,  $x_2$  – вероятность применения первым игроком 2-й стратегии, причем  $x_2 = 1 - x_1$ ,  $y_1$  – вероятность применения вторым игроком 1-й стратегии,  $y_2$  – вероятность применения вторым игроком 2-й стратегии и т.д.,  $y_n$  – вероятность применения вторым игроком  $n$ -й стратегии. Ожидаемый выигрыш первого игрока при применении вторым игроком 1-й стратегии составит

$$a_{11}x_1 + a_{21}x_2 = a_{11}x_1 + a_{21}(1 - x_1) =$$

$$= a_{11}x_1 + a_{21} - a_{21}x_1 = (a_{11} - a_{21})x_1 + a_{21}.$$

Аналогично найдем ожидаемые выигрыши первого игрока при применении вторым игроком 2, 3, ...,  $n$ -й стратегий. Полученные данные поместим в таблицу.

Из таблицы видно, что ожидаемый выигрыш первого игрока линейно зависит от  $x_1$ . На оси  $OY$  построим выражения ожидаемых выигрышей первого игрока. На оси  $OX_1$  отметим точки  $x_1 = 0$  и  $x_1 = 1$ , через которые проведем прямые, перпендикулярные оси  $OX_1$ . Подставляя  $x_1 = 0$  и  $x_1 = 1$  в выражения ожидаемых выигрышей, найдем значения, которые отложим на соответствующих перпендикулярных прямых. Соединив эти точки, получим прямые.

Чистые стратегии второго игрока	Ожидаемые выигрыши первого игрока
1	$(a_{11} - a_{21})x_1 + a_{21}$
2	$(a_{12} - a_{22})x_1 + a_{22}$
...	...
$n$	$(a_{1n} - a_{2n})x_1 + a_{2n}$

Первый игрок должен выбирать такие стратегии, чтобы максимизировать свой минимальный ожидаемый выигрыш. Поэтому оптимальная стратегия первого игрока определяется как точка пересечения прямых, максимизирующих его минимальный ожидаемый выигрыш.

Аналогично определяется оптимальная стратегия второго игрока. Она определяется как точка пересечения прямых, минимизирующих его максимальные ожидаемые проигрыши.

## 5.6. Приведение матричной игры к задаче линейной оптимизации

В общем случае смешанные стратегии сторон могут содержать значительное число активных стратегий и платежную матрицу существенно упростить не удастся. Однако оптимальные смешанные стратегии игроков  $p^*$  и  $q^*$  всегда можно найти приведением матричной игры двух лиц с нулевой суммой к задаче линейного программирования.

Без ограничения общности можно предполагать, что цена игры – это положительная величина. В противном случае достаточно увеличить все элементы платежной матрицы на постоянную величину, что приведет к такому же увеличению цены игры без изменения искомых смешанных стратегий.

	$B_1$	$B_2$	...	$B_j$	...	$B_n$	
$A_1$	$a_{11}$	$a_{12}$	...	$a_{1j}$	...	$a_{1n}$	$p_1$
$A_2$	$a_{21}$	$a_{22}$	...	$a_{2j}$	...	$a_{2n}$	$p_2$
...	...	...	...	...	...	...	...
$A_i$	$a_{i1}$	$a_{i2}$	...	$a_{ij}$	...	$a_{in}$	$p_i$
...	...	...	...	...	...	...	...
$A_m$	$a_{m1}$	$a_{m2}$	...	$a_{mj}$	...	$a_{mn}$	$p_m$
	$q_1$	$q_2$	...	$q_j$	...	$q_n$	

Для игры размерностью  $m \times n$  при использовании игроком **A** оптимальной смешанной стратегии  $p^* = (p_1, p_2, \dots, p_m)$  любая чистая стратегия оппонента  $B_j$  приведет к выигрышу игрока **A** не менее чем цена игры  $v$ , следовательно, для платежной матрицы, заданной таблицей, имеем:

$$a_{1j}p_1 + a_{2j}p_2 + \dots + a_{ij}p_j + \dots + a_{mj}p_m \geq v, \quad j = \overline{1, n},$$

$$p_1 + p_2 + \dots + p_j + \dots + p_m = 1.$$

Разделив эти соотношения на положительную величину  $v$ , введем обозначения  $p_i/v = x_i$ ,  $i = \overline{1, m}$  и получим:

$$a_{1j}x_1 + a_{2j}x_2 + \dots + a_{ij}x_j + \dots + a_{mj}x_m \geq 1, \quad j = \overline{1, n},$$

$$x_1 + x_2 + \dots + x_j + \dots + x_m = 1.$$

Так как цель игрока **A** максимизировать цену игры  $v$ , т.е.  $\min Z = 1/v$ , то критерий – это

$$\min Z = x_1 + x_2 + \dots + x_j + \dots + x_m$$

при записанных выше  $n$  ограничениях. Решив полученную задачу линейной оптимизации, найдем как цену игры  $v$ , так и оптимальную смешанную стратегию игрока **A**  $p^* = (p_1, p_2, \dots, p_m)$  по формулам:

$$v = 1/(x_1 + x_2 + \dots + x_j + \dots + x_m),$$

$$p_i = vx_i, \quad i = \overline{1, m}.$$

Для смешанной стратегии оппонента построение линейной модели выполняется аналогично, но так как игрок **B** стремится к минимизации своего проигрыша, то решается оптимизационная задача на максимум. Таким образом, для оппонента необходимо решить симметричную двойственную задачу:

Найдем оптимальную смешанную стратегию  $q$  игрока **B**. Применяя ее, игрок **B** проигрывает не более  $v$  при любой чистой стратегии  $A_i$  игрока **A**, т. е.

$$\sum_{j=1}^n a_{ij} q_j \leq v, \quad i = \overline{1, m}.$$

Разделим обе части неравенства на  $v$  и введем переменные  $y_j = \frac{q_j}{v}$ ,  $j = \overline{1, n}$ , получаем

$$\sum_{j=1}^n a_{ij} y_j \leq 1, \quad i = \overline{1, m}, \quad y_j \geq 0, \quad j = \overline{1, n}$$

Игрок **B** стремится сделать свой гарантированный проигрыш  $v$  возможно меньше, а значит возможно большее величину

$$y_1 + \dots + y_n = \frac{1}{v}.$$

Учитывая соотношения, приходим к следующей задаче: максимизировать линейную функцию

$$z(y) = y_1 + \dots + y_n$$

при линейных ограничениях

$$\begin{aligned} \sum_{j=1}^n a_{ij} y_j &\leq 1, \quad i = \overline{1, m}, \\ y_j &\geq 0, \quad j = \overline{1, n}. \end{aligned}$$

Это задача линейного программирования, записанная в симметричной форме. Решив ее (например симплексным методом), найдем оптимальный вектор  $y^* = (y_1; \dots; y_n)$  и максимальное значение целевой функции  $z(y^*)$ , а затем определим цену игры и компоненты оптимальной смешанной стратегии

$$v = \frac{1}{z(y^*)}, \quad q_j = vy_j \quad (j = \overline{1, n}).$$

Сформулированные задачи образуют пару симметричных двойственных задач линейного программирования, а поэтому нет необходимости решать обе эти задачи. Найдя решение одной из них, воспользуемся соответствием между двойственными переменными, устанавливаемым применительно к каноническим формам задач:

Свободные			Базисные		
$x_1$	$\dots$	$x_m$	$x_{m+1}$	$\dots$	$x_{m+n}$
<u><math>y_{n+1} \dots y_{n+m}</math></u>			<u><math>y_1 \dots y_n</math></u>		
Базисные			Свободные		

и выпишем оптимальные решения двух задач линейного программирования.

### Статистические игры

Под *статистической игрой (игрой с природой)* будем понимать парную матричную игру, в которой один игрок заинтересован в наиболее выгодном для него исходе игры, а второй игрок («природа») совершенно безразличен к результату игры.

Так, в матричных играх предполагается, участие двух игроков, интересы которых противоположны. Поэтому действия каждого игрока направлены на увеличение

выигрыша (уменьшение проигрыша). Однако в некоторых задачах, приводящихся к игровым, имеется неопределенность, вызванная отсутствием информации об условиях, в которых осуществляется действие (погода, покупательский спрос и т.д.). Эти условия зависят не от сознательных действий другого игрока, а от объективной действительности. Такие игры называются играми с природой. Человек в играх с природой старается действовать осмотрительно, второй игрок (природа, покупательский спрос) действует случайно.

Предположим, что в игре с природой сознательный игрок **A** может использовать  $m$  чистых стратегий  $A_1, A_2, \dots, A_m$ , а природа **P** может реализовать  $n$  различных состояний  $P_1, P_2, \dots, P_n$ . Игроку **A** могут быть известны вероятности  $q_1, \dots, q_n$ , с которыми природа реализует свои состояния, но он может и не знать их.

Действуя против природы, игрок **A** имеет возможность использовать как чистые стратегии  $A_i$ , так и смешанные стратегии. Если игрок **A** в состоянии оценить (величиной  $a_{ij}$ ) последствия применения каждой своей чистой стратегии  $A_i$  при любом состоянии  $P_j$  природы, то игру можно задать матрицей:

$$A_{m \times n} = \begin{matrix} & \begin{matrix} P_1 & \dots & P_n \end{matrix} \\ \begin{matrix} A_1 \\ \dots \\ A_m \end{matrix} & \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \end{matrix},$$

которая называется *платежной матрицей*.

Игры с природой, хотя и являются частным случаем парных матричных игр, обладают и некоторыми особенностями. Например, при упрощении платежной матрицы отбрасывать те или иные состояния природы нельзя, так как она может реализовать любое состояние независимо от того, выгодно оно игроку **A** или нет.

Кроме того, решение достаточно найти только для игрока **A**, поскольку природа наши рекомендации воспринять не может.

Также в играх с природой смешанные стратегии имеют ограниченное значение: они приобретают смысл только при многократном повторении игры.

С учетом отмеченных особенностей сформулирован ряд критериев, которыми пользуются при выборе оптимальных стратегий игрока **A**. Эти критерии позволяют оценить принимаемое решение и высказать рекомендации по тому или иному образу действий. Если рекомендации, вытекающие из некоторых критериев, совпадают, принимается рекомендуемое решение. Если же рекомендации не совпадают, то необходимо дополнительное исследование.

Таким образом, цель при решении статистической игры заключается в определении такой стратегии (чистой или смешанной), которая при ее применении обеспечила бы наибольший выигрыш.

## 5.7. Решение статистической игры

1. Вывявление и отбрасывание дублирующих и заведомо невыгодных стратегий лица, играющего с природой;
- Стратегии природы отбрасывать нельзя!
2. Построить и исследовать матрицу рисков.
3. Оценить выигрыш при различных игровых ситуациях: критерии Вальда, Байеса, Сэвиджа и Гурвица и др.;
4. Сделать вывод о выборе наилучшей стратегии.

### Построение матрицы рисков

**Риском**  $r_{ij}$  игрока А, когда он пользуется чистой стратегией  $A_i$  при состоянии  $\Pi_j$  природы, называется разность между максимальным выигрышем, который он мог бы получить, если бы точно знал, что природой будет реализовано именно состояние  $\Pi_j$ , и тем выигрышем, который он получит, используя стратегию  $A_i$ , не зная, какое же состояние реализует природа:

$$r_{ij} = \beta_j - a_{ij} \geq 0,$$

где  $\beta_j = \max_i a_{ij}$  — максимальный элемент  $j$ -го столбца платежной матрицы.

Элементы матрицы рисков, соответствующие стратегиям  $A_i$  и  $\Pi_j$  характеризуют общую благоприятность или неблагоприятность для игрока А отдельных состояний природы.

**Матрица рисков имеет вид:**

	$\Pi_1$	$\Pi_2$	...	$\Pi_n$
$A_1$	$r_{11}$	$r_{12}$		$r_{1n}$
$A_2$	$r_{21}$	$r_{22}$		$r_{2n}$
...				
$A_m$	$r_{m1}$	$r_{m2}$		$r_{mn}$

### Критерии для принятия решений в статистических играх

1. Критерий, основанный на известных вероятностях условий.

Пусть известны вероятности  $q_j$  состояний  $\Pi_j$  природы, тогда пользуются **критерием Байеса**, в соответствии с которым оптимальной считается чистая стратегия  $A_i$ , при которой максимизируется средний выигрыш

$$\bar{a}_i = \sum_{j=1}^n a_{ij} q_j$$

игрока А, т.е. обеспечивается

$$\bar{a} = \max_i \bar{a}_i = \max_i \sum_{j=1}^n a_{ij} q_j.$$

Следует отметить, что когда известны вероятности состояний природы  $q_j$ ,  $j = \overline{1, n}$ , игроку А нет смысла пользоваться смешанными стратегиями.

Применение в игре с природой в этом случае любой смешанной стратегии  $p$  не может быть выгоднее для игрока А, чем применение оптимальной чистой стратегии.

2. Если объективные оценки состояний природы получить невозможно, то вероятности состояний природы могут быть оценены субъективно на основе **принципа недостаточного основания Лапласа**, согласно которому все состояния природы

полагаются равновероятными, т.е.  $q_1 = \dots = q_n = 1/n$ , и оптимальной считают чистую стратегию  $A_i$ , обеспечивающую максимальное среднее значение выигрыша:

$$\max_i \bar{a}_i = \frac{1}{n} \max_i \sum_{j=1}^n a_{ij}.$$

Если вероятности  $q_j$  состояний  $\Pi_j$  природы неизвестны и нельзя сделать о них никаких предположений, то пользуются критериями Вальда, Сэвиджа и Гурвица.

**3. Максиминный критерий Вальда.** По этому критерию рекомендуется применять максиминную стратегию. Она достигается из условия

$$\alpha = \max_i \min_j a_{ij}, \quad i = \overline{1, m}, \quad j = \overline{1, n},$$

и совпадает с нижней ценой игры. Критерий является пессимистическим, считается, что природа будет действовать наихудшим для человека образом.

**4. Критерий максимума.** Оптимальная стратегия выбирается из условия

$$m = \max_i \max_j a_{ij}, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Критерий является оптимистическим, считается, что природа будет наиболее благоприятна для человека.

**5. Критерий Гурвица.** Критерий рекомендует стратегию, определяемую по формуле

$$s = \max_i \left\{ \lambda \min_j a_{ij} + (1 - \lambda) \max_j a_{ij} \right\}, \quad i = \overline{1, m}, \quad j = \overline{1, n},$$

где  $\lambda$  (степень оптимизма) изменяется в диапазоне  $[0, 1]$ .

Критерий придерживается некоторой промежуточной позиции, учитывающей возможность как наихудшего, так и наилучшего поведения природы. При  $\lambda=1$  критерий превращается в критерий Вальда; при  $\lambda=0$  — в критерий максимума. На  $\lambda$  оказывает влияние степень ответственности лица, принимающего решение по выбору стратегии. Чем хуже последствия ошибочных решений, больше желания застраховаться, тем  $\lambda$  ближе к единице. В общем случае число  $\lambda$  выбирают из опыта или субъективных соображений.

**6. Критерий Сэвиджа.** Суть критерия состоит в выборе такой стратегии, чтобы не допустить чрезмерно высоких потерь, к которым она может привести.

Этот критерий, так же как и критерий Вальда, является критерием крайнего пессимизма (но пессимизм понимается иначе, чем в критерии Вальда: рекомендуется всячески избегать большого риска).

Согласно этому критерию, рекомендуется выбирать ту стратегию, при которой в наихудших условиях величина риска принимает наименьшее значение:

$$r = \min_i \max_j r_{ij}, \quad \text{— оптимальная стратегия, где } r_{ij} \text{ — элементы матрицы рисков.}$$

## 6.1. Введение

До появления сетевых методов планирование работ, проектов осуществлялось в небольшом объеме. Современное сетевое планирование начинается с разбиения программы работ на операции. Определяются оценки продолжительности операций, и строится сетевая модель (график). Построение сетевой модели позволяет проанализировать все операции и внести улучшения в структуру модели до начала ее реализации. Строится календарный график, определяющий начало и окончание каждой операции, а также взаимосвязи с другими операциями графика. Календарный график выявляет критические операции, которым надо уделять особое внимание, чтобы закончить все работы в директивный срок. Что касается некритических операций, то календарный план позволяет определить резервы времени, которые можно выгодно использовать при задержке выполнения работ или эффективном применении как трудовых, так и финансовых ресурсов.

*Сетевая модель* — графическое изображение плана выполнения комплекса работ, состоящего из нитей (работ) и узлов (событий), которые отражают логическую взаимосвязь всех операций. В основе методов сетевого планирования и управления (СПУ) лежит графическое представление проекта в виде сетевого графика. Сетевой график можно рассматривать как совокупность некоторого количества точек  $E_1, E_2, \dots$  и соответственно между ними установленных связей  $\vec{e}_1, \vec{e}_2, \dots$ , т.е.  $G=(E, \vec{e})$ . Если в паре вершин  $E_i, E_j$  указано направление связи, т.е. какая из них является первой, то соединяющий их отрезок называется дугой; если же ориентация не указана — ребром. Объект  $G$  называется *графом*, точки  $E_1, E_2, \dots$  — его *вершинами*, связи между ними  $\vec{e}_1, \vec{e}_2, \dots$  — *ребрами* (дугами). Ориентация дуг, т. е. указание "начала" и "конца" каждой из них, делает граф *ориентированным (орграфом)*. Граф  $G=(E, \vec{e})$  считается заданным, если заданы все его вершины и дуги. Исследование таких сетей проводится методами теории графов. Теория графов определяет понятием пути, объединяющим последовательность взаимосвязанных ребер. Контур означает такой путь, у которого начальная вершина совпадает с конечной. Сетевой график — это ориентированный граф без контуров. Будем отождествлять вершины орграфа с событиями, а дуги — с работами. События и работы — основные понятия в СПУ.

*Работа* — это любые действия, трудовые процессы, сопровождающиеся затратами ресурсов или времени и приводящие к определенным результатам. Это активный процесс, требующий затрат ресурсов, либо пассивный (ожидание), приводящий к достижению намеченного результата. На сетевых графиках работы изображают стрелками. Рядом со стрелкой указываются числовые характеристики: время выполнения работы, расход ресурса, количество исполнителей и т. д. Под работами подразумеваются не только реальные хозяйствственные или технологические процессы, требующие затрат времени и ресурсов для их осуществления, но и процессы, потребляющие только время. Также принято считать работами и те процессы, которые не требуют ни затрат времени, ни ресурсов. Это так называемые *фиктивные работы*. Они показывают, что одна работа не может совершаться раньше другой. На сетевых графиках фиктивные работы изображают пунктирными стрелками.

*Событие* — это результат (промежуточный или конечный) выполнения одной и/или нескольких предшествующих работ. *Событие* означает факт окончания всех работ в него входящих или начала работ из него выходящих. Событие не имеет протяженности во времени. На сетевом графике события изображаются кругами с указанием номера события. В каждое событие может входить и выходить из него несколько работ, а каждая работа ограничена двумя событиями. Событие выражает логическую связь между работами, заключающуюся в том, что работы, входящие в данное событие, непосредственно предшествуют работам, выходящим из него; ни одна выходящая из

данного события работа не может начинаться до окончания всех работ, входящих в это событие.

Событие, с которого начинается выполнение проекта, является *исходным*, оно не имеет предшествующих работ. Событие, которое констатирует факт завершения проекта, называется *завершающим*, оно не имеет последующих работ. Все прочие события являются *промежуточными*.

## 6.2. Правила построения сетевых графиков.

Прежде чем представить проект сетевым графиком, необходимо составить перечень работ, оценить продолжительность каждой работы и установить последовательность работ, т. е. точно определить, какие работы обязательно должны быть закончены, чтобы могла начаться любая из работ, входящих в проект. Такой перечень удобно представить в виде структурно-временной таблицы.

При построении сетевых графиков следует соблюдать определенные правила, вот некоторые из них:

- 1) сеть изображается слева направо, и каждое событие с большим порядковым номером изображается правее предыдущего. Общее направление стрелок, изображающих работы, также в основном должно быть расположено слева направо, при этом каждая работа должна выходить из события с меньшим номером и входить в событие с большим номером.
- 2) в сетевых графиках не должно быть "тупиков", т. е. событий, из которых не выходит ни одна работа (за исключением завершающего события);
- 3) в сетевых графиках не должно быть событий (кроме исходного), которым не предшествует хотя бы одна работа;
- 4) при построении сетевых графиков нельзя допускать, чтобы два смежных события были связаны двумя или большим количеством работ, что чаще всего бывает при изображении параллельно выполняемых работ; для изображения параллельных работ вводятся промежуточное событие и фиктивная работа
- 5) в сети не должно быть замкнутых циклов, т. е. цепей, соединяющих некоторые события с ними же самими;
- 6) если какие-либо сложные работы могут быть начаты до полного окончания непосредственно предшествующей им работы, то последняя изображается как ряд последовательно выполняемых работ, каждая из которых завершается определенным событием.
- 7) если для выполнения одной из работ необходимо получение результатов всех работ, входящих в предшествующее для нее событие, а для другой работы достаточно получить результат только одной или нескольких из этих работ, то должно быть дополнительно введено новое событие, отражающее результаты только этих последних работ, а также фиктивная работа, связывающая новое событие с прежним.

Построенный с соблюдением этих правил график является сетевой моделью выполнения проекта.

Для правильной нумерации событий можно воспользоваться графическим способом упорядочения вершин графа по рангам (методом вычеркивания дуг).

1. Исходную вершину (в которую не входит ни одна дуга) отнесем к рангу 0 и присвоим ей номер 1.
2. Вычеркнем все дуги, выходящие из вершины (1) и отнесем события, оказавшиеся без входящих дуг, к первому рангу. Этим событиям присвоим в произвольном порядке номера 2, 3, ... $k_1$ .
3. Вычеркнув все дуги, выходящие из вершины предыдущего ранга  $i$ , отнесем вершины, оказавшиеся без входящих дуг, к следующему  $i+1$ -му рангу. Присвоим им номера  $k_i+1, \dots, k_{i+1}$ . Этот шаг повторяем до тех пор пока все вершины не будут пронумерованы.

Любая последовательность работ сети, в которой конечное событие каждой работы совпадает с начальным событием следующей за ней работы, называется *путем*. Под длиной пути  $(i, j_1), (j_1, j_2), \dots (j_k, j)$  из  $(i)$  в  $(j)$  будем понимать продолжительность выполнения всей последовательности работ, составляющих этот путь,  $t_{i,j_1} + t_{j_1,j_2} + \dots + t_{j_k,j}$ . Путь, в котором начальная вершина совпадает с исходным

событием, а конечная — с завершающим, называется *полным*. Путь от исходного события до любого промежуточного события  $i$  называется *предшествующим событию i путем*. Предшествующий событию  $i$  путь, имеющий наибольшую длину, будет *максимальным предшествующим*. Он обозначается  $L_1(i)$ , а его продолжительность —  $t[L_1(i)]$ . Путь от данного события  $i$  до завершающего называется *следующим путем*. Такой путь с наибольшей длиной будет *максимальным следующим*, он обозначается  $L_2(i)$ , его продолжительность —  $t[L_2(i)]$ . *Критическим* называется полный путь, имеющий наибольшую продолжительность. Таких путей в сети может быть несколько. *Критический путь* — это путь, не имеющий резервов и включающий самые напряженные работы комплекса. Работы и события, принадлежащие критическому пути, называются *критическими*. Все остальные работы являются некритическими (ненапряженными) и обладают резервами времени, которые позволяют передвигать сроки их выполнения, не влияя на общую продолжительность выполнения всего комплекса работ.

Суммарная продолжительность работ, принадлежащих критическому пути, называется *критическим временем*  $t_{\text{кр}}$  выполнения всего комплекса работ. На сетевом графике критический путь выделяется двойной или жирной линией.

Продолжительность выполнения работ устанавливается на основании действующих нормативов или по экспертным оценкам специалистов. В первом случае временные оценки являются детерминированными (однозначными), во втором — стохастическими (вероятностными).

### 6.3. Временные параметры сетевого графика.

Основным временным параметром сетевого графика является продолжительность критического пути. Расчет критического пути включает два этапа. Первый называется прямым проходом. Вычисления начинают с исходного события и продолжают до тех пор, пока не будет достигнуто завершающее событие. Для каждого события определяется одно число, представляющее ранний срок его наступления. На втором этапе, называемом обратным проходом, вычисления начинают с завершающего события и продолжают, пока не будет достигнуто исходное событие. Для каждого события вычисляется поздний срок его наступления.

*Ранним сроком  $t_p(i)$  свершения события  $i$*  называется самый ранний момент времени, к которому завершаются все предшествующие этому событию работы. Т.к. может быть несколько путей, предшествующих данному событию, то ранний срок свершения события определяется продолжительностью максимального предшествующего пути

$$t_p(i) = t[L_1(i)]$$

*Поздним сроком  $t_n(i)$  свершения события  $i$*  является самый поздний момент, после которого остается ровно столько времени, сколько необходимо для завершения всех работ, следующих за этим событием, без превышения критического времени  $t_{kp}$ . Очевидно, что  $t_n(i)$  определяется разностью между  $t_{kp}$  и длиной максимального из последующих путей:

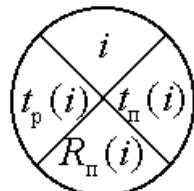
$$t_n(i) = t_{kp} - t[L_2(i)]$$

Для событий критического пути ранний и поздний сроки свершения совпадают.

Разность между поздним и ранним сроками свершения события составляет *резерв времени события  $R(i) = t_n(i) - t_p(i)$* . Резервы критических событий равны 0.

При расчете временных параметров вручную удобно проводить вычисления непосредственно на графике, воспользовавшись четырехсекторной схемой. В этом случае каждый кружок, обозначающий событие, делим на четыре сектора, в каждом из которых записываем следующую информацию.

1. Проставляем в верхних секторах номера событий (в соответствии с ранжированием).
2. Рассматривая события в порядке возрастания номеров, по входящим в данное событие работам определяем  $t_p(i)$  и записываем в левом секторе.
3. Начиная с конечного события, для которого  $t_n(n) = t_{kp}$  ( $n$  - номер конечного события), для каждого события по выходящим из него работам определяем  $t_n(i)$  и записываем в правом секторе.
4. В нижнем секторе записываем резерв времени события  $R(i)$ .



Зная сроки свершения событий, можно определить **временные параметры работ**.

*Ранний срок начала работы  $(i, j)$*  равен раннему сроку свершения события  $i$ :

$$t_{ph}(i, j) = t_p(i).$$

*Ранний срок окончания работы*  $(i, j)$  равен сумме раннего срока свершения начального события работы и ее продолжительности:

$$t_{\text{ро}}(i, j) = t_{\text{п}}(i) + t_{ij}.$$

*Поздний срок окончания работы*  $(i, j)$  совпадает с поздним сроком свершения ее конечного события:  $t_{\text{по}}(i, j) = t_{\text{п}}(j)$ .

*Поздний срок начала работы*  $(i, j)$  равен разности между поздним сроком свершения ее конечного события и продолжительностью:

$$t_{\text{пн}}(i, j) = t_{\text{п}}(j) - t_{ij}.$$

Так как сроки выполнения работ находятся в границах, определяемых  $t_{\text{ро}}(i, j)$  и  $t_{\text{по}}(i, j)$ , то они могут иметь разного вида резервы времени.

*Полный резерв времени работы* – это максимально возможный запас времени, на который можно отсрочить начало работы или увеличить продолжительность ее выполнения при условии, что конечное для данной работы событие наступит не позднее своего позднего срока:

$$R_{\text{п}}(i, j) = t_{\text{п}}(j) - t_{\text{п}}(i) - t_{ij}.$$

Таким образом, полный резерв времени работы есть максимальное время, на которое можно увеличить ее продолжительность, не изменяя продолжительности критического пути. Все некритические работы имеют полный резерв времени отличный от нуля.

*Независимый (свободный) резерв времени работы* – это запас времени, которым можно располагать при выполнении данной работы при условии, что начальное ее событие наступит в свой поздний срок, а конечное – в ранний срок:

$$R_{\text{н}}(i, j) = t_{\text{п}}(j) - t_{\text{п}}(i) - t_{ij}.$$

Величина необходимого резерва показывает продолжительность вынужденного ожидания наступления конечного события данной работы. Следует отметить, что критические операции должны иметь нулевой полный резерв времени, при этом свободный резерв также должен быть равен нулю.

*Частный резерв времени работы первого вида*  $R'(i, j)$  (гарантийный) отличается от полного тем, что его использование на данной работе возможно без уменьшения резервов у предшествующих:

$$R'(i, j) = t_{\text{п}}(j) - t_{\text{п}}(i) - t_{ij}.$$

*Частный резерв времени работы второго вида*  $R''(i, j)$  – это часть полного резерва, которая может быть использована для увеличения продолжительности данной работы или предшествующих ей работ без нарушения раннего срока наступления конечного события работы и без сокращения резервов времени у последующих работ:

$$R''(i, j) = t_{\text{п}}(j) - t_{\text{п}}(i) - t_{ij}.$$

#### **6.4. Линейный график (график Ганта) и учет потребностей в ресурсах.**

На графике Ганта каждая работа  $(i; j)$  изображается горизонтальным отрезком, длина которого в соответствующем масштабе равна времени ее выполнения. Начало каждой работы совпадает с ожидаемым сроком свершения ее начального события. Полный резерв времени работы изображается пунктирной линией. По графику Ганта можно определить критическое время выполнения комплекса работ и критический путь.

При решении задач СПУ для каждой из работ иногда задается количество ресурсов, необходимых для ее выполнения, так как одновременное выполнение некоторых операций из-за ограничений, связанных с рабочей силой, оборудованием и другими видами ресурсов, иногда оказывается невозможным. Именно в этом отношении представляют ценность полные резервы времени некритических операций.

Пусть  $r_{ij}$  – потребности в трудовых ресурсах для выполнения каждой работы и  $R$  – наличие трудовых ресурсов.

Выясним теперь потребности в трудовых ресурсах. Для этого на основе сетевого графика составляем линейный график (график Ганта).

На графике Ганта над каждой работой  $(i, j)$  проставляется потребность в ресурсах  $r_{ij}$ . Проектируем на ось времени начало и конец каждой работы. Проекцию, совпадающую с началом координат, обозначим  $\tau_0$ , следующую –  $\tau_1$  и т. д. В строке  $\sum r_{ij}$  записываем сумму ресурсов  $r_{ij}$  для каждого дня выполнения проекта. Полученные  $\sum r_{ij}$  наносим на график интенсивности использования ресурса. Пунктирная линия на графике проводится на уровне  $R$  ограничения наличного ресурса.

## 6.5. Оптимизация комплекса работ по ресурсам.

Под оптимальным распределением ресурсов понимается такое размещение работ во времени, при котором в любой момент времени было бы достаточно ресурса для выполнения работ, а время выполнения всего комплекса работ было бы минимальным. На практике получили широкое применение эвристические методы распределения ресурсов. Отметим, что приведенный ниже алгоритм не всегда позволяет найти оптимальное решение задачи, однако дает хорошее приближение к нему.

Алгоритм решения задачи: *Первый шаг*. 1. Нумеруем работы, расположенные над промежутком  $(\tau_0, \tau_1)$  на графике Ганта, в порядке возрастания их полных резервов. Работы с одинаковыми полными резервами времени нумеруем в порядке убывания интенсивностей.

2. Суммируем последовательно интенсивности работ, расположенных над промежутком  $(\tau_0, \tau_1)$  в порядке возрастания присвоенных им номеров, и сравниваем полученные суммы с заданной величиной ресурса  $R$ . Все работы, сумма интенсивностей которых не превосходит  $R$ , оставляем в первоначальном положении. Если после прибавления интенсивности какой-нибудь работы окажется, что суммарное потребление ресурсов больше  $R$ , то эту работу сдвигаем вправо на величину рассматриваемого промежутка, переходим к добавлению интенсивности следующей работы и так продолжаем до тех пор, пока не будут просмотрены все работы, расположенные над промежутком  $(\tau_0, \tau_1)$ . Результатом выполнения этого действия является новый график Ганта, момент  $\tau_1$  которого считаем началом оставшейся части комплекса работ.

*Общий шаг*. Предположим, что выполнено  $k$  шагов алгоритма и получен линейный график, момент  $\tau_k$  которого является началом оставшейся части комплекса работ.

1. Проектируем на ось времени начало и конец каждой работы, расположенной над промежутком  $(\tau_k, t_{kp})$ , и обозначаем проекцию, ближайшую к  $\tau_k$ , через  $\tau_{k+1}$ . Таким образом, выделен новый промежуток  $(\tau_k, \tau_{k+1})$ .

2. Определяем полные резервы работ  $R_p(i, j)$ , расположенных над промежутком  $(\tau_k, \tau_{k+1})$ , и нумеруем их. Сначала нумеруем работы  $(i, j)$ , начатые левее момента  $\tau_k$ , согласно возрастанию разностей между полными резервами этих работ и длительностями от начала до момента  $\tau_{k+1}$ . Работы с одинаковыми разностями нумеруем в порядке убывания интенсивностей. Все остальные работы нумеруем в порядке возрастания их полных резервов, а с одинаковыми резервами  $\square$  в порядке убывания интенсивностей.

3. Эти действия выполняются так же, как и действие 2 первого шага. Однако следует иметь в виду, что если сдвигу подлежит работа  $(i, j)$ , начатая левее  $\tau_k$ , то сдвигаем всю работу, т. е. начало этой работы устанавливаем в момент  $\tau_{k+1}$ .

4. Проверяем, все ли работы комплекса просмотрены. Если все, решение закончено, если нет, то возвращаемся к п. 1 общего повторяющегося шага.

## 6.6. Оптимизация сетевого графика по затратам сведением к задаче линейного программирования

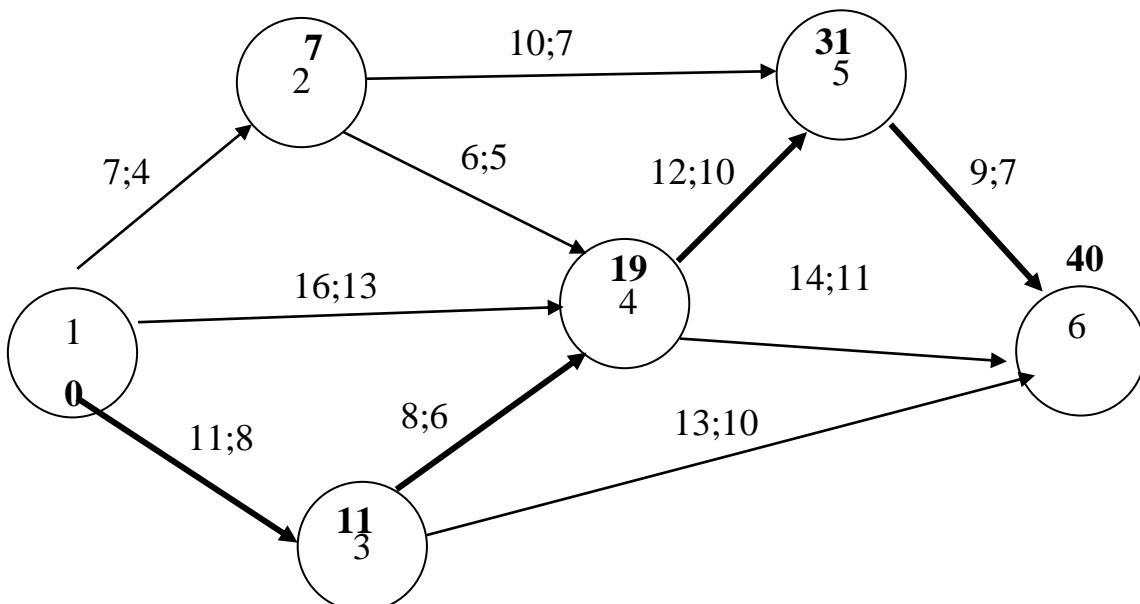
Проект представлен сетевым графиком. Для каждой работы известна ее продолжительность  $t_{ij}$  и минимально возможное время выполнения  $d_{ij}$ . Пусть задан срок выполнения проекта  $t_0$ , а расчетное  $t_{kp} > t_0$ . Продолжительность выполнения работы  $(i,j)$  линейно зависит от суммы дополнительно вложенных средств  $x_{ij}$  и выражается соотношением:  $t_{ij} = t_{ij} - k_{ij}x_{ij}$ . Технологические коэффициенты  $k_{ij}$  известны.

Требуется найти такие  $t^H_{ij}$ ,  $t^O_{ij}$ ,  $x_{ij}$ , чтобы:

- срок выполнения всего комплекса работ не превышал заданной величины  $t_0$ ;
- суммарное количество дополнительно вложенных средств было минимальным;
- продолжительность выполнения каждой работы  $t_{ij}$  была не меньше заданной величины  $d_{ij}$ .

Параметры	Работы										Срок выполнения проекта $t_0$
	1,2	1,3	1,4	2,4	2,5	3,4	3,6	4,5	4,6	5,6	
$t_{ij}$	7	11	16	6	10	8	13	12	14	9	
$d_{ij}$	4	8	13	5	7	6	10	10	11	7	
$k_{ij}$	0,1	0,3	0,2	0,05	0,25	0,2	0,12	0,5	0,08	0,02	34

1 Запишем все данные на сетевой график и рассчитаем сроки свершения событий.



Расчеты показали, что срок выполнения проекта  $t_{kp} = 40$ , т.е. превышает директивный срок  $t_0 = 34$ .

2. Составление математической модели задачи.

Целевая функция имеет вид

$$f = x_{12} + x_{13} + x_{14} + x_{34} + x_{35} + x_{45} + x_{14} + x_{34} + x_{35} + x_{45} \quad (\min).$$

Запишем ограничения задачи:

a) срок выполнения проекта не должен превышать  $t_0 = 34$ :

$$t^O_{36} \leq 34; t^O_{46} \leq 34; t^O_{56} \leq 34;$$

б) продолжительность выполнения каждой работы должна быть не меньше минимально возможного времени:

$$\begin{aligned} t^o_{12} - t^H_{12} &\geq 4; & t^o_{34} - t^H_{34} &\geq 6; \\ t^o_{13} - t^H_{13} &\geq 8; & t^o_{36} - t^H_{36} &\geq 10; \\ t^o_{14} - t^H_{14} &\geq 13; & t^o_{45} - t^H_{45} &\geq 10; \\ t^o_{24} - t^H_{24} &\geq 5; & t^o_{46} - t^H_{46} &\geq 11; \\ t^o_{25} - t^H_{25} &\geq 7; & t^o_{56} - t^H_{56} &\geq 7; \end{aligned}$$

в) зависимость продолжительности работ от вложенных средств:

$$\begin{aligned} t^o_{12} - t^H_{12} &= 7 - 0,1x_{12}; & t^o_{13} - t^H_{13} &= 11 - 0,3x_{13}; \\ t^o_{14} - t^H_{14} &= 16 - 0,2x_{14}; & t^o_{24} - t^H_{24} &= 6 - 0,05x_{24}; \\ t^o_{25} - t^H_{25} &= 10 - 0,25x_{25}; & t^o_{34} - t^H_{34} &= 8 - 0,2x_{34}; \\ t^o_{36} - t^H_{36} &= 13 - 0,12x_{36}; & t^o_{45} - t^H_{45} &= 12 - 0,5x_{45}; \\ t^o_{46} - t^H_{46} &= 14 - 0,08x_{46}; & t^o_{56} - t^H_{56} &= 9 - 0,02x_{56}; \end{aligned}$$

г) время начала выполнения каждой работы должно быть не меньше времени окончания непосредственно предшествующей ей работы:

$$\begin{aligned} t^H_{12} &= 0; & t^H_{13} &= 0; & t^H_{14} &= 0; \\ t^H_{24} &\geq t^o_{12}; & t^H_{25} &\geq t^o_{12}; \\ t^H_{34} &\geq t^o_{13}; & t^H_{36} &\geq t^o_{13}; \\ t^H_{45} &\geq t^o_{14}; & t^H_{45} &\geq t^o_{24}; \\ t^H_{45} &\geq t^o_{34}; \\ t^H_{46} &\geq t^o_{14}; & t^H_{46} &\geq t^o_{24}; \\ t^H_{46} &\geq t^o_{34}; \\ t^H_{56} &\geq t^o_{25}; & t^H_{56} &\geq t^o_{45}; \end{aligned}$$

д) условие неотрицательности неизвестных:

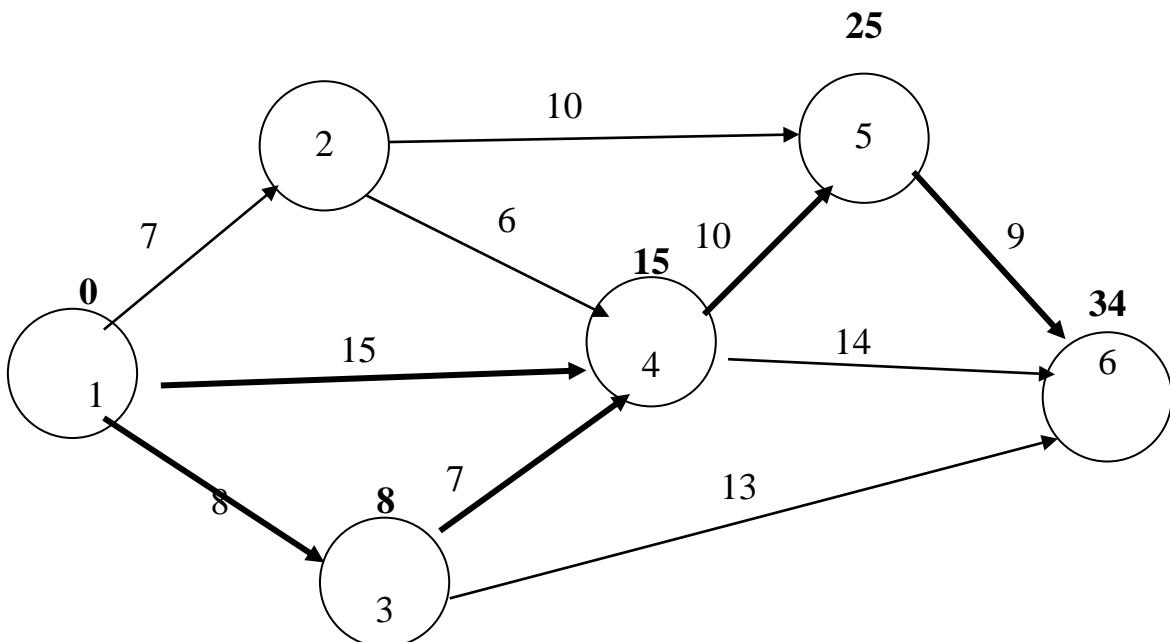
$$t^H_{ij} \geq 0, t^o_{ij} \geq 0, x_{ij} \geq 0, (i,j) \in \vec{e}.$$

### 3. Численное решение задачи:

Решив данную задачу, получаем следующие результаты:

$$\begin{aligned} t^H_{12} &= 0; & t^o_{12} &= 7; & t^H_{13} &= 0; & t^o_{13} &= 8; & t^H_{14} &= 0; & t^o_{14} &= 15; \\ t^H_{24} &= 7; & t^o_{24} &= 13; & t^H_{25} &= 7; & t^o_{25} &= 17; \\ t^H_{34} &= 8; & t^o_{34} &= 15; & t^H_{36} &= 8; & t^o_{36} &= 21; \\ t^H_{45} &= 15; & t^o_{45} &= 25; & t^H_{56} &= 25; & t^o_{56} &= 34; \\ x_{12} &= 0; & x_{13} &= 10; & x_{14} &= 5; & x_{24} &= 0; & x_{25} &= 0; \\ x_{34} &= 5; & x_{36} &= 0; & x_{45} &= 4; & x_{46} &= 0; & x_{56} &= 0; \\ f_{min} &= 24. \end{aligned}$$

Результаты представим на сетевом графике:



4. *Анализ полученных результатов.* Чтобы выполнить работы проекта за время  $t_0=34$ , необходимо дополнительно вложить 24 ден.ед. При этом средства распределятся следующим образом: 10 ден.ед. – в работу (1,3), 5 ден.ед. – в работу (1,4), 5 ден.ед. – в работу (3,4) и 4 ден.ед. – в работу (4,5), что приведет к сокращению продолжительности работы (1,3) на 3 дня, работы (1,4) - на 1 день, работы (3,4) - на 1 день и работы (4,5) - на 2 дня. Сокращение срока реализации проекта за счет вложения дополнительных средств составит 6 ед. времени.

## 6.7. Оптимизация сетевого графика по времени сведением к задаче линейного программирования

Проект представлен сетевым графиком. Для каждой работы известна ее продолжительность  $t_{ij}$  и минимально возможное время выполнения  $d_{ij}$ . Для сокращения срока реализации проекта выделено  $B$  ден.ед. Вложение дополнительных средств  $x_{ij}$  в работу  $(i,j)$  сокращает время ее выполнения до  $t_{ij} = t_{ij} - k_{ij}x_{ij}$ . Технологические коэффициенты  $k_{ij}$  известны.

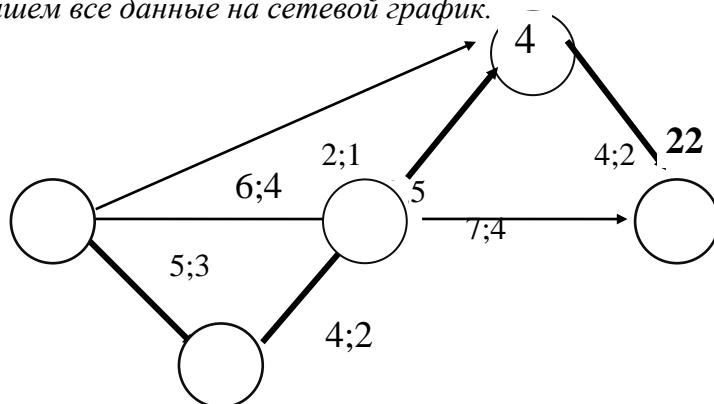
Требуется найти такие  $t^H_{ij}$ ,  $t^O_{ij}$ ,  $x_{ij}$ , чтобы:

- время выполнения всего комплекса работ было минимальным;
- количество используемых дополнительных средств не превышало  $B$  ден. ед.;
- продолжительность выполнения каждой работы была не меньше заданной величины  $d_{ij}$ .

Параметры	(1,2)	(1,3)	(1,4)	(2,3)	(3,4)	(3,5)	(4,5)	Сумма средств $B$
$t_{ij}$	5	6	2	4	9	7	4	
$d_{ij}$	3	4	1	2	5	4	2	
$k_{ij}$	0,5	0,2	0,3	0,25	0,4	0,2	0,1	47

**Решение.**

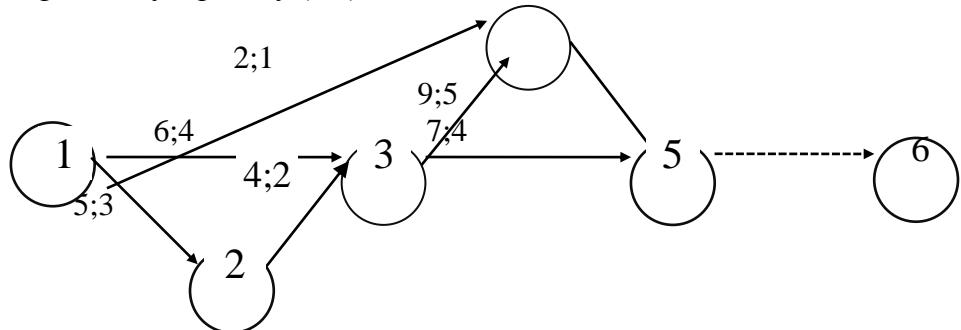
1. Запишем все данные на сетевой график.



По первоначальному условию  $t_{kp} = 22$ , т.е. проект может быть выполнен за 22 ед. времени.

2. Составление математической модели задачи.

Чтобы однозначно записать целевую функцию, добавим на сетевом графике фиктивную работу (5,6).



Целевая функция имеет вид  $t_{kp} = t^o_{56}$  (min).

Запишем ограничения задачи:

а) сумма вложенных средств не должна превышать их наличного количества:

$$x_{12} + x_{13} + x_{14} + x_{23} + x_{34} + x_{35} + x_{45} \leq 47;$$

б) продолжительность выполнения каждой работы должна быть не меньше минимально возможного времени:

$$t^o_{12} - t^h_{12} \geq 3; \quad t^o_{34} - t^h_{34} \geq 5;$$

$$t^o_{13} - t^h_{13} \geq 4; \quad t^o_{35} - t^h_{35} \geq 4;$$

$$t^o_{14} - t^h_{14} \geq 1; \quad t^o_{45} - t^h_{45} \geq 2;$$

$$t^o_{23} - t^h_{23} \geq 2; \quad t^o_{56} - t^h_{56} = 0;$$

в) зависимость продолжительности работ от вложенных средств:

$$t^o_{12} - t^h_{12} = 5 - 0,5x_{12}; \quad t^o_{13} - t^h_{13} = 6 - 0,2x_{13};$$

$$t^o_{14} - t^h_{14} = 2 - 0,3x_{14}; \quad t^o_{23} - t^h_{23} = 4 - 0,25x_{23};$$

$$t^o_{34} - t^h_{34} = 9 - 0,4x_{34}; \quad t^o_{35} - t^h_{35} = 7 - 0,2x_{35};$$

$$t^o_{45} - t^h_{45} = 4 - 0,1x_{45};$$

г) время начала выполнения каждой работы должно быть не меньше времени окончания непосредственно предшествующей ей работы:

$$t^h_{12} = 0; \quad t^h_{13} = 0; \quad t^h_{14} = 0;$$

$$t^h_{23} \geq t^o_{12}; \quad t^h_{34} \geq t^o_{13}; \quad t^h_{34} \geq t^o_{23}; \quad t^h_{35} \geq t^o_{13};$$

$$t^h_{35} \geq t^o_{23}; \quad t^h_{45} \geq t^o_{14}; \quad t^h_{45} \geq t^o_{34}; \quad t^h_{56} \geq t^o_{35}; \quad t^h_{56} \geq t^o_{45};$$

д) условие неотрицательности неизвестных:  $t^h_{ij} \geq 0, t^o_{ij} \geq 0, x_{ij} \geq 0, (i,j) \in \vec{E}$ .

### 3. Численное решение задачи:

Решив данную задачу, получаем следующие результаты:

$$t^h_{12} = 0; \quad t^o_{12} = 3; \quad t^h_{13} = 0; \quad t^o_{13} = 3; \quad t^h_{14} = 0; \quad t^o_{14} = 2; \quad t^h_{23} = 3; \quad t^o_{23} = 3; \quad t^h_{34} = 3;$$

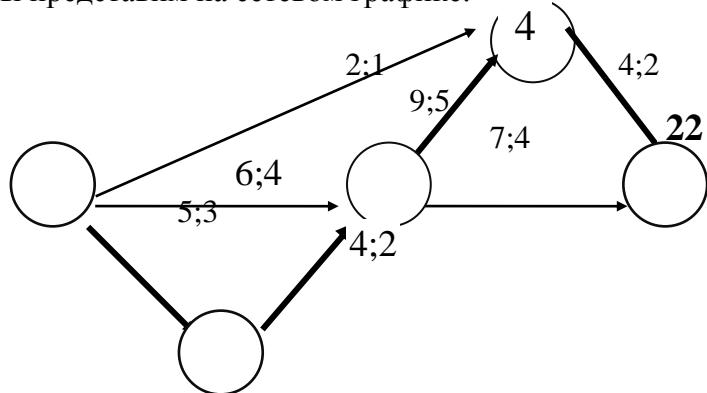
$$t^o_{34} = 8; \quad t^h_{35} = 3; \quad t^o_{35} = 10; \quad t^h_{45} = 8; \quad t^o_{45} = 10; \quad t^h_{56} = 10; \quad t^o_{56} = 10;$$

$$x_{12} = 20; \quad x_{13} = 0; \quad x_{23} = 0; \quad x_{14} = 0; \quad x_{34} = 10; \quad x_{35} = 0; \quad x_{45} = 20, \quad t_{kp} = 10.$$

Строка формул															O	P	Q	R	S	T	U	V	W	X	Y	Z	Левая	Вид	о/г	Ограничени
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z					
1 Переменн	x12	x13	x14	x23	x34	x35	x45	tн0	t01	t12	t23	t34	t35	t45	t56	t02	t13	t24	t35	t46	t57	t68	t79	t810	t912	t1014	min			
2 Значения	4	5	0	8	10	0	20	3	5	2	3	5	5	10	5	12	10	12	12	12	12	12	12	12	12	12				
3 Нижн. гр.																														
4 Верхн. гр.																														
5 Цел.Ф-ия																														
6																														
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
8	2																													
9	3																													
10	4																													
11	5																													
12	6																													
13	7																													
14	8																													
15	9																													
16	10	U,5																												
17	11	U,2																												
18	12	U,3																												
19	13	U,3																												
20	14	U,4																												
21	15	U,2																												
22	16	U,1																												
23	17																													
24	18																													
25	19																													
26	20																													
27	21																													
28	22																													
29	23																													
30	24																													
31	25																													



Результаты представим на сетевом графике:



**5. Анализ полученных результатов.** При дополнительном вложении 47 ден.ед., проект может быть выполнен за 10 ед. времени. При этом средства распределяются следующим образом: 20 ден.ед. – в работу (1,2), 10 ден.ед. – в работу (3,4) и 20 ден.ед. – в работу (4,5), что приведет к сокращению продолжительности работы (1,2). Сокращение срока реализации проекта за счет вложения дополнительных средств составит 8 ед. времени.

## 12.1. Постановка задачи оптимального управления

Пусть нестационарная динамическая система описывается системой обыкновенных дифференциальных уравнений (ОДУ)

$$\begin{cases} \dot{x}_1(t) = f_1(t, x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m), x_1(t_0) = x_1^0, \\ \dot{x}_2(t) = f_2(t, x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m), x_2(t_0) = x_2^0, \\ \dots \\ \dot{x}_n(t) = f_n(t, x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m), x_n(t_0) = x_n^0, \\ t \in [t_0, T] \end{cases}$$

Здесь  $\mathbf{X}^0$  —  $(n * 1)$ -вектор начальных условий,  $[t_0, T]$  — анализируемый период времени (где величина  $T$  не обязательно фиксирована!),  $\mathbf{X}(t)$  —  $(n * 1)$ -вектор фазовых переменных системы,  $\mathbf{U}(t)$  —  $(m * 1)$ -вектор управления. В той или иной форме могут быть заданы также условия на конце траектории системы, например, в виде  $\mathbf{X}(T) = \mathbf{X}^T$ .

Заметим, что правильнее было бы последние условия задавать в виде  $\mathbf{X}(t_k) = \mathbf{X}^T$ , где  $t_k \in [t_0, T]$  — момент достижения системой состояния  $\mathbf{X}^T$ . Однако для простоты записи мы будем использовать первую запись (если не оговорено противное).

Далее будем использовать описание анализируемой динамической системы в векторной форме

$$\dot{\mathbf{X}}(t) = \mathbf{F}(t, \mathbf{X}(t), \mathbf{U}(t)), \mathbf{X}(t_0) = \mathbf{X}^0, \mathbf{X}(T) = \mathbf{X}^T, t \in [t_0, T]. \quad (1)$$

Напомним, что если время  $t$  явно не входит в функцию  $\mathbf{F}$ , то динамическая система, описываемая системой ОДУ (1) называется стационарной.

На вектор  $\mathbf{X}(t)$  в общем случае могут быть наложены ограничения вида

$$\mathbf{X}(t) \in D_X \subset \Omega_X, \quad (2)$$

где  $D_X$  — множество допустимых значений вектора фазовых переменных системы,  $\Omega_X$  — некоторое функциональное пространство, например, пространство  $C^{(0)}[t_0, T]$  непрерывных на интервале функций (см. рис. 1).

На рис. 1 множество  $D_X = \{x(t) \mid x-1(t) \leq x(t) \leq x+1(t), t \in [t_0, T]\} \subset C(0)[t_0, T]$ .

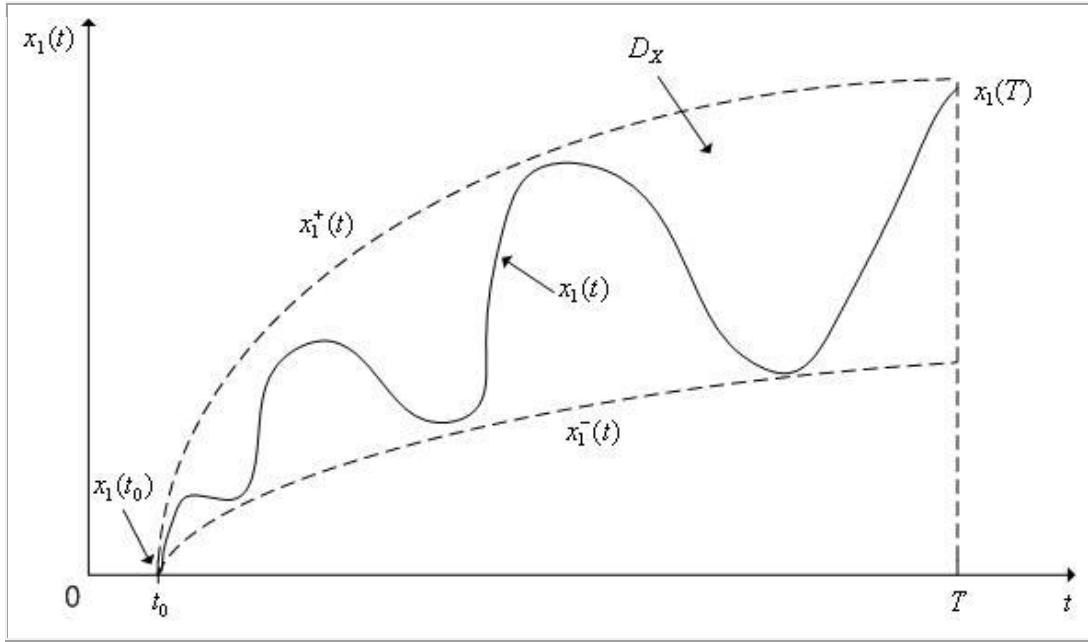


Рис. 1. Пример области допустимых значений вектора фазовых координат для одномерной системы ( $n = 1$ ).

На вектор управления  $\mathbf{U}(t)$  также обычно накладываются некоторые ограничения вида

$$\mathbf{U}(t) \in D_U \subset \Omega_U, \quad (3)$$

где  $D_U$  – множество допустимых управлений,  $\Omega_U$  – некоторое функциональное пространство, например, пространство  $L_2[t_0, T]$  функций "интегрируемых с квадратом" на интервале  $[t_0, T]$  (см. рис. 2).

На рис. 2 множество  $D_U = \{u(t) \mid u_1(t) \leq u(t) \leq u_2(t), t \in [t_0, T]\} \subset L_2[t_0, T]$ , а пространство допустимых управлений  $\Omega_U$ , полагается, составляют кусочно-постоянные функции, имеющие на интервале  $[t_0, T]$  конечное число разрывов первого рода.

Примечание 1

Используя уравнение (1), формально ограничения на вектор фазовых переменных  $D_X$  можно пересчитать в ограничения на вектор управления  $D_U$ . Однако, задача такого пересчета сложна и обычно рассматривают и те, и другие ограничения.

В качестве критерия качества управления (критерий оптимальности управления) в задачах оптимального управления обычно используют интегральный функционал вида

$$J_0(\mathbf{X}, \mathbf{U}) = \int_{t_0}^T f_0(t, \mathbf{X}(t), \mathbf{U}(t)) dt, \quad (4)$$

где  $f_0(t, \mathbf{X}(t), \mathbf{U}(t))$  – некоторая одномерная функция указанных переменных.

Заметим, что здесь и далее обозначение функционала вида  $J_0(\mathbf{X}, \mathbf{U})$  (без указания зависимости вектор-функций  $\mathbf{X}, \mathbf{U}$  от времени  $t$ ) означает зависимость  $J_0$  от этих функций в целом, как элементов функционального пространства. Часто в литературе вместо записи  $J_0(\mathbf{X}, \mathbf{U})$  в том же смысле используют запись  $J_0(\mathbf{X}(\cdot), \mathbf{U}(\cdot))$ .

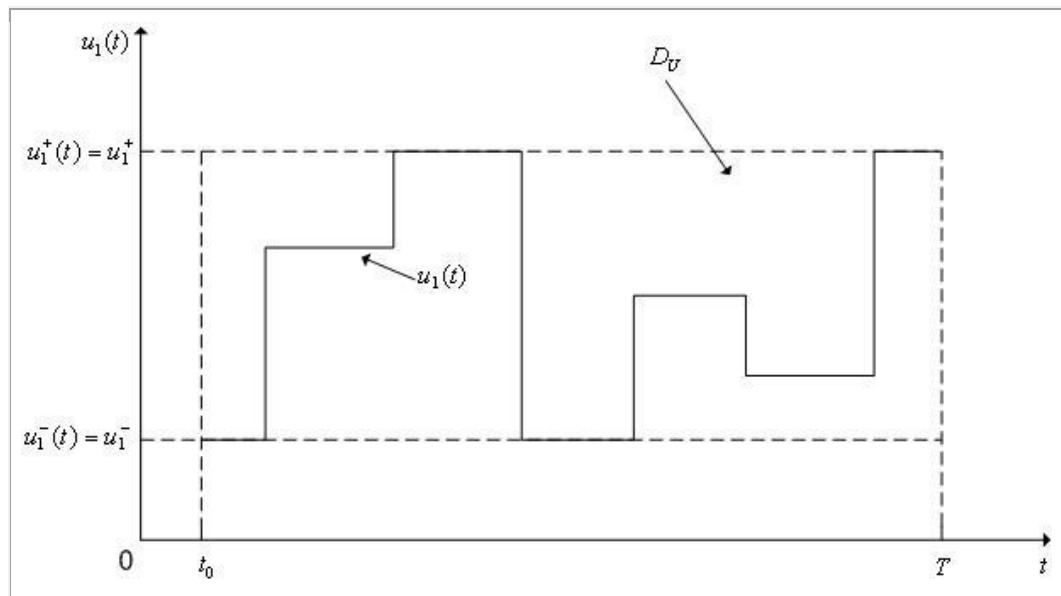


Рис. 2. Пример множества допустимых управлений для системы с одномерным управлением ( $m = 1$ ).

Итого, [задача оптимального управления динамической системой](#) обычно формализуется следующим образом: найти такой вектор управления  $\mathbf{U}(t)$ , удовлетворяющий условию (3), который на решениях системы ОДУ (1) обеспечивает минимум критерия оптимальности (4) при выполнении ограничений (2) на вектор фазовых переменных  $\mathbf{X}(t)$ . Заметим, что возможны и не интегральные формы функционалов качества управления, например:

$J_0(\mathbf{X}, \mathbf{U}) = f_0(t, \mathbf{X}(t), \mathbf{U}(t))$ , где  $t^* \in [t_0, T]$  – заданная точка;

$$J_0(\mathbf{X}, \mathbf{U}) = \max_{t \in [t_0, T]} f_0(t, \mathbf{X}(t), \mathbf{U}(t))$$

Важным частным случаем [задачи оптимального управления](#) является [задача оптимального быстродействия](#). [Критерий оптимальности управления](#) получается в этом случае из критерия оптимальности (4) при  $f_0(t, \mathbf{X}(t), \mathbf{U}(t)) \equiv 1$  и имеет вид  
 $J_0(\mathbf{X}, \mathbf{U}) = T - t_0$ .

## 12.2. Принцип максимума Л. С. Понtryгина

Для простоты записи положим, что  $t_0 = 0$  и рассмотрим стационарную динамическую систему

$$\begin{cases} \dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{U}(t)), t \in [0, T], \\ \mathbf{X}(0) = \mathbf{X}^0, \mathbf{X}(T) = \mathbf{X}^T, \\ \mathbf{U}(t) \in D_U, \end{cases} \quad (1)$$

Требуется найти управление  $\mathbf{U}(t) \in D_U$ , которое переводит эту систему из состояния  $\mathbf{X}^0$  в состояние  $\mathbf{X}^T$  и минимизирует [критерий оптимальности управления](#) - функционал

$$J_0(\mathbf{X}, \mathbf{U}) = \int_{t_0}^T f_0(t, \mathbf{X}(t), \mathbf{U}(t)) dt. \quad (2)$$

Введем в рассмотрение константу  $\psi^0$  и вспомогательную вектор-функцию  $\Psi(t) = (\psi_1(t), \psi_2(t), \dots, \psi_n(t)), t \in [0, T]$ , являющуюся решением системы ОДУ

$$\frac{d\psi_i}{dt} = - \frac{\partial \mathbf{H}(\Psi(t), \mathbf{X}(t), \mathbf{U}(t))}{\partial x_i}, i \in [1, n]$$

или в векторной форме

$$\dot{\Psi}(t) = \mathbf{H}_X(\Psi(t), \mathbf{X}(t), \mathbf{U}(t)),$$

где

$$\mathbf{H}(\Psi(t), \mathbf{X}(t), \mathbf{U}(t)) = \sum_{i=0}^n \psi_i(t) f_i(\mathbf{X}(t), \mathbf{U}(t)) - \quad (3)$$

гамильтониан динамической системы (1). Более удобна другая форма системы обыкновенных дифференциальных уравнений (ОДУ) для вспомогательной вектор-функции  $\Psi(t)$ :

$$\dot{\Psi}(t) = -\mathbf{F}_X(\mathbf{X}(t), \mathbf{U}(t))\Psi(t), \quad (4)$$

где  $\mathbf{F}_X(\mathbf{X}(t), \mathbf{U}(t))$  -  $(n * n)$ -матрица частных производных вектор-функции  $\mathbf{F}(\mathbf{X}(t), \mathbf{U}(t))$  по  $\mathbf{X}$ :

$$\mathbf{F}_X(\mathbf{X}(t), \mathbf{U}(t)) = \left( \frac{\partial f_i(\mathbf{X}(t), \mathbf{U}(t))}{\partial x_j} \right), i, j \in [1, n]$$

Система ОДУ (4) называется сопряженной системой.

Теорема 1 ([принцип максимума Л.С. Понtryгина](#)). Пусть  $\mathbf{U}^*(t), t \in [0, T]$  – допустимое управление, переводящее систему (1) из точки  $\mathbf{X}^0$  в точку  $\mathbf{X}^T$ , а  $\mathbf{X}^*(t)$  – соответствующая фазовая траектория. Для оптимальности (в смысле минимума функционала (2)) процесса  $(\mathbf{U}^*(t), \mathbf{X}^*(t), t \in [0, T])$  необходимо существование такой константы  $\psi^0 \leq 0$  и такого решения  $\Psi(t) = (\psi_1(t), \psi_2(t), \dots, \psi_n(t)), t \in [0, T]$  системы

ОДУ (5), что вектор-функция  $(\psi_0(t), \psi_1(t), \psi_2(t), \dots, \psi_n(t))$  не тривиальна и для любого момента времени  $t \in [0, T]$  выполняется условие максимума

$$\max_{U \in D_U} H(\Psi(t), X(t), U(t)) = H(\Psi(t), X^*(t), U^*(t)), \quad (5)$$

где гамильтониан системы  $H(\Psi(t), X(t), U(t))$  определяется выражением (4) •

Не тривиальность вектор-функции  $(\psi_0(t), \psi_1(t), \psi_2(t), \dots, \psi_n(t))$  означает, что среди величин  $(\psi_0(t), \psi_1(t), \psi_2(t), \dots, \psi_n(t))$  имеется хотя бы одна тождественно не равная нулю.

Отметим, что начальные условия для системы ОДУ (5) не заданы, т.е. можно получить только общее решение этой системы.

Примечание 1

Поскольку принцип максимума определяет лишь необходимое условие оптимальности, из того факта, что некоторая траектория удовлетворяет ему, не следует, что она оптимальна. Т.е. принцип максимума дает траектории лишь "подозрительные" на оптимальность. Для определения из них числа оптимальной траектории необходима дополнительная проверка.

Вслед за Федоренко Р.П. назовем систему уравнений (1), (2), (3), (4), (5) П-системой.

Рассмотрим теперь принцип максимума Л.С. Понтрягина для задачи оптимального быстродействия. Напомним, что в этом случае критерия оптимальности управления имеет вид (см. параграф 12.1)

$$J_0(X, U) = T. \quad (6)$$

Легко видеть, что гамильтониан системы (1) в этом случае равен

$$H(\Psi(t), X(t), U(t)) = \sum_{i=0}^n \psi_i(t) f_i(X(t), U(t)) \quad (7)$$

Теорема 2 (принцип максимума Л.С. Понтрягина для задачи оптимального быстродействия). Пусть  $U^*(t), t \in [0, T]_-$  – допустимое управление, переводящее систему (1) из точки  $X^0$  в точку  $X^T$ , а  $X^*(t)$  – соответствующая фазовая траектория. Для оптимальности (в смысле минимума функционала (6)) процесса  $(U^*(t), X^*(t), t \in [0, T])$  необходимо существование такого решения  $\Psi(t) = (\psi_1(t), \psi_2(t), \dots, \psi_n(t)), t \in [0, T]$  системы ОДУ (3), что вектор-функция  $(\psi_1(t), \psi_2(t), \dots, \psi_n(t))$  не тривиальна и для любого момента времени  $t \in [0, T]$  и выполняется условие максимума

$$\max_{U \in D_U} H(\Psi(t), X(t), U(t)) = H(\Psi(t), X^*(t), U^*(t)), \quad (8)$$

где гамильтониан системы определяется выражением (7) •

Известно множество обобщений принципа максимума Л.С. Понтрягина. Рассмотрим некоторые из них.

Задача с подвижными концами. Здесь векторы  $X^0, X^T$  не фиксированы, а заданы лишь некоторые гладкие многообразия  $M_0, M_T$  (гладкие поверхности, расположенные в пространстве  $X$ , размерность которых меньше  $n$ ) такие, что  $X^0 \in M_0, X^T \in M_T$ .

Дополнительные ограничения на вектор управления. Кроме требования минимизации критерия оптимальности  $J_0(\mathbf{X}, \mathbf{U})$  заданы ограничения вида  $J_i(\mathbf{U}(t)) = 0, i \in [1, m]$ ,

где  $J_i(\mathbf{U}(t))$  – некоторые функционалы над  $\mathbf{U}(t)$ .

Обобщение для нестационарных динамических систем.

Обобщение для динамических систем с параметрами:

$$\dot{\mathbf{X}}(t) = F(\mathbf{X}(t), \mathbf{U}(t), \mathbf{W}), t \in [0, T],$$

$$\mathbf{X}(t_0) = \mathbf{X}^0, \mathbf{X}(T) = \mathbf{X}^T, \mathbf{U}(t) \in D_U,$$

$$J_0(\mathbf{X}, \mathbf{U}, \mathbf{W}) = \int_{t_0}^T f_0(\mathbf{X}(t), \mathbf{U}(t), \mathbf{W}) dt,$$

где  $\mathbf{W} = (w_1, w_2, \dots, w_s)^T$  – неизвестный вектор параметров (констант). Требуется выбрать такой вектор  $\mathbf{W}^*$  и такое управление  $\mathbf{U}^*(t) \in D_U$ , чтобы перевести систему из состояния  $\mathbf{X}^0$  в состояние  $\mathbf{X}^T$  и минимизировать функционал  $J_0$ .

Пример 1

Рассмотрим материальную точку массой  $m = 1$ , которая свободно без трения движется по горизонтальной прямой. Пусть эта точка снабжена двигателем, развивающим силу тяги  $u(t)$  такую, что  $|u(t)| \leq 1$ .

Введем обозначения

$$x_1(t) = x(t), x_2(t) = \dot{x}(t) = V(t).$$

Таким образом, для рассматриваемой динамической системы размерность вектора фазовых переменных равна  $n = 2$  и этот вектор можно записать в виде  $\mathbf{X}(t) = (x_1(t), x_2(t))^T$ . Аналогично примеру 12.1.1 имеем следующее формальное описание системы (см. рис. 1):

$$\begin{cases} \dot{x}_1(t) = x_2(t), x_1(0) = x_1^0, \\ \dot{x}_2(t) = u(t), x_2(0) = x_2^0. \end{cases} \quad (9)$$

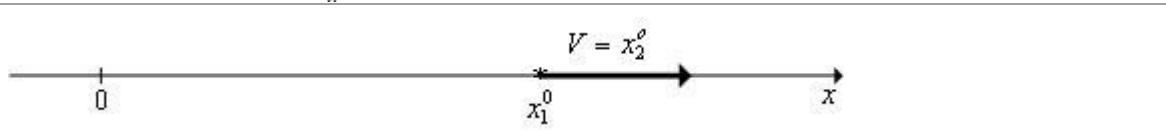


Рис. 1. К прим. 1.

Поставим [задачу оптимального быстродействия](#) – задачу о быстрейшем попадании рассматриваемой точки в начало координат  $(0, 0)$  из заданного начального состояния  $\mathbf{X}^0 = (x_1^0, x_2^0)^T$ . Другими словами, поставим задачу перевода за кратчайшее время материальной точки, имеющей начальное положение  $x_1^0$  и начальную скорость  $x_2^0$ , в начало координат с нулевой скоростью (чтобы точка перешла в начало координат и остановилась там).

Гамильтониан (7) системы (9) для [задачи оптимального быстродействия](#) имеет вид

$$\mathbf{H}(\Psi(t), \mathbf{X}(t), \mathbf{U}(t)) = \psi_1 f_1(\mathbf{X}, u) + \psi_2 f_2(\mathbf{X}, u) = \psi_1 x_2 + \psi_2 u$$

а система ОДУ для вспомогательной вектор-функции  $\Psi(t) = (\psi_1(t), \psi_2(t))$  — вид

$$\begin{cases} \dot{\psi}_1 = -\frac{\partial \mathbf{H}(\Psi, \mathbf{X}, u)}{\partial x_1} = 0, \\ \dot{\psi}_2 = -\frac{\partial \mathbf{H}(\Psi, \mathbf{X}, u)}{\partial x_2} = -\psi_1. \end{cases} \quad (10)$$

Легко получить явное общее решение системы ОДУ (10)

$$\begin{cases} \psi_1 = c_1, \\ \psi_2 = -c_1 t + c_2, \end{cases}$$

где  $c_1, c_2$  — произвольные постоянные.

Уравнение (8) имеет в данном случае вид

$$\max_{|u| \leq 1} (\psi_1 x_2 + \psi_2 u). \quad (11)$$

Из (11) следует, что оптимальное управление  $u^*(t)$  должно удовлетворять условию  $u^*(t) = \text{sign}(\psi_2(t)) = \text{sign}(-c_1 t + c_2)$ , т.е.

$$\begin{cases} u^*(t) = +1, \psi_2(t) > 0, \\ u^*(t) = -1, \psi_2(t) < 0. \end{cases}$$

Таким образом, оптимальное управление  $u^*(t), t \in [0, T]$ , является кусочно-постоянной функцией, принимающей значения  $\pm 1$  и имеющей не более двух интервалов постоянства (т.к. линейная функция  $-c_1 t + c_2$  не более одного раза меняет знак на отрезке  $[0, T]$ ). Приведем графическую иллюстрацию полученных результатов.

На отрезке времени, на котором  $u^*(t) = 1$ , из (9) последовательно имеем

$$\begin{cases} dx_1 = x_2 dt, \\ dx_2 = dt, \end{cases} \Rightarrow \frac{dx_1}{dx_2} = x_2, \Rightarrow x_1 = \frac{1}{2} x_2^2 + c.$$

Таким образом, фазовые траектории, для которых  $u^*(t) \equiv 1$ , представляют собой семейство парабол, соответствующих разным значениям константы  $c$  (см. рис. 2а). По этим параболам фазовые точки движутся снизу вверх (поскольку  $\dot{x}_2 = 1 > 0$ ).

Аналогично, управлению  $u^*(t) \equiv -1$  соответствует семейство парабол  $x_1 = -\frac{1}{2} x_2^2 + c$ , по которым фазовые точки движутся сверху вниз (т.к.  $\dot{x}_2 = -1 < 0$ ) — см. рис. 2б.

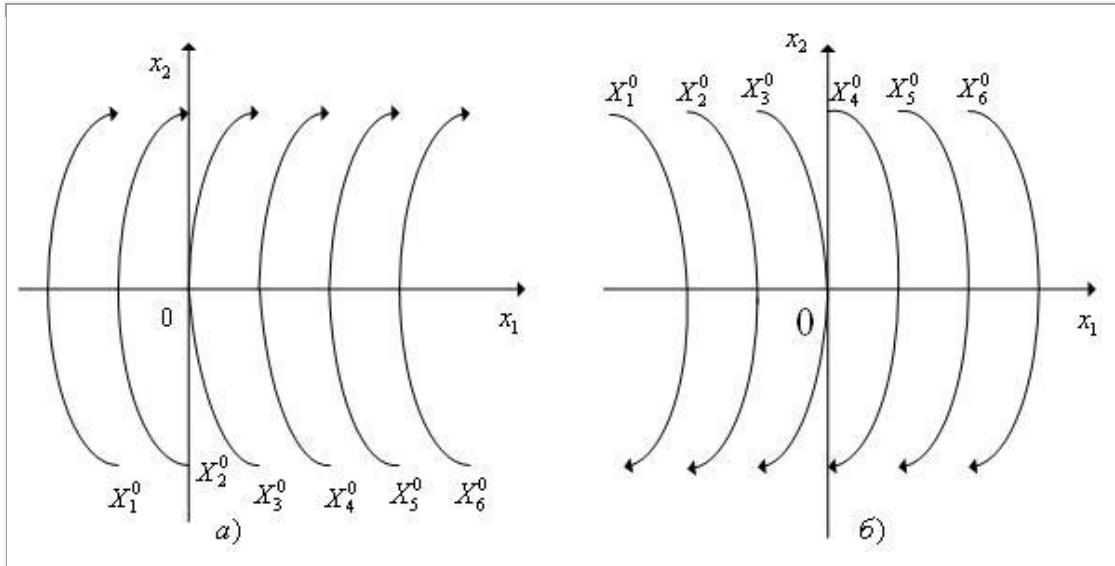


Рис. 2. Фазовые траектории системы (9): а – при  $u^*(t) \equiv 1$ ; б – при  $u^*(t) \equiv -1$ .

Так как оптимальное управление  $u^*(t)$  является кусочно-постоянной функцией, принимающей значения  $\pm 1$  и имеющей не более двух интервалов постоянства, возможны только варианты оптимальных фазовых траекторий системы, представленные на рис. 3.

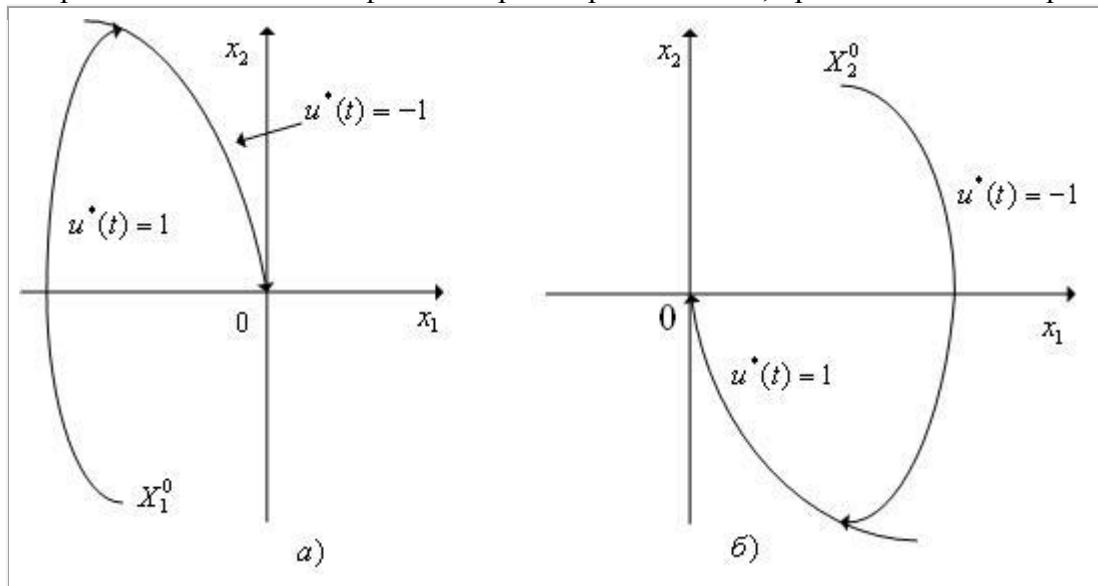


Рис. 3. Варианты фазовых траекторий системы (9).

Т.е. оптимальные фазовые траектории состоят из двух кусков парабол, примыкающих друг к другу. Причем второй из этих кусков лежит на той из парабол, которая проходит через начало координат.

Для произвольных начальных условий имеем следующую картину, представленную на рис. 4.

На рис. 4 дуга ОА имеет уравнение  $x_1 = \frac{1}{2}x_2^2$ , а дуга BO – уравнение  $x_1 = -\frac{1}{2}x_2^2$ .

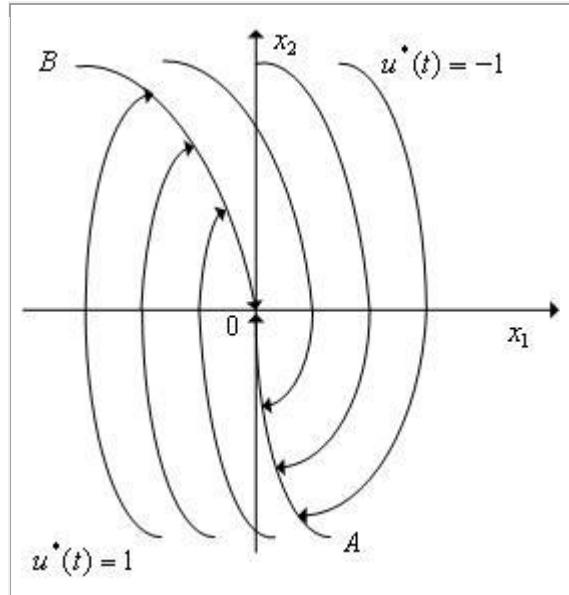


Рис. 4. Вид оптимальных фазовых траекторий системы (9).

Итак, согласно принципу максимума Л.С. Понtryгина оптимальные (по быстродействию) фазовые траектории системы (9) могут быть только вида, приведенного на рис. 4.

### 12.3. Метод решения задачи оптимального управления, использующий П-систему

Рассмотрим [задачу оптимального управления](#)

$$\begin{cases} \dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{U}(t)), \\ t \in [0, T], \mathbf{X}(0) = \mathbf{X}^0, \mathbf{X}(T) = \mathbf{X}^T, \mathbf{U}(t) \in D_U, \end{cases} \quad (1)$$

$$J_0(\mathbf{X}, \mathbf{U}) = \int_0^T f_0(\mathbf{X}(t), \mathbf{U}(t)) dt \rightarrow \min_{\mathbf{U}(t) \in D_U} \quad (2)$$

Напомним, что выражения (1), (2) совместно с выражением для соответствующего гамильтониана

$$\mathbf{H}(\Psi(t), \mathbf{X}(t), \mathbf{U}(t)) = \sum_{i=0}^n \psi_i(t) f_i(\mathbf{X}(t), \mathbf{U}(t)), \quad (3)$$

сопряженной системой ОДУ для вспомогательной вектор-функции  
 $\Psi(t) = (\psi_1(t), \psi_2(t), \dots, \psi_n(t))$   
 $\dot{\Psi}(t) = -\mathbf{F}_X(\mathbf{X}(t), \mathbf{U}(t))\Psi(t),$  (4)

и условием максимума

$$\max_{\mathbf{U} \in D_U} \mathbf{H}(\Psi(t), \mathbf{X}(t), \mathbf{U}(t)) = \mathbf{H}(\Psi(t), \mathbf{X}^*(t), \mathbf{U}^*(t)), \quad (5)$$

образуют [П-систему задачи оптимального управления](#) (1), (2).

Наиболее точные и аккуратные методы численного решения [задач оптимального управления](#) связаны с решением соответствующих [П-систем](#).

Положим, что уравнение (5) можно разрешить относительно  $\mathbf{U}^*(t)$ , т.е. найти функцию  $\mathbf{U}^*(t) = \mathbf{V}(\Psi(t), \mathbf{X}(t))$ . Тогда формально [П-система](#) (1), (2), (3), (4) (5) формально сводится к системе  $2n$  уравнений

$$\begin{cases} \dot{\mathbf{X}}^*(t) = \mathbf{F}(\mathbf{X}^*(t), \mathbf{V}(\Psi(t), \mathbf{X}^*(t))), \\ t \in [0, T], \mathbf{X}(0) = \mathbf{X}^0, \mathbf{X}(T) = \mathbf{X}^T \end{cases} \quad (6)$$

$$\dot{\Psi}(t) = -\mathbf{F}_X(\mathbf{X}(t), \mathbf{V}(\Psi(t), \mathbf{X}(t)))\Psi(t), \quad (7)$$

Введем в рассмотрение [П-процедуру](#):

Задаем некоторые начальные условия  $\Psi(0) = \Psi^0$  для функции  $\Psi(t)$ .

С заданными начальными условиями  $\Psi^0$  решаем задачу Коши для сопряженной системы ОДУ (7) – находим функцию  $\Psi(t)$ .

С найденной функцией  $\Psi(t)$  решаем задачи (6) – отыскиваем оптимальную фазовую траекторию системы  $\mathbf{X}^*(t)$ , соответствующую начальным условиям  $\Psi^0$ .

Находим разность  $\mathbf{X}(T) - \mathbf{X}^T$  (которая, очевидно, в общем случае не будет равна 0) •

П-процедура устанавливает функциональную зависимость разности вектора  $\Psi^0$ . Обозначим эту функциональную зависимость  $\mathbf{Z}$ :

$$\mathbf{X}(T) - \mathbf{X}^T = \mathbf{Z}(\Psi^0), \quad (8)$$

где  $\mathbf{Z}$  -  $n$ -мерная вектор-функция. (9)

Теперь, формально, решение П-системы сводится к решению системы нелинейных алгебраических уравнений (9): найти вектор  $\Psi^0$ , при котором  $\mathbf{Z}(\Psi^0) = 0$  или, что то же самое, при котором  $\mathbf{X}(T) = \mathbf{X}^T$ .

Чаще всего для решения СНАУ (9) используют метод касательных (метод Ньютона). Напомним схему этого метода для одномерного случая (см. параграф 4.8).

Пусть  $n = 1$ . Система (9) при этом имеет вид (см. рис. 1)

$$x(T) - x^T = z(\psi^0), \quad (10)$$

где  $x^T, \Psi^0$  — соответствующие скалярные константы, а  $x(T), z(\psi^0)$  — скалярные функции.

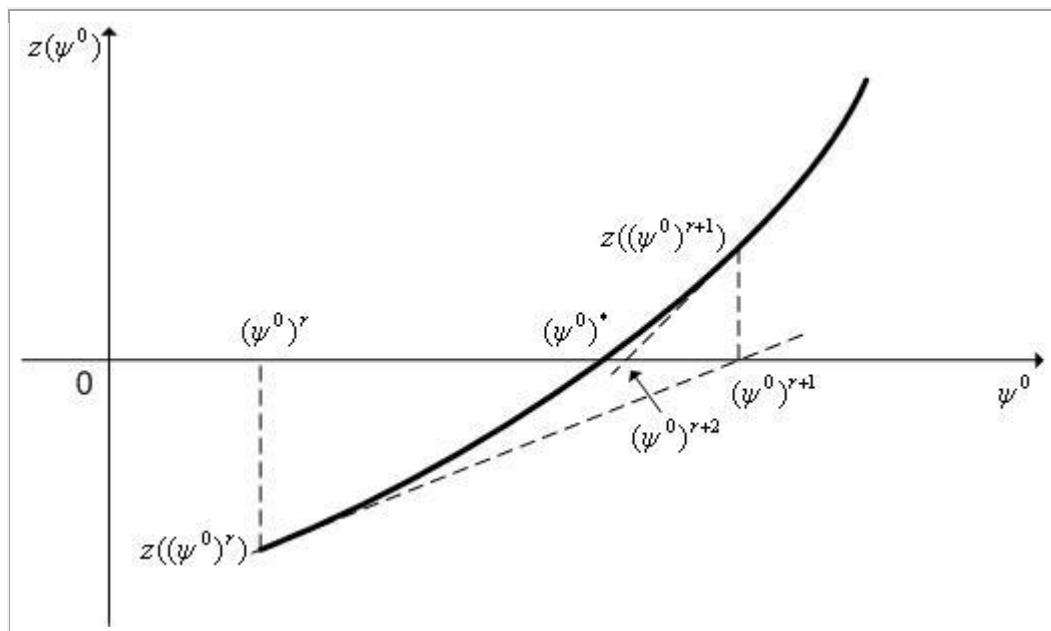


Рис. 1. К схеме метода касательных (метода Ньютона). Одномерный случай ( $n = 1$ ).

Линейная функция, аппроксимирующая функцию  $z(\psi^0)$  в точке  $(\psi^0)^r$ , записывается в виде

$$\tilde{z}(\psi^0, (\psi^0)^r) = z((\psi^0)^r) + z'_\psi((\psi^0)^r)(\psi^0 - (\psi^0)^r).$$

Приравняв правую часть этого выражения к нулю, получим итерационную формулу метода касательных

$$(\psi^0)^{r+1} = (\psi^0)^r - \frac{z((\psi^0)^r)}{z'_\psi((\psi^0)^r)}. \quad (11)$$

В многомерном случае ( $n = 1$ ) итерационная формула (11) имеет вид

$$(\Psi^0)^{r+1} = (\Psi^0)^r - (\mathbf{Z}'_\Psi((\Psi^0)^r))^{-1} \mathbf{Z}((\Psi^0)^r), \quad (12)$$

где  $(\mathbf{Z}'_\Psi)^{-1}$  — матрица, обратная матрице  $\mathbf{Z}'_\Psi$ ,

$$\mathbf{Z}'_\Psi((\Psi^0)^r) = \left( \frac{\partial z_i}{\partial \psi_j} \right)_{\Psi^0 = (\Psi^0)^r}, \quad i, j \in [1, n]. \quad (13)$$

Схема метода приближенного решения задачи оптимального управления, использующего П-систему

Полагаем счетчик числа итераций  $r = 0$ .

Из каких либо соображений задаем вектор  $(\Psi^0)^r$  — начальное значение вектора  $\Psi^0$ .

Выполняем [П-процедуру](#) для вектора  $(\Psi^0)^r$  — вычисляем значение функции  $\mathbf{Z}$  в точке  $(\Psi^0)^r$ :

$$\mathbf{Z}(\Psi^0) = \mathbf{X}(T) - \mathbf{X}^T.$$

Если условие окончание итераций не выполнено (см. ниже), то по формуле (12) вычисляем следующее приближение к  $(\Psi^0)^*$ , полагаем  $r = r + 1$  и переходим к п.3. Иначе переходим к следующему пункту.

В качестве приближения к оптимальному управлению принимаем  $\mathbf{U}^*(t) = \mathbf{V}(\Psi(t), \mathbf{X}(t))$ ,

где  $\Psi(t)$  — решение системы (7) с начальными условиями  $(\Psi^0)^r$ .

В качестве условия окончания итераций естественно использовать условие

$$\|\mathbf{X}(T) - \mathbf{X}^T\| \leq \varepsilon_x,$$

где  $\|\cdot\|$  — некоторая векторная норма, например, евклидова;  $\varepsilon_x$  — требуемая точность выполнения условия  $\mathbf{X}(T) - \mathbf{X}^T = 0$ .

Рассмотрим в заключение основные трудности, возникающие при решении [задачи оптимального управления](#) данным методом:

Поскольку функция  $\mathbf{Z}(\Psi^0)$  задана неявно, для вычисления  $(n * n)$ -матрицы  $\mathbf{Z}'_\Psi((\Psi^0)^r)$  приходится использовать численное дифференцирование. Для этого на каждой итерации, как минимум, приходится  $n$  раз решать задачи Коши (6), (7).

Метод Ньютона сходится лишь в достаточно малой окрестности решения. Поэтому на практике приходится использовать различные модификации метода Ньютона, обеспечивающие ускорение сходимости.

Решение уравнения (10) может быть не единственным.

Содержательные соображения для выбора вектора  $(\Psi^0)^0$  практически отсутствуют.

Иногда для выбора этого вектора используют приближенное решение [задачи оптимального управления](#) (1), (2) каким-либо другим методом, дающим грубое приближение к  $\mathbf{U}^*(t)$ .

## 12.4. Решение задачи оптимального управления методом вариаций в фазовом пространстве

Метод вариаций в фазовом пространстве разработан под руководством Н.Н. Моисеева в ВЦ АН СССР. Имеется несколько вариантов метода. Наиболее развитым является [метод локальных вариаций](#) Ф.Л. Черноусько.

Рассмотрим [задачу оптимального управления](#)

$$\begin{cases} \dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{U}(t)), \\ t \in [0, T], \mathbf{X}(0) = \mathbf{X}^0, \mathbf{X}(T) = \mathbf{X}^T, \mathbf{U}(t) \in D_U, \mathbf{X}(t) \in D_X, \end{cases} \quad (1)$$

$$J_0(\mathbf{X}, \mathbf{U}) = \int_0^T f_0(\mathbf{X}(t), \mathbf{U}(t)) dt \rightarrow \min_{\mathbf{U}(t) \in D_U}. \quad (2)$$

Обратим внимание на то, что в постановке задачи присутствуют кроме ограничений на вектор управления также ограничения на вектор фазовых координат.

Пусть  $\mathbf{X}(t)$  — некоторая траектория динамической системы (1), удовлетворяющая краевым условиям  $\mathbf{X}(0) = \mathbf{X}^0, \mathbf{X}(T) = \mathbf{X}^T$  и фазовым ограничениям  $\mathbf{X}(t) \in D_X$ . Покроем интервал времени  $[0, T]$  сеткой с узлами  $t_0 = 0, t_1, t_2, \dots, t_N = T$  и рассмотрим последовательность точек  $\mathbf{X}(t_0), \mathbf{X}(t_1), \mathbf{X}(t_2), \dots, \mathbf{X}(t_N)$ , которую будем называть траекторий системы (1).

[Элементарной операцией](#) называется решение задачи (1), (2) для интервала  $[t_i, t_{i+1}]$ , т.е. решение следующей задачи

$$\begin{cases} \dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{U}(t)), \\ t \in [t_i, t_{i+1}], \mathbf{X}(t_i) = \mathbf{X}^i, \mathbf{X}(t_{i+1}) = \mathbf{X}^{i+1}, \mathbf{U}(t) \in D_U, \mathbf{X}(t) \in D_X, \end{cases} \quad (3)$$

$$J_0(\mathbf{X}, \mathbf{U}) = \int_{t_i}^{t_{i+1}} f_0(\mathbf{X}(t), \mathbf{U}(t)) dt \rightarrow \min_{\mathbf{U}(t) \in D_U}. \quad (4)$$

Обозначим "цену" [элементарной операции](#)  $\partial J_0[\mathbf{X}^i, \mathbf{X}^{i+1}]$ . Т.е.  $\partial J_0[\mathbf{X}^i, \mathbf{X}^{i+1}]$  — это значение [критерия качества управления](#)  $J_0(\mathbf{X}, \mathbf{U})$  при оптимальном (в смысле минимума функционала (4)) переводе системы (1) из состояния  $\mathbf{X}^i$  в состояние  $\mathbf{X}^{i+1}$ .

Заметим, что в связи с малостью интервала  $[t_i, t_{i+1}]$  решать [задачу оптимального управления](#) (3), (4) не обязательно очень точно. Это обстоятельство позволяет во многих случаях достаточно просто получить цену [элементарной операции](#)  $\partial J_0[\mathbf{X}^i, \mathbf{X}^{i+1}]$ . Во введенных обозначениях функционал (2) на траектории  $\mathbf{X}(t_0), \mathbf{X}(t_1), \mathbf{X}(t_2), \dots, \mathbf{X}(t_N)$  равен

$$J_0(\mathbf{X}^0, \mathbf{X}^T) = \sum_{i=0}^{N-1} \partial J_0[\mathbf{X}^i, \mathbf{X}^{i+1}]. \quad (5)$$

[Локальной вариацией](#) траектории  $\mathbf{X}(t_0), \mathbf{X}(t_1), \mathbf{X}(t_2), \dots, \mathbf{X}(t_N)$  (в фазовом пространстве) называется траектория, отличающаяся от данной траектории только

значением  $\mathbf{X}$  в точке  $t_i$ . Обычно рассматриваются локальные вариации, в которых точка  $\mathbf{X}(t_i)$  смещается только вдоль координатных направлений (см.рис. 1):  $x_k^i = x_k^i \pm h_k e_k$ ,  $k \in [1, n]$ . Здесь  $x_k^i$  —  $k$ -я компонента вектора  $\mathbf{X}^i$ ,  $e_k$  —  $k$ -й орт в  $n$ -мерном пространстве,  $h_k$  — шаг по  $k$ -ой компоненте фазового вектора.

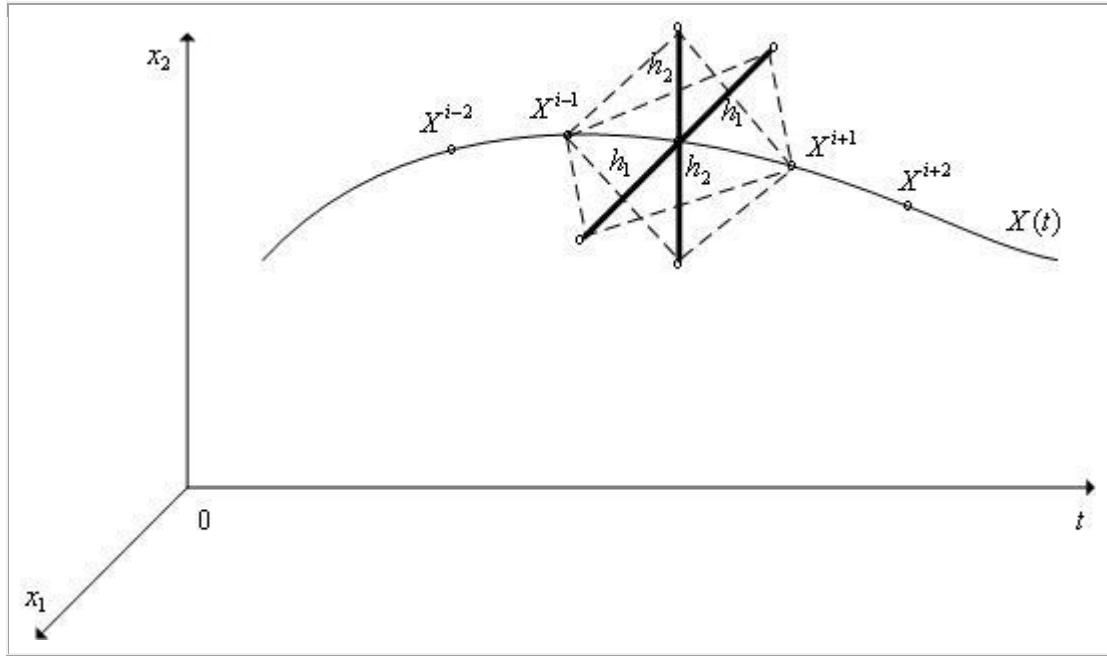


Рис. 1. К определению локальной вариации в фазовом пространстве.

Отметим, что [локальная вариация](#) траектории  $\mathbf{X}(t_0), \mathbf{X}(t_1), \mathbf{X}(t_2), \dots, \mathbf{X}(t_N)$  в точке  $t_i$  приводит к изменению в сумме (5) только двух слагаемых —  $\partial J_0[\mathbf{X}^{i-1}, \mathbf{X}^i]$  и  $\partial J_0[\mathbf{X}^i, \mathbf{X}^{i+1}]$ .

Схема метода локальных вариаций.

Из каких либо соображений задаем начальное приближение к оптимальной траектории  $\mathbf{X}(t)$ ,  $t \in [0, T]$ , удовлетворяющее краевым условиям  $\mathbf{X}(0) = \mathbf{X}^0$ ,  $\mathbf{X}(T) = \mathbf{X}^T$  и фазовым ограничениям  $\mathbf{X}(t) \in D_X$ . Покрываем интервал  $[0, T]$  сеткой с узлами  $t_0 = 0, t_1, t_2, \dots, t_N = T$ . Счетчику числа итераций  $r$  присваиваем значение 0.

Последовательно для  $i = 1, 2, \dots, N$  на интервале  $[t_{i-1}, t_i]$  выполняем [элементарную операцию](#) и определяем ее "цену"  $\partial J_0[\mathbf{X}^{i-1}, \mathbf{X}^i]$ .

Для каждого  $i = 1, 2, \dots, N-1$  последовательно для каждого  $k = 1, 2, \dots, n$  выполняем следующие действия:

3.1) выполняем [локальную вариацию](#)  $x_k^i = x_k^i + h_k e_k$ ;

3.2) если полученная в результате новая точка  $\mathbf{X}_k^i$  является допустимой (т.е.  $\mathbf{X}_k^i \in D_X$ ), то переходим к следующему пункту. Иначе переходим к п.3.4;

3.3) на интервалах  $[t_{i-1}, t_i]$ ,  $[t_i, t_{i+1}]$  выполняем [элементарные операции](#) и определяем их "цены"  $\partial J_0[\mathbf{X}^{i-1}, \mathbf{X}_k^i]$ ,  $\partial J_0[\mathbf{X}_k^i, \mathbf{X}^{i+1}]$ . Если данная [локальная вариация](#) была успешной — привела к уменьшению [критерия качества управления](#) (5)

$$\partial J_0[\mathbf{X}^{i-1}, \mathbf{X}_k^i] + \partial J_0[\mathbf{X}_k^i, \mathbf{X}^{i+1}] < \partial J_0[\mathbf{X}^{i-1}, \mathbf{X}^i] + \partial J_0[\mathbf{X}^i, \mathbf{X}^{i+1}]$$

то

полагаем

$$\partial J_0[\mathbf{X}^{i-1}, \mathbf{X}^i] = \partial J_0[\mathbf{X}^{i-1}, \mathbf{X}_k^i],$$

$$\partial J_0[\mathbf{X}^i, \mathbf{X}^{i+1}] = \partial J_0[\mathbf{X}_k^i, \mathbf{X}^{i+1}]$$

и переходим к следующему пункту;

$$3.4) \text{ выполняем локальную вариацию } \mathbf{x}_k^i = \mathbf{x}_k^i - h_k e_k;$$

3.5) если полученная в результате новая точка  $\mathbf{X}_k^i$  является допустимой (т.е.  $\mathbf{X}_k^i \in D_X$ ), то переходим к следующему пункту. Иначе, полагаем  $k = k + 1$  переходим к п.3.1.

3.6) выполняем действия, указанные в п.3.3.

Проверяем выполнение условия окончания итераций (см. ниже). Если это условие выполнено, то в качестве приближения к оптимальной траектории  $\mathbf{X}^*(t), t \in [0, T]$  принимаем текущую траекторию  $\mathbf{X}^0, \mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^N$ , а в качестве приближения к оптимальному управлению  $\mathbf{U}^*(t), t \in [0, T]$  — управления, найденные при выполнении элементарных операций, соответствующих траектории  $\mathbf{X}^0, \mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^N$ . Иначе полагаем  $r = r + 1$  и переходим к п.2 •

В качестве условия окончания итераций используется равенство нулю количества удачных локальных вариаций после данной итерации.

Действия, указанные в пункте 3 приведенной схемы метода локальных вариаций, будем называть основным циклом метода локальных вариаций. Легко видеть, что временная сложность основного цикла равна  $O(nM)$ .

В изложенном виде метод локальных вариаций обладает рядом серьезных недостатков. Назовем основные из этих недостатков:

Существование тупиковых ситуаций — во множестве локальных вариаций не оказывается удачной не потому, что данная траектория оптимальна, а потому, что исследуются не все возможные вариации траектории, а только чрезвычайно узкое множество соседних с данной.

Медленная сходимость.

Трудность построения элементарной операции.

Указанные недостатки приводят к тому, что в изложенном виде метод локальных вариаций используется редко. В вычислительной практике используются различные модификации метода локальных вариаций — метод дробных шагов, метод бегущей волны, метод трубки и др.

## 12.5. Решение задачи оптимального управления методом вариаций в пространстве управлений

Метод разработан в ИПМ АН СССР Федоренко Р.П.

Рассмотрим [задачу оптимального управления](#)

$$\begin{cases} \dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{U}(t)), \\ t \in [0, T], \mathbf{X}(0) = \mathbf{X}^0, \mathbf{U}(t) \in D_U, \end{cases} \quad (1)$$

$$J_0(\mathbf{X}, \mathbf{U}) = \int_0^T f_0(\mathbf{X}(t), \mathbf{U}(t)) dt \rightarrow \min_{\mathbf{U}(t) \in D_U} \quad (2)$$

Обратим внимание на то, что данная постановка [задачи оптимального управления](#) не содержит граничного условия  $\mathbf{X}(T) = \mathbf{X}^T$  и ограничений на вектор фазовых координат  $\mathbf{X}(t)$ . Заметим также, что поскольку вектор фазовых координат  $\mathbf{X}(t)$  при фиксированных начальных условиях  $\mathbf{X}(0) = \mathbf{X}^0$  определяется только управлением  $\mathbf{U}(t)$ , в число аргументов [критерия качества управления](#)  $J_0$  можно не включать вектор  $\mathbf{X}$ .

Положим, что известно некоторое управление  $\mathbf{U}(t) \in D_U$ , которое мы будем называть невозмущенным управлением.

В рассматриваемом методе существенно используется производная функционала  $J_0(\mathbf{U})$ . Если для любых достаточно малых возмущений  $\delta \mathbf{U}$  невозмущенного управления  $\mathbf{U}(t)$  справедливо соотношение

$$J_0(\mathbf{U} + \delta \mathbf{U}) = J_0(\mathbf{U}) + \int_0^T \mathbf{W}_0^T(t) \delta \mathbf{U}(t) dt + O(\|\delta \mathbf{U}\|),$$

то  $\mathbf{W}_0(t)$  называется функциональной производной в смысле Фреше функционала  $J_0(\mathbf{U})$  на невозмущенной траектории  $\mathbf{U}(t)$  и обозначается

$$\mathbf{W}_0(t) = \frac{\partial J_0(\mathbf{U})}{\partial \mathbf{U}}. \quad (3)$$

Здесь  $\mathbf{W}_0(t)$  —  $(m * 1)$ -вектор-столбец,  $\mathbf{W}_0^T(t)$  —  $(1 * m)$ -вектор-строка (транспонированный вектор  $\mathbf{W}_0(t)$ ),  $\|*\|$  — некоторая векторная норма.

Техника дифференцирования функционалов, определенных на траекториях динамической системы, достаточно сложна и ее рассмотрение выходит за рамки данного курса. Будем полагать, однако, что мы умеем вычислять функциональные производные (3).

Заметим, что [метод вариаций в пространстве управлений](#) применим и к функционалам, отличным от функционала (2), например, к функционалу вида

$$J_0(\mathbf{U}) = \int_0^T |f_0(\mathbf{X}(t), \mathbf{U}(t))| dt.$$

По существу, при этом изменяется лишь техника вычисления функциональных производных.

В [методе вариаций в пространстве управлений](#) на каждой итерации вариация  $\delta \mathbf{U}$  управления  $\mathbf{U}(t)$  определяется путем минимизации линейной части приращения функционала  $J_0(\mathbf{U})$ , вызванного этой вариацией:

$$\min_{\delta \mathbf{U} \in \delta D_U} \delta J_0(\delta \mathbf{U}) = \min_{\delta \mathbf{U} \in \delta D_U^0} \int_0^T \mathbf{W}_0^T(t) \delta \mathbf{U}(t) dt. \quad (4)$$

Здесь  $\delta D_U$  — некоторая малая окрестность невозмущенного управления  $\mathbf{U}(t)$ .

Окрестность  $\delta D_U$  имеет важное технологическое значение — удачное построение этой окрестности может значительно повысить вычислительную эффективность метода. Однако задача построения этой окрестности однозначного решения не имеет.

При построении множества  $\delta D_U$  следует учитывать следующие требования:

Из того факта, что  $\delta \mathbf{U}(t) \in \delta D_U$ , должно следовать, что  $\mathbf{U}(t) + \delta \mathbf{U}(t) \in D_U$ ;

Множество  $\delta D_U$  должно быть достаточно малой окрестностью траектории  $\mathbf{U}(t)$ , чтобы линейная часть  $\delta J_0(\delta \mathbf{U})$  приращения функционала  $\Delta J_0 = J_0(\mathbf{U} + \delta \mathbf{U}) - J_0(\mathbf{U})$  достаточно точно описывала это приращение;

Множество  $\delta D_U$  должно быть достаточно большой окрестностью траектории  $\mathbf{U}(t)$ , чтобы сходимость управления к оптимальному управлению не была слишком медленной;

Множество  $\delta D_U$  должно быть полной окрестностью невозмущенного управления  $\mathbf{U}(t)$ .

Окрестность  $\delta D_U$  траектории  $\mathbf{U}(t)$  называется полной, если для любой допустимой вариации управления  $\mathbf{v}(t)$  (т.е. такой вариации, что  $\mathbf{U}(t) + \mathbf{v}(t) \in D_U$ ) существует такое число  $s^* > 0$ , что  $s\mathbf{v}(t) \in \delta D_U$  для всех  $t \in [0, T]$  и для всех  $0 \leq s \leq s^*$ . Понятие полной окрестности формализует требование полноты допустимых вариаций — окрестность  $\delta D_U$  должна содержать вариации невозмущенного управления во всех допустимых направлениях.

Общая схема метода вариаций в пространстве управлений.

Из каких либо соображений задаем начальное приближение к оптимальному управлению  $\mathbf{U}^0(t) \in D_U$  и полагаем счетчик числа итераций равным .

С управлением  $\mathbf{U}^0(t)$  решаем задачу Коши для системы ОДУ (1) — получаем фазовую траекторию  $\mathbf{X}^r(t)$ .

Вычисляем  $J_0^r(\mathbf{U})$  — значение [критерия качества управления](#) (2) на невозмущенной траектории  $\mathbf{U}^r(t)$ .

В окрестности невозмущенной траектории  $\mathbf{U}^r(t)$  выполняем линеаризацию задачи — вычисляем функциональную производную

$$\mathbf{W}_0^r(t) = \frac{\partial J_0^r(\mathbf{U})}{\partial \mathbf{U}}$$

и определяем окрестность  $\delta D_U^r$  невозмущенной траектории.  
Из условия

$$\min_{\delta \mathbf{U}} \delta J_0^r(\delta \mathbf{U}) = \min_{\delta \mathbf{U}} \int_0^T (\mathbf{W}_0^r(t))^T \delta \mathbf{U}(t) dt, \quad \delta \mathbf{U} \in \delta \mathcal{D}_U^r \quad (5)$$

находим приращение  $\delta \mathbf{U}^r$  управления  $\mathbf{U}^r(t)$ .

$$\text{Полагаем } \mathbf{U}^{r+1}(t) = \mathbf{U}^r(t) + \delta \mathbf{U}^r.$$

Если условие окончания итераций выполнено (см. ниже), то в качестве приближения к оптимальному управлению принимаем управление  $\mathbf{U}^{r+1}(t)$  и заканчиваем вычисления. Иначе – полагаем  $r = r + 1$  и переходим к п.2 •

В качестве условия окончания итераций естественно принять условие

$$\|\delta \mathbf{U}^r\| \leq \varepsilon_U,$$

где  $\|\cdot\|$  — некоторая функциональная норма,  $\varepsilon_U$  заданная константа.

Заметим, что задача (5) может быть сведена к [задаче линейного программирования](#), что является значительным достоинством метода.

## 12.6. Метод динамического программирования Беллмана

Идею [метода динамического программирования Беллмана](#) рассмотрим на примере [задачи оптимального быстродействия](#)

$$\begin{cases} \dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{U}(t)), \\ t \in [0, T], \mathbf{X}(0) = \mathbf{X}^0, \mathbf{X}(T) = \mathbf{X}^T, \mathbf{U}(t) \in D_U, \mathbf{X}(t) \in D_X, \end{cases} \quad (1)$$

$$J_0(\mathbf{X}, \mathbf{U}) = T \rightarrow \min_{\mathbf{U}(t) \in D_U} \quad (2)$$

Гипотеза 1. Какова бы ни была отличная от  $\mathbf{X}^T$  допустимая точка фазового пространства, существует оптимальная (в смысле быстродействия) траектория перехода динамической системы из точки  $\mathbf{X}$  в точку  $\mathbf{X}^T$  (см. рис. 1) •

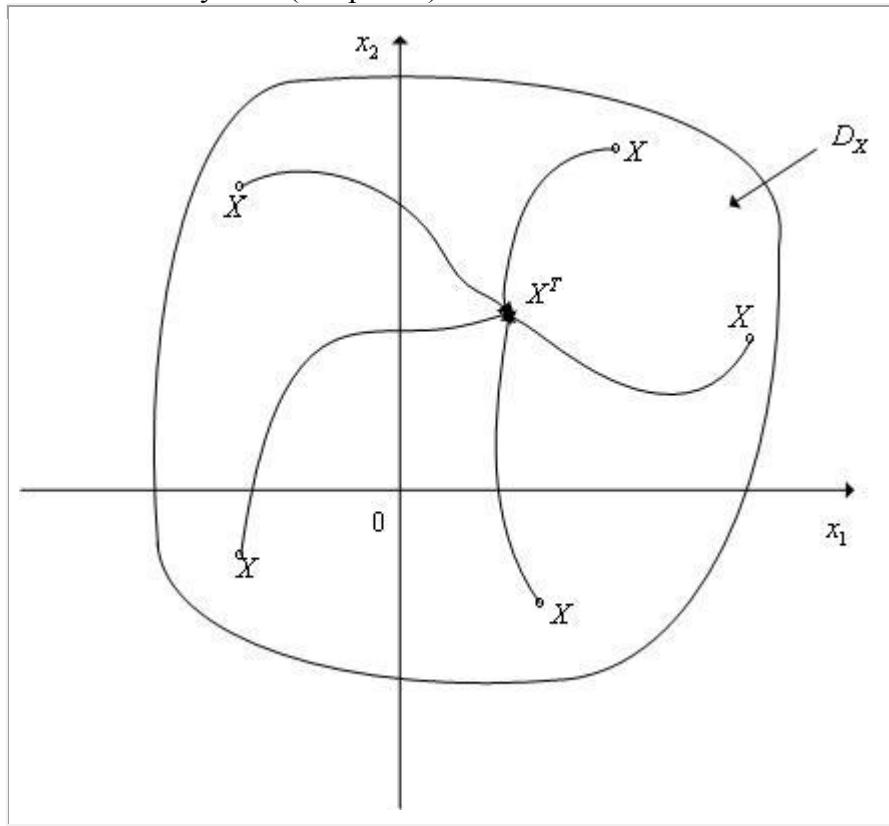


Рис. 1. К гипотезе 1. ( $n = 2$ ).

Время, в течение которого осуществляется оптимальный переход из точки  $\mathbf{X}$  в точку  $\mathbf{X}^T$ , обозначим  $T(\mathbf{X})$ .

В этих терминах гипотеза 1 может быть сформулирована следующим образом. Для любой точки  $\mathbf{X} \in D_X$  существует такое управление  $\mathbf{U} \in D_U$ , под действием которого динамическая система (1) переходит за время, равное  $T(\mathbf{X})$ , из точки  $\mathbf{X}$  в точку  $\mathbf{X}^T$ , но за время, меньшее, чем  $T(\mathbf{X})$ , перейти из точки  $\mathbf{X}$  в точку  $\mathbf{X}^T$  невозможно.

Гипотеза 2. Функция  $T(\mathbf{X})$  непрерывна и всюду, кроме, быть может, точки  $\mathbf{X}^T$ , имеет непрерывные частные производные

$$\frac{\partial T}{\partial x_1}, \frac{\partial T}{\partial x_2}, \dots, \frac{\partial T}{\partial x_n} •$$

Пусть теперь  $(\mathbf{X}^*(t), \mathbf{U}^*(t))$  — оптимальный процесс перехода системы (1) из состояния  $\mathbf{X}^0$  в состояние  $\mathbf{X}^T$  и  $(t_k - t_0)$  — отрезок времени, в течение которого этот переход происходит;  $t_k \leq T$ . Другими словами, положим, что

$$\mathbf{X}^*(t) = \mathbf{X}^T,$$

$$t_k = t_0 + T(\mathbf{X}^0).$$

Утверждение 1 ([принцип оптимальности](#)). Если процесс  $(\mathbf{X}^*(t), \mathbf{U}^*(t)), t \in [t_0, t_k]$  оптimalен, то процесс  $(\mathbf{X}^*(t), \mathbf{U}^*(t)), t \in [t, t_k]$  также оптimalен.

Доказательство (см. рис. 2). Движение по рассматриваемой оптимальной траектории от точки  $\mathbf{X}^0$  до точки  $\mathbf{X}^*(t)$  осуществляется за время  $(t - t_0)$ , а движение из точки  $\mathbf{X}^*(t)$  до точки  $\mathbf{X}^T$  — в течение времени  $(t_k - t) = T(\mathbf{X}^0) - (t - t_0)$ . Быстрее, чем за это время из точки  $\mathbf{X}^*(t)$  попасть в точку  $\mathbf{X}^T$  невозможно.

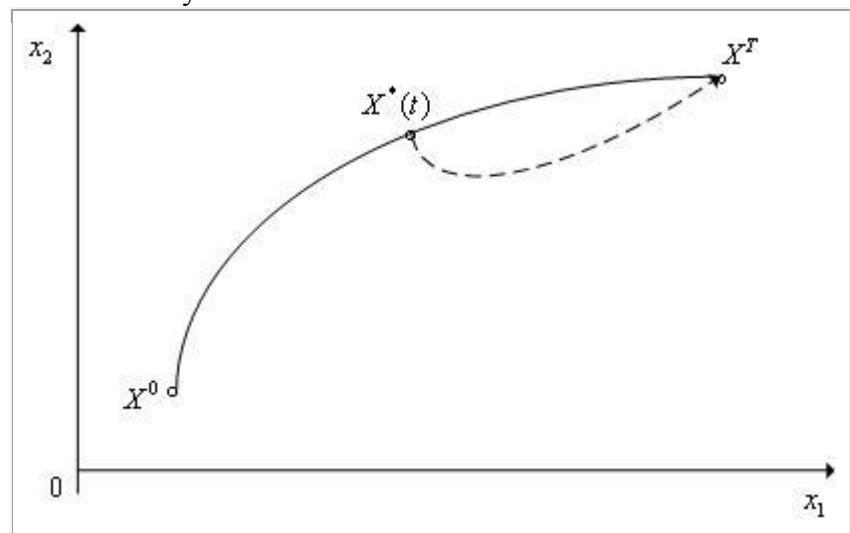


Рис. 2. К утверждению 1. ( $n = 2$ ).

Действительно, если бы такое движение существовало (пунктир на рис. 2), то переместившись из точки  $\mathbf{X}^0$  в точку  $\mathbf{X}^*(t)$  за время  $(t - t_0)$ , а затем из точки  $\mathbf{X}^*(t)$  в точку  $\mathbf{X}^T$  быстрее, чем за время  $(t_k - t) = T(\mathbf{X}^0) - (t - t_0)$ , мы осуществили бы переход из  $\mathbf{X}^0$  в  $\mathbf{X}^T$  за время, меньшее, чем  $T(\mathbf{X}^0)$ , что противоречит предположению об оптимальности процесса  $\mathbf{X}^*(t), \mathbf{U}^*(t)$ .

Утверждение 2. Если процесс  $(\mathbf{X}^*(t), \mathbf{U}^*(t)), t \in [t_0, t_k]$  оптimalен, то справедливо уравнение

$$\sum_{i=1}^n \frac{\partial B(\mathbf{X}^*(t))}{\partial x_i} f_i(\mathbf{X}^*(t), \mathbf{U}^*(t)) = 1,$$

где  $B(\mathbf{X}(t)) = -T(\mathbf{X}(t))$  называется функцией Беллмана.

Доказательство. Из [принципа оптимальности](#) следует, что  $T(\mathbf{X}^0) - (t - t_0)$  есть время оптимального движения из точки  $\mathbf{X}^*(t)$  в точку  $\mathbf{X}^T$ , т.е.

$$T(\mathbf{X}^0) - (t - t_0) = T(\mathbf{X}^*(t)). \quad (3)$$

Заменив в формуле (3)  $T$  на  $B$ , получим

$$B(\mathbf{X}^*(t)) = B(\mathbf{X}(t_0)) + (t - t_0)$$

или

$$\frac{B(\mathbf{X}^*(t)) - B(\mathbf{X}(t_0))}{t - t_0} = 1. \quad (4)$$

Переходя в формуле (4) к пределу при  $t \rightarrow t_0$ , получим, что на оптимальной траектории выполняется равенство

$$\left. \frac{d}{dt} B(\mathbf{X}^*(t)) \right|_{t=t_0} = 1. \quad (5)$$

По правилам дифференцирования сложной функции с учетом уравнения (1) из равенства (5) имеем

$$\left. \frac{d}{dt} B(\mathbf{X}^*(t)) \right|_{t=t_0} = \sum_{i=1}^n \left. \frac{\partial B(\mathbf{X}^*(t))}{\partial x_i} \right|_{t=t_0} f_i(\mathbf{X}^*(t), \mathbf{U}^*(t)) = 1. \quad (6)$$

Из [принципа оптимальности](#) следует, что соотношение (6) верно для любых  $t \in [t_0, t_k]$ . Таким образом, окончательно имеем

$$\sum_{i=1}^n \left. \frac{\partial B(\mathbf{X}^*(t))}{\partial x_i} \right|_{t=t_0} f_i(\mathbf{X}^*(t), \mathbf{U}^*(t)) = 1$$

•

Аналогично утверждению 2 можно доказать справедливость следующего утверждения.

**Утверждение 3.** Если процесс  $(\mathbf{X}^*(t), \mathbf{U}^*(t)), t \in [t_0, t_k]$  оптimalен, то справедливо [уравнение динамического программирования Беллмана для непрерывной системы](#)

$$\max_{\mathbf{U} \in D} \sum_{i=1}^n \left. \frac{\partial B(\mathbf{X}(t))}{\partial x_i} \right|_{U_i} f_i(\mathbf{X}(t), \mathbf{U}(t)) = 1 \quad (7)$$

[Методом динамического программирования Беллмана](#) называется метод решения [задач оптимального управления](#), использующий [принцип оптимальности](#) или [уравнение динамического программирования Беллмана](#).

Подчеркнем следующие обстоятельства:

уравнение динамического программирования Беллмана дает необходимое условие минимума;

уравнение динамического программирования Беллмана требует выполнения гипотезы

12.6.2 относительно неизвестной функции Беллмана  $B(\mathbf{X}(t))$ . Однако, даже в простейших [задачах оптимального управления](#) функция  $B(\mathbf{X}(t))$  оказывается не всюду дифференцируемой. По этой причине при решении задач оптимального управления методом динамического программирования уравнение (7) в явном виде не используется — используется [принцип оптимальности](#).

## 12.7. Решение задачи оптимального управления методом динамического программирования Беллмана

Рассмотрим (только для простоты записи) [задачу оптимального управления](#) для стационарной динамической системы и [критерием оптимальности управления](#), зависящим только от вектора управления:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t)), \\ t \in [0, T], \mathbf{x}(0) = \mathbf{x}^0, \mathbf{x}(T) = \mathbf{x}^T, \mathbf{u}(t) \in D_u, \end{cases} \quad (1)$$

$$J_0(\mathbf{u}) = \int_0^T f_0(\mathbf{u}(t)) dt \rightarrow \min_{\mathbf{u}(t) \in D_u}. \quad (2)$$

Обратим внимание на то, что, в отличие от того, как это делалось ранее, для обозначения  $(n * 1)$  фазового вектора использована маленькая буква  $\mathbf{x}$ , а для обозначения  $(m * 1)$  вектора управления – маленькая буква  $\mathbf{u}$ .

Покроем интервал  $[0, T]$  сеткой  $t_k, k \in [0, N]$  с шагом  $\tau$  (см. рис. 1).

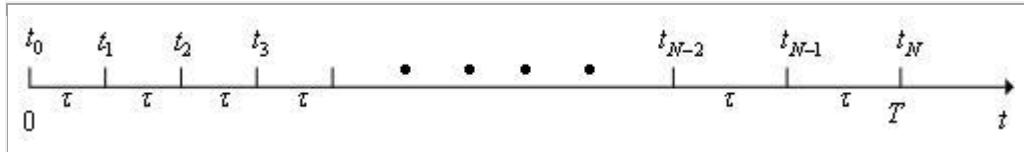


Рис. 1. Равномерная временная сетка на интервале  $[0, T]$ .

Систему ОДУ (1) заменим ее конечно-разностным аналогом

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \tau \mathbf{F}(\mathbf{x}_{k-1}, \mathbf{u}_k), \mathbf{x}_0 = \mathbf{x}^0, \mathbf{x}_N = \mathbf{x}^T, \mathbf{u}_k \in D_u, k \in [1, N], \quad (3)$$

а функционал (2) заменим его приближенным значением, вычисленным по формуле прямоугольников

$$J_0(\mathbf{U}) = \sum_{i=1}^N f_0(\mathbf{u}_k) \tau \rightarrow \min_{\mathbf{u}_k \in D_u}, \quad (4)$$

где  $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N)$  есть  $(m * N)$ -матрица.

Таким образом, [задача оптимального управления](#) (1), (2) в дискретной форме имеет вид (3), (4).

Аналогично матрице  $\mathbf{U}$  введем в рассмотрение  $(n * N)$ -матрицу  $\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{N-1})$  и сформулируем [принцип оптимальности](#) (см. параграф 12.6) для задачи (3), (4).

Утверждение 1 ([принцип оптимальности](#) для дискретной системы). Пусть  $\mathbf{U}^* = (\mathbf{u}_1^*, \mathbf{u}_2^*, \dots, \mathbf{u}_N^*) = \mathbf{U}_1^*$  — оптимальное управление для [задачи оптимального управления](#) (3), (4) и пусть  $\mathbf{X}^* = (\mathbf{x}_0^* = \mathbf{x}^0, \mathbf{x}_1^*, \dots, \mathbf{x}_N^* = \mathbf{x}^T) = \mathbf{X}_0^*$  — соответствующая оптимальная фазовая траектория. Тогда для любых  $k \in [2, N]$  управление  $\mathbf{U}_k^* = (\mathbf{u}_k^*, \mathbf{u}_{k+1}^*, \dots, \mathbf{u}_N^*)$  и соответствующая траектория  $\mathbf{X}_{k-1}^* = (\mathbf{x}_{k-1}^*, \mathbf{x}_k^*, \dots, \mathbf{x}_N^* = \mathbf{x}^T)$  будут оптимальными на интервале времени  $[t_{k-1}, t_N]$ .

Другими словами, если траектория  $\mathbf{x}_0^*$  оптимальна, то и любая ее завершающая часть, начинающаяся из точки  $\mathbf{x}_{k-1}^*$ , будет оптимальной на последних  $N - k + 1$  шагах. А всякая другая траектория из того же состояния, вообще говоря, не является оптимальной на этих шагах (см. рис. 2).

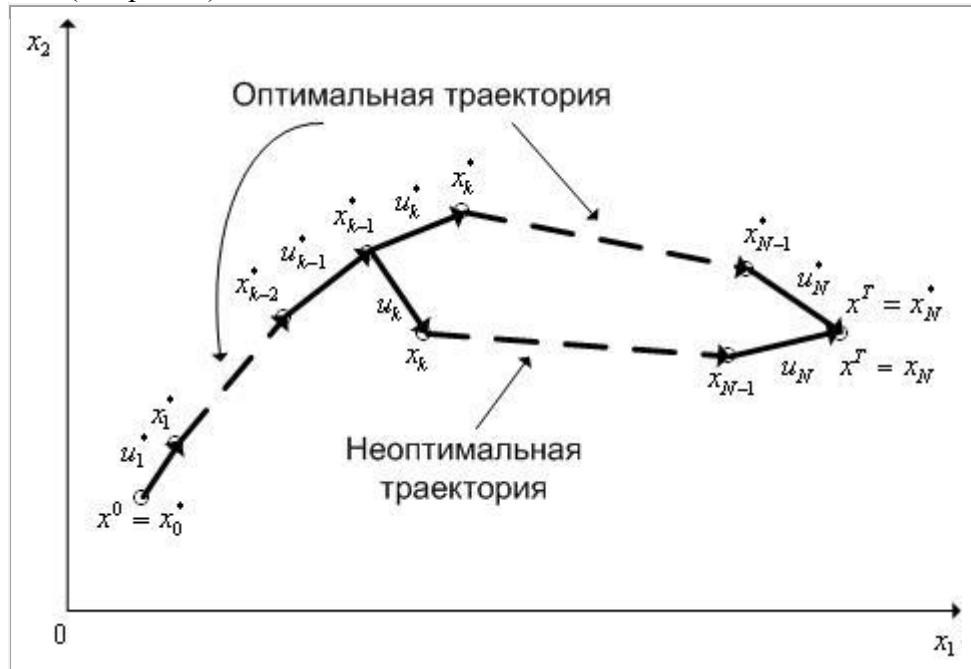


Рис. 2. К принципу оптимальности для дискретной системы.  $n = 2$ .

Обозначим значение функционала (4) на завершающих  $N - k + 1$  шагах

$$J_0(\mathbf{U}_k) = \tau \sum_{l=k}^N f_0(\mathbf{u}_l), \quad k \in [1, N-l].$$

Тогда если на завершающих  $N - k + 1$  шагах управление оптимально, имеет место равенство

$$J_0(\mathbf{U}_k^*) = \tau \sum_{l=k}^N f_0(\mathbf{u}_l^*) = B_k(\mathbf{x}_{k-l}), \quad (5)$$

где  $B_k(\mathbf{x}_{k-1})$  — функция Беллмана для дискретной системы последних  $N - k + 1$  шагов для дискретной [задачи оптимального управления](#) (3), (4).

Из утверждения 1 следует, что на последнем шаге (когда  $k = N$ )

$$B_N(\mathbf{x}_{N-1}) = \min_{\mathbf{u}_N \in D_U} J_0(\mathbf{u}_N). \quad (6)$$

Найдем рекуррентное соотношение, связывающее между собой функции  $B_k(\mathbf{x}_{k-1})$ ,  $B_{k+1}(\mathbf{x}_k)$ . Положим для этого, что функция  $B_{k+1}(\mathbf{x}_k)$  известна. Тогда если на  $k$ -ом шаге с начальным состоянием  $\mathbf{x}_{k-1}$  выбрать управление  $\mathbf{u}_k$ , то процесс перейдет в состояние  $\mathbf{x}_k$  (начальное для последующих  $N - k$  шагов). Если этот переход оптimalен, то опять же из утверждения 1 следует искомое соотношение

$$\min_{\mathbf{u}_k \in D_U} J_0(\mathbf{u}_k) = B_k(\mathbf{x}_{k-1}) = \min_{\mathbf{u}_k \in D_U} (B_{k+1}(\mathbf{x}_k) + \tau f_0(\mathbf{x}_{k+1}, \mathbf{u}_k)). \quad (7)$$

Уравнения (6), (7) позволяют последовательно найти функции  $B_N(\mathbf{x}_{N-1}), B_{N-1}(\mathbf{x}_{N-2}), \dots, B_1(\mathbf{x}_0)$  и называются уравнениями динамического программирования Беллмана для дискретной системы (3), (4). Отметим, что одновременно с нахождением функций  $B_N(\mathbf{x}_{N-1}), B_{N-1}(\mathbf{x}_{N-2}), \dots, B_1(\mathbf{x}_0)$  оказываются определенными и управления  $\mathbf{u}_k^*(\mathbf{x}_{k-1}), k = 1, 2, \dots, N$ . Поскольку управление  $\mathbf{u}_k^*(\mathbf{x}_{k-1}), k = 1, 2, \dots, N$  зависит от состояния  $\mathbf{x}_{k-1}$ , это управление называется условно оптимальным управлением.

После нахождения условно оптимальных управлений  $\mathbf{u}_k^*(\mathbf{x}_{k-1}), k = 1, 2, \dots, N$  можно найти искомые управлении  $\mathbf{u}_1^*, \mathbf{u}_2^*, \dots, \mathbf{u}_N^*$  по следующей схеме:

поскольку состояние  $\mathbf{x}_0 = \mathbf{x}_0^* = \mathbf{x}^0$  известно, находим управление  $\mathbf{u}_1^* = \mathbf{u}_1^*(\mathbf{x}_0)$ ; с этим управлением по формуле (3) находим состояние  $\mathbf{x}_1^* = \mathbf{x}_0^* + \tau \mathbf{F}(\mathbf{x}_0^*, \mathbf{u}_1^*)$ ;

поскольку состояние  $\mathbf{x}_1^*$  известно, находим управление  $\mathbf{u}_2^* = \mathbf{u}_2^*(\mathbf{x}_1^*)$ ; с этим управлением по формуле (3) находим состояние  $\mathbf{x}_2^* = \mathbf{x}_1^* + \tau \mathbf{F}(\mathbf{x}_1^*, \mathbf{u}_2^*)$ ;

...

поскольку состояние  $\mathbf{x}_{N-1}^*$  известно, находим управление  $\mathbf{u}_N^* = \mathbf{u}_N^*(\mathbf{x}_{N-1}^*)$ ; очевидно, что  $\mathbf{x}_N^* = \mathbf{x}^T = \mathbf{x}_{N-1}^* + \tau \mathbf{F}(\mathbf{x}_{N-1}^*, \mathbf{u}_N^*)$ .

Схема метода приближённого решения задач оптимального управления методом динамического программирования Беллмана.

Этап 1

Шаг 1. Из условия (6) находим условно оптимальное управление  $\mathbf{u}_N^*(\mathbf{x}_{N-1})$  и функцию Беллмана  $B_N(\mathbf{x}_{N-1}^*)$ .

Шаг 2. Используя результаты предыдущего шага, из условия (7) находим условно оптимальное управление  $\mathbf{u}_{N-1}^*(\mathbf{x}_{N-2})$  и функцию Беллмана  $B_{N-1}(\mathbf{x}_{N-2}^*)$ .

...

Шаг N. Используя результаты предыдущего шага, из условия (7) находим условно оптимальное управление  $\mathbf{u}_1^*(\mathbf{x}_0)$  и функцию Беллмана  $B_1(\mathbf{x}_0^*)$ .

Этап 2.

Шаг 1. Находим управление  $\mathbf{u}_1^* = \mathbf{u}_1^*(\mathbf{x}_0)$  и состояние  $\mathbf{x}_1^* = \mathbf{x}_0^* + \tau \mathbf{F}(\mathbf{x}_0^*, \mathbf{u}_1^*)$ ;

Шаг 2. Находим управление  $\mathbf{u}_2^* = \mathbf{u}_2^*(\mathbf{x}_1)$  и состояние  $\mathbf{x}_2^* = \mathbf{x}_1^* + \tau \mathbf{F}(\mathbf{x}_1^*, \mathbf{u}_2^*)$ ;

...

Шаг N. Находим управление  $\mathbf{u}_N^* = \mathbf{u}_N^*(\mathbf{x}_{N-1}^*)$  и состояние  $\mathbf{x}_N^* = \mathbf{x}^T = \mathbf{x}_{N-1}^* + \tau \mathbf{F}(\mathbf{x}_{N-1}^*, \mathbf{u}_N^*)$ .

Заметим, что при приближённом решении задач оптимального управления методом динамического программирования остается открытым вопрос о сходимости решения

$\mathbf{u}_1^*, \mathbf{u}_2^*, \dots, \mathbf{u}_N^*$  к решению  $\mathbf{U}^*(t)$  исходной непрерывной задачи оптимального управления (1), (2).

## 12.8. Решение задачи оптимального управления методом сведения к задаче нелинейного программирования

Рассмотрим [задачу оптимального управления](#)

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t)), t \in [0, T], \\ \mathbf{x}(0) = \mathbf{x}^0, \mathbf{x}(T) = \mathbf{x}^T, \\ \mathbf{u}(t) \in D_U, \end{cases} \quad (1)$$

$$J_0(\mathbf{u}) = \int_0^T f_0(\mathbf{x}(t), \mathbf{u}(t)) dt \rightarrow \min_{\mathbf{u} \in D_U}. \quad (2)$$

Обратим внимание на то, что так же, как в предыдущем параграфе, для обозначения ( $n * 1$ ) фазового вектора использована маленькая буква  $\mathbf{x}$ , а для обозначения ( $m * 1$ ) вектора управления – маленькая буква  $\mathbf{u}$ .

Метод решения [задачи оптимального управления](#) (1), (2) путем сведения этой задачи к [задаче нелинейного программирования](#) ([метод сведения к задаче нелинейного программирования](#)) рассматривался (на одном частном примере) в главе 1. Изложим общую схему этого метода.

Покроем интервал  $[0, T]$  равномерной или неравномерной сеткой  $t_k, k \in [0, N]$  с шагом  $\tau_k$  (см. рис. 1).

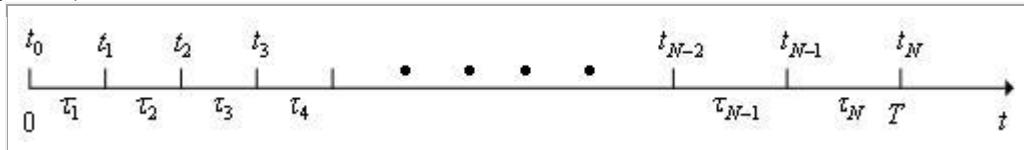


Рис. 1. Временная сетка  $t_k, k \in [0, N]$  на интервале  $[0, T]$ .

Систему ОДУ (1) заменим ее конечно-разностным аналогом

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \tau_k \mathbf{F}(\mathbf{x}_{k-1}, \mathbf{u}_k), \mathbf{x}_0 = \mathbf{x}^0, \mathbf{x}_N = \mathbf{x}^T, \mathbf{u}_k \in D_U, k \in [1, N], \quad (3)$$

а [критерий качества управления](#) (2) заменим его приближенным значением, вычисленным по формуле прямоугольников

$$J_0(\mathbf{U}) = \sum_{k=1}^N f_0(\mathbf{x}_k, \mathbf{u}_k) \tau_k \rightarrow \min_{\mathbf{u}_k \in D_U}, \quad (4)$$

где  $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N)$  есть  $(m * N)$ -матрица.

Дискретная [задача оптимального управления](#) (3), (4) представляет собой [задачу нелинейного программирования](#)

$$\min_{\mathbf{U} \in D_U} J_0(\mathbf{U}) = J_0(\mathbf{U}^*) = J_0^*, \quad (5)$$

где, в отличие от рассматриваемой ранее постановки задачи оптимизации, используется не [вектор варьируемых параметров](#), а матрица [варьируемых параметров](#). Очевидно, что легко перейти от  $(m * N)$  матрицы варьируемых параметров  $\mathbf{U}$  к привычному для нас  $(Nm * 1)$  вектору варьируемых параметров.

Приведем схему вычисления значений [критерия оптимальности управления](#)  $J_0(\mathbf{U})$ . Пусть тем или иным способом заданы значения элементов  $(m * N)$  матрицы  $\mathbf{U}$ . Тогда для вычисления соответствующего значения критерия  $J_0(\mathbf{U})$  необходимо выполнить следующие действия:

при управлении  $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_1, \dots, \mathbf{u}_N)$  по формуле (3) последовательно вычисляем значения компонент вектора  $\mathbf{x}_k$  для  $k = 1, 2, \dots, N$ ;

с найденными векторами  $\mathbf{x}_k, k = 1, 2, \dots, N$  и управлением  $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_1, \dots, \mathbf{u}_N)$  по формуле (5) вычисляем значение [критерия оптимальности управления](#)  $J_0(\mathbf{U})$ .

[Задача нелинейного программирования](#) (5) может быть решена рассмотренными ранее [методами условной оптимизации](#).

Изложенный метод создает впечатление тривиальности решения [задач оптимального управления](#). Действительно, есть теоремы о том, что решение дискретной задачи (3), (4) сколь угодно точно (при  $N \rightarrow \infty$ ) аппроксимирует решение исходной задачи (1), (2). Есть теоремы о сходимости методов оптимизации, с помощью которых может быть найден минимум (4). Однако всегда остаются открытыми вопросы: можно ли данное  $N$  считать достаточно большим?; можно ли ограничиться данным числом итераций при решении задачи (4)? Т.е. необходим тщательный содержательный контроль результатов. Иначе легко получить решения, сколь угодно далекие от действительно оптимальных решений. В целом, данный метод (впрочем, как и любой другой) может быть рекомендован только в комбинации с другими методами.

## 12.9. Выбор оптимальной стратегии замены оборудования как задача динамического программирования

В общем виде проблема ставится следующим образом: определить оптимальную стратегию использования оборудования в период времени длительностью  $m$  лет, причем прибыль за каждые  $i$  лет,  $i = \overline{1, m}$ , от использования оборудования возраста  $t$  лет должна быть максимальной.

Известны:  $r(t)$  — выручка от реализации продукции, произведенной за год на оборудовании возраста  $t$  лет;  $l(t)$  — годовые затраты, зависящие от возраста оборудования  $t$ ;  $c(t)$  — остаточная стоимость оборудования возраста  $t$  лет;  $p$  — стоимость нового оборудования. Под возрастом оборудования понимается период эксплуатации оборудования после последней замены, выраженный в годах.

Для построения математической модели последовательно выполняются этапы, сформулированные ниже.

1. Определение числа шагов. Число шагов равно числу лет, в течение которых эксплуатируется оборудование.

2. Определение состояний системы. Состояние системы характеризуется возрастом оборудования  $t$ ,  $t = \overline{0, m}$ .

3. Определение управлений. В начале  $i$ -го шага,  $i = \overline{1, m}$ , может быть выбрано одно из двух управлений: заменять или не заменять оборудование. Каждому варианту управления приписывается число

$$x_i = \begin{cases} 0, & \text{если оборудование не заменяется;} \\ 1, & \text{если оборудование заменяется.} \end{cases} \quad (3.7)$$

4. Определение функции выигрыша на  $i$ -м шаге. Функция выигрыша на  $i$ -м шаге — это прибыль от использования оборудования к концу  $i$ -го года эксплуатации,  $t = \overline{0, m}$ ,  $i = \overline{1, m}$ .

$$f_i(t) = \begin{cases} r(t) - l(t), & \text{если оборудование в начале } i\text{-го года} \\ & \text{не заменяется;} \\ c(t) - p + r(0) - l(0), & \text{если оборудование заменяется.} \end{cases} \quad (3.8)$$

Таким образом, если оборудование не продается, то прибыль от его использования — это разность между стоимостью произведенной продукции и эксплуатационными издержками. При замене оборудования прибыль составляет разность между остаточной стоимостью оборудования и стоимостью нового оборудования, к которой прибавляется разность между стоимостью продукции и эксплуатационными издержками для нового оборудования, возраст которого в начале  $i$ -го шага составляет 0 лет.

5. Определение функции изменения состояния:

$$\varphi_i(t) = \begin{cases} t + 1, & \text{если } x_i = 0; \\ 1, & \text{если } x_i = 1. \end{cases} \quad (3.9)$$

6. Составление функционального уравнения для  $i = m$ :

$$B_m(t) = \max_{x_m \in \{0,1\}} \begin{cases} r(t) - l(t); \\ c(t) - p + r(0) - l(0). \end{cases} \quad (3.10)$$

7. Составление основного функционального уравнения:

$$B_i(t) = \max_{x_i \in \{0,1\}} \begin{cases} r(t) - l(t) + B_{i+1}(t+1); \\ c(t) - p + r(0) - l(0) + B_{i+1}(1), \end{cases} \quad (3.11)$$

где  $B_i(t)$  — прибыль от использования оборудования возраста  $t$  лет с  $i$ -го шага (с конца  $i$ -го года) до конца периода эксплуатации;  $B_{i+1}(t+1)$  — прибыль от использования оборудования возраста  $t+1$  год с  $(i+1)$ -го шага до конца периода эксплуатации.

Математическая модель задачи построена.

**Пример.** Определить оптимальную стратегию использования оборудования в период времени длительностью  $m$  лет, причем прибыль за каждые  $i$  лет,  $i = 1, m$ , от использования оборудования возраста  $t$  лет должна быть максимальной. Известны:  $r(t)$  — выручка от реализации продукции, произведенной за год на оборудовании возраста  $t$  лет;  $l(t)$  — годовые затраты, зависящие от возраста оборудования  $t$ ;  $c(t)$  — остаточная стоимость оборудования возраста  $t$  лет;  $p$  — стоимость нового оборудования. Под возрастом оборудования понимается период эксплуатации оборудования после последней замены, выраженный в годах.  $m=12$ ,  $p=10$ ,  $c(t)=0$ ,  $r(t) - l(t) = f(t)$ . Значения  $f(t)$  заданы в табл.

Таблица

$t$	0	1	2	3	4	5	6	7	8	9	10	11	12
$f(t)$	10	9	8	7	6	5	4	3	2	1	0	0	0

**Решение 1:** Для данного примера функциональные уравнения будут иметь вид

$$B_m(t) = \max_{x_m \in \{0,1\}} \begin{cases} f(t); \\ -p + f(0). \end{cases}$$

$$B_i(t) = \max_{x_i \in \{0,1\}} \begin{cases} f(t) + B_{i+1}(t+1); \\ -p + f(0) + B_{i+1}(1). \end{cases}$$

Для решения данной задачи заполняется табл. 24. Поясним, как заполняется таблица для нескольких шагов.

1. Условная оптимизация начинается с последнего 12-го шага. Для  $i=12$  рассматриваются возможные состояния системы  $t = 0, 1, 2, \dots, 12$ . Функциональное уравнение на 12-м шаге имеет вид:

$$B_{12}(t) = \max_{x_m \in \{0,1\}} \begin{cases} f(t); \\ -p + f(0). \end{cases}$$

$$1) t=0, B_{12}(0) = \max_{\{0,1\}} \left\{ \frac{10}{-10+10} \right\} = 10, \quad x_{12}(0) = 0.$$

$$2) t=1, B_{12}(1) = \max_{\{0,1\}} \left\{ \frac{9}{-10+10} \right\} = 9, \quad x_{12}(1) = 0. \quad \dots$$

$$10) t=9, B_{12}(9) = \max_{\{0,1\}} \left\{ \frac{1}{-10+10} \right\} = 1, \quad x_{12}(9) = 0.$$

Таблица

$t$	$i = 12$		$i = 11$		$i = 10$		$i = 9$		$i = 8$		$i = 7$		$i = 6$		$i = 5$		$i = 4$		$i = 3$		$i = 2$		$i = 1$	
	$x_{12}$	$B_{12}$	$x_{11}$	$B_{11}$	$x_{10}$	$B_{10}$	$x_9$	$B_9$	$x_8$	$B_8$	$x_7$	$B_7$	$x_6$	$B_6$	$x_5$	$B_5$	$x_4$	$B_4$	$x_3$	$B_3$	$x_2$	$B_2$	$x_1$	$B_1$
0	0	10	0	19	0	27	0	34	0	40	0	45	0	51	0	58	0	64	0	70	0	75	0	<b>82</b>
1	0	9	0	17	<b>0</b>	<b>24</b>	0	30	0	35	0	41	<b>0</b>	<b>48</b>	0	54	0	60	0	65	<b>0</b>	<b>72</b>	0	78
2	0	8	<b>0</b>	<b>15</b>	0	21	0	26	0	32	<b>0</b>	<b>39</b>	0	45	0	51	0	56	<b>0</b>	<b>63</b>	0	69	0	75
3	<b>0</b>	<b>7</b>	0	13	0	18	0	24	<b>0</b>	<b>31</b>	0	37	0	43	0/1	48	<b>0</b>	<b>55</b>	0	61	0	67	0	73
4	0	6	0	11	1	17	<b>1</b>	<b>24</b>	0/1	30	0	36	0/1	41	<b>1</b>	<b>48</b>	0/1	54	0/1	60	0	66	1	72
5	0	5	0/1	9	1	17	1	24	1	30	0/1	35	1	41	1	48	1	54	1	60	0/1	65	1	72
6	0	4	1	9	1	17	1	24	1	30	1	35	1	41	1	48	1	54	1	60	1	65	1	72
7	0	3	1	9	1	17	1	24	1	30	1	35	1	41	1	48	1	54	1	60	1	65	1	72
8	0	2	1	9	1	17	1	24	1	30	1	35	1	41	1	48	1	54	1	60	1	65	1	72
9	0	1	1	9	1	17	1	24	1	30	1	35	1	41	1	48	1	54	1	60	1	65	1	72
10	0/1	0	1	9	1	17	1	24	1	30	1	35	1	41	1	48	1	54	1	60	1	65	1	72
11	0/1	0	1	9	1	17	1	24	1	30	1	35	1	41	1	48	1	54	1	60	1	65	1	72
12	0/1	0	1	9	1	17	1	24	1	30	1	35	1	41	1	48	1	54	1	60	1	65	1	72

В левой колонке таблицы записываются возможные состояния системы  $t = \overline{0,12}$ , в верхней строке — номера шагов  $i = \overline{1,12}$ . Для каждого шага определяются условные оптимальные управления  $x_i(t)$  и условный оптимальный выигрыш  $B_i(t)$  с  $i$ -го шага и до конца для оборудования возраста  $t$  лет.

$$11) t=10, B_{12}(10)=\max_{\{0,1\}} \left\{ \begin{array}{l} 0 \\ -10+10 \end{array} \right\}=0, \quad x_{12}(10)=0, \quad x_{12}(10)=1.$$

$$12) t=11, B_{12}(11)=\max_{\{0,1\}} \left\{ \begin{array}{l} 0 \\ -10+10 \end{array} \right\}=0, \quad x_{12}(11)=0, \quad x_{12}(11)=1.$$

$$13) t=12, B_{12}(12)=\max_{\{0,1\}} \left\{ \begin{array}{l} 0 \\ -10+10 \end{array} \right\}=0, \quad x_{12}(12)=0, \quad x_{12}(12)=1.$$

Таким образом, на 12-м шаге оборудование возраста 0—9 лет заменять не надо. Оборудование возраста 10—12 лет можно заменить или продолжить его эксплуатацию, так как для  $t = 10, 11, 12$  имеется два условных оптимальных управления 1 и 0.

По результатам расчетов заполняются два столбца таблицы, соответствующие  $i = 12$ .

## 2. Условная оптимизация 11-го шага.

Для  $i = 11$  рассматриваются все возможные состояния системы  $t = 0, 1, 2, \dots, 12$ .

Функциональное уравнение на 11-м шаге имеет вид

$$B_{11}(t)=\max_{0,1} \left\{ \begin{array}{l} f(t)+B_{12}(t+1); \\ -p+f(0)+B_{12}(1). \end{array} \right.$$

1)  $t=0$ ,

$$B_{11}(0)=\max_{\{0,1\}} \left\{ \begin{array}{l} f(0)+B_{12}(1) \\ -p+f(0)+B_{12}(1) \end{array} \right.=\max_{\{0,1\}} \left\{ \begin{array}{l} 10+9 \\ -10+10+9 \end{array} \right\}=19, \quad x_{11}(0)=0.$$

2)  $t=1$ ,

$$B_{11}(1)=\max_{\{0,1\}} \left\{ \begin{array}{l} f(1)+B_{12}(2) \\ -p+f(0)+B_{12}(1) \end{array} \right.=\max_{\{0,1\}} \left\{ \begin{array}{l} 9+8 \\ -10+10+9 \end{array} \right\}=17, \quad x_{11}(1)=0.$$

.....

6)  $t=5$ ,

$$B_{11}(5)=\max_{\{0,1\}} \left\{ \begin{array}{l} f(5)+B_{12}(6) \\ -p+f(0)+B_{12}(1) \end{array} \right.=\max_{\{0,1\}} \left\{ \begin{array}{l} 5+4 \\ -10+10+9 \end{array} \right\}=9, \quad x_{11}(5)=0, \\ x_{11}(5)=1.$$

7)  $t=6$ ,

$$B_{11}(6)=\max_{\{0,1\}} \left\{ \begin{array}{l} f(6)+B_{12}(7) \\ -p+f(0)+B_{12}(1) \end{array} \right.=\max_{\{0,1\}} \left\{ \begin{array}{l} 4+3 \\ -10+10+9 \end{array} \right\}=9, \quad x_{11}(6)=1,$$

.....

13)  $t=12$ ,

$$B_{11}(12)=\max_{\{0,1\}} \left\{ \begin{array}{l} f(12) \\ -p+f(0)+B_{12}(1) \end{array} \right.=\max_{\{0,1\}} \left\{ \begin{array}{l} 0 \\ -10+10+9 \end{array} \right\}=9, \quad x_{11}(12)=1.$$

Таким образом, на 11-м шаге не следует заменять оборудование возраста 0—4 года. Для оборудования возраста 5 лет возможны две стратегии использования: заменить или продолжать эксплуатировать.

Начиная с 6-го года оборудование следует заменять. По результатам расчетов заполняются два столбца таблицы, соответствующие  $i = 11$ .

Аналогичным образом заполняются остальные десять столбцов таблицы. При

расчетах  $B_{i+1}(t)$  на каждом шаге значения  $f(t)$  для каждого  $t = \overline{0,12}$  берутся из таблицы исходных данных, приведенной в условии задачи, а значения  $B_i(t)$  — из последнего, заполненного на предыдущем шаге столбца.

Этап условной оптимизации заканчивается после заполнения табл. 24.

Безусловная оптимизация начинается с первого шага.

Предположим, что на первом шаге  $i=1$  имеется новое оборудование, возраст которого 0 лет.

Для  $t = t_1 = 0$  оптимальный выигрыш составляет  $B_1(0) = 82$ . Это значение соответствует максимальной прибыли от использования нового оборудования в течение 12 лет.

$$B^* = B_1(0) = 82.$$

Выигрышу  $B_1(0) = 82$  соответствует безусловное оптимальное управление  $x_1(0) = 0$ .

Для  $i=2$  по формуле (3.9)  $t_2 = t_1 + 1 = 1$ .

Безусловное оптимальное управление  $x_2(1) = 0$ .

Для  $i=3$   $t_3 = t_2 + 1 = 2$ .

Безусловное оптимальное управление  $x_3(2) = 0$ .

И далее соответственно

$$\begin{array}{lll} i=4 & t_4 = t_3 + 1 = 3 & x_4(3) = 0, \\ i=5 & t_5 = t_4 + 1 = 4 & x_5(4) = 1, \\ i=6 & t_6 = 1 & x_6(1) = 0, \\ i=7 & t_7 = t_6 + 1 = 2 & x_7(2) = 0, \\ i=8 & t_8 = t_7 + 1 = 3 & x_8(3) = 0, \\ i=9 & t_9 = t_8 + 1 = 4 & x_9(4) = 1, \\ i=10 & t_{10} = 1 & x_{10}(1) = 0, \\ i=11 & t_{11} = t_{10} + 1 = 2 & x_{11}(2) = 0, \\ i=12 & t_{12} = t_{11} + 1 = 3 & x_{12}(3) = 0. \end{array}$$

Управления, составляющие оптимальную стратегию использования оборудования, выделены в табл. 24 полужирным шрифтом.

В рамках данной задачи оптимальная стратегия заключается в замене оборудования при достижении им возраста 4-х лет. Аналогичным образом можно определить оптимальную стратегию использования оборудования любого возраста.

При решении задач данного типа может использоваться как метод обратной так и метод прямой прогонки. Рассмотрим решение предыдущей задачи, начиная решение с первого шага.

**Решение 2:** Рассмотрим период  $m$  лет, в пределах которого требуется определить оптимальный цикл замены оборудования.

$B_m(t)$  максимальный доход, получаемый от оборудования возраста  $t$  лет за оставшиеся  $m$  лет цикла использования оборудования при условии оптимальной стратегии.

Будем отсчитывать возраст оборудования в направлении течения процесса. Так,  $t = 0$  соответствует случаю использования нового оборудования. Временные же стадии процесса нумеруются в обратном направлении по отношению к ходу процесса. Так,  $m = 1$

относится к одной временной стадии, остающейся до завершения процесса, а  $m = m$  — к началу процесса.



На каждом этапе  $m$ -стадийного процесса должно быть принято решение о сохранении или замене оборудования. Выбранный вариант должен обеспечивать получение максимальной прибыли.

Функциональные уравнения, основанные на принципе оптимальности, имеют вид:

$$B_m(t) = \max_{x_i \in \{0,1\}} \begin{cases} f(t) + B_{m-1}(t+1); \\ -p + f(0) + B_{m-1}(1). \end{cases} \quad \begin{array}{l} \longrightarrow \text{Сохранение} \\ \longrightarrow \text{Замена} \end{array}$$

$$B_1(t) = \max_{x_m \in \{0,1\}} \begin{cases} f(t); \\ -p + f(0). \end{cases} \quad \begin{array}{l} \longrightarrow \text{Сохранение} \\ \longrightarrow \text{Замена} \end{array}$$

Оба уравнения состоят из двух частей: верхняя строка определяет доход, получаемый при сохранении оборудования; нижняя — доход, получаемый при замене оборудования и продолжении процесса работы на новом оборудовании.

Для  $m=1$

$$B_1(0) = \max \begin{cases} f(0); \\ -p + f(0) \end{cases} = \max \begin{cases} 10 \\ -10 + 10 \end{cases} = 10,$$

$$B_1(1) = \max \begin{cases} f(1); \\ -p + f(0) \end{cases} = \max \begin{cases} 9 \\ -10 + 10 \end{cases} = 9,$$

.....

$$B_1(12) = \max \begin{cases} f(12); \\ -p + f(0) \end{cases} = \max \begin{cases} 0 \\ -10 + 10 \end{cases} = 0,$$

Для  $m=2$

$$B_2(0) = \max \begin{cases} f(0) + B_1(1) \\ -p + f(0) + B_1(1) \end{cases} = \max \begin{cases} 10 + 9 \\ -10 + 10 + 9 \end{cases} = 19,$$

$$B_2(1) = \max \begin{cases} f(1) + B_1(2) \\ -p + f(0) + B_1(1) \end{cases} = \max \begin{cases} 9 + 8 \\ -10 + 10 + 9 \end{cases} = 17.$$

Вычисления продолжаем до тех пор, пока не будет выполнено условие  $B_1(1) > B_2(2)$ , т.е. в данный момент оборудование необходимо заменить, так как величина прибыли, получаемая в результате замены оборудования, больше, чем в случае использования старого. Результаты расчетов помещаем в таблицу, момент замены отмечаем звездочкой, после чего дальнейшие вычисления по строчке прекращаем (табл.).

$F_N(t)$	0	1	2	3	4	5	6	7	8	9	10	11	12
	$N$	$N - 1$	11	10	9	8	7	6	5	4	3	2	1
$f_1(t)$	10	9	8	7	6	5	4	3	2	1	0	0	0
$f_2(t)$	19	17	15	13	11	9							9*
$f_3(t)$	27	24	21	18	17		17*						
$f_4(t)$	34	30	26	24		24*							
$f_5(t)$	40	35	32	31	30		30*						
$f_6(t)$	45	41	39	37	36	35		35*					
$f_7(t)$	51	48	45	43	41		41*						
$f_8(t)$	58	54	51	48		48*							
$f_9(t)$	64	60	56	55	54		54*						
$f_{10}(t)$	70	65	63	61	60		60*						
$f_{11}(t)$	75	72	69	67	66	65		65*					
$f_{12}(t)$	82	78	75	73	72		72*						

Можно не решать каждый раз уравнение, а вычисления проводить в таблице.

Например, вычислим  $B_4(t)$ :

$$f_4(0) = f_1(0) + f_3(1) = 10 + 24 = 34 > f_3(1) = 24,$$

$$f_4(1) = f_1(1) + f_3(2) = 9 + 21 = 30 > f_3(1),$$

$$f_4(2) = f_1(2) + f_3(3) = 8 + 18 = 26 > f_3(1),$$

$$f_4(3) = f_1(3) + f_3(4) = 7 + 17 = 24 > f_3(1),$$

$$f_4(4) = f_1(4) + f_3(5) = 6 + 17 = 23 < f_3(1).$$

Дальнейшие расчеты для  $B_4(t)$  прекращаем, так как  $B_4(4) = 23 < B_3(1) = 24$ .

По результатам вычислений и по линии, разграничитывающей области решений сохранения и замены оборудования, находим оптимальный цикл замены оборудования. Для данной задачи он составляет 4 года.

*Ответ.* Для получения максимальной прибыли от использования оборудования в двенадцатиступенчатом процессе оптимальный цикл состоит в замене оборудования через каждые 4 года.

## 12.10. Задача выбора кратчайшего пути

Рассмотрим задачу о расчете кратчайшего пути между двумя городами. На рисунке показаны возможные маршруты, проходящие через разные промежуточные пункты, расстояния нанесены над стрелками. Требуется найти кратчайший путь от пункта 1 к пункту 10. Перебор всех маршрутов возможен, но неэффективен, так как при увеличении размеров сети число маршрутов (в примере их 18) будет лавинообразно возрастать.

Чтобы применить метод динамического программирования, разделим задачу на шаги, которые показаны вертикальными пунктирными линиями. Нумерация шагов выполнена от конца к началу (это обычный порядок для метода динамического программирования), поскольку последующий расчет будет выполняться в таком же порядке. Выделение этапов основано на том, что каждый из них надо пройти, выбрав один из путей, соединяющий один из начальных пунктов этапа с одним из его конечных пунктов.

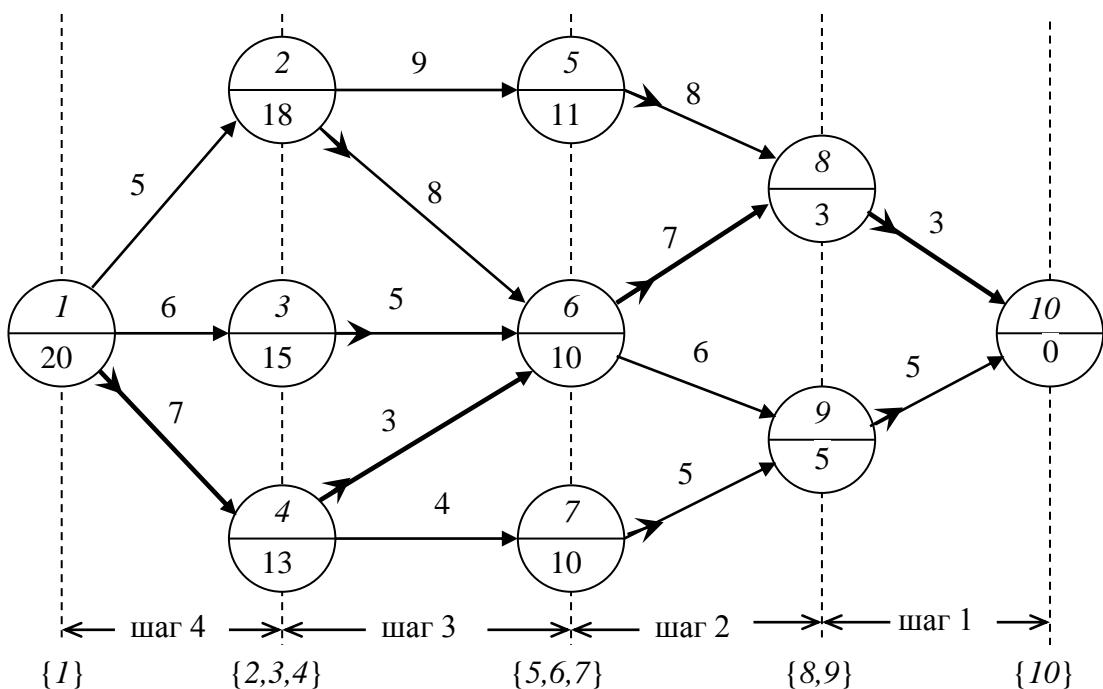


Рис. 6.1

При расчете удобно нижнюю половину круга, соответствующую пункту, использовать для записи расстояния, которое еще осталось пройти от текущего (промежуточного) пункта до конечного. Для 10-го пункта это расстояние равно нулю. Расстояние, которое предстоит пройти на первом шаге, зависит от того, будет ли этот шаг начинаться из пункта 8 или 9. Это пока неизвестно, но каждому из этих двух вариантов начального состояния для шага 1 соответствуют свои очевидные значения оставшихся расстояний, так же как и оптимальные направления дальнейшего движения. Расстояния записываются в круги для пунктов 8 и 9, а направления (их называют условно-оптимальными) изображены короткими стрелками.

На шаге 2 для пунктов 5 и 7 расчет отличается только тем, что оставшееся расстояние получается суммированием расстояния до следующего пункта с расстоянием от последнего до конечного пункта, которое уже известно. Для пункта 6 надо выбирать одно из двух возможных направлений, в пункт 8 (до конечного пункта останется  $3+7=10$ ),

или в пункт 9 (останется  $6+5=11$ ). Выбираем кратчайший путь и показываем условно-оптимальное направление.

Оставшиеся два шага выполняются аналогично. Найдено не только кратчайшее расстояние до конечного пункта при заданной дорожной сети, но и условно-оптимальные направления для всех остальных пунктов, что делает очевидным выделение оптимального маршрута, если за начальный пункт принять любой из промежуточных. Достаточно двигаться от такого пункта по уже выделенным условно-оптимальным направлениям. На рисунке оптимальный маршрут от пункта 1 до пункта 10 выделен жирной линией.

## 12.11. Оптимальное распределение средств на расширение производства

Широкий класс составляют задачи, в которых речь идет о наиболее целесообразном распределении во времени тех или иных ресурсов (денежных средств, рабочей силы, сырья и т.п.). Рассмотрим пример задачи такого рода.

Группе предприятий выделяются дополнительные средства на реконструкцию и модернизацию производства. По каждому из  $n$  предприятий известен возможный прирост  $f_i(x)$ ,  $i = \overline{1, n}$  выпуска продукции в зависимости от выделенной ему суммы  $x$ . Требуется так распределить между предприятиями имеющиеся средства  $c$ , чтобы общий прирост  $B_n(c)$  выпуска продукции был максимальным.

В соответствии с вычислительной схемой динамического программирования рассмотрим сначала случай  $n=1$ , т. е. предположим, что все имеющиеся средства выделяются на реконструкцию и модернизацию одного предприятия. Обозначим через  $B_1(x)$  максимально возможный прирост выпуска продукции на этом предприятии, соответствующий выделенной сумме  $x$ . Каждому значению  $x$  отвечает вполне определенное (единственное) значение  $f_1(x)$  выпуска, поэтому можно записать, что  $B_1(x) = \max(f_1(x)) = f_1(x)$ .

Пусть теперь  $n=2$ , т.е. средства распределяются между двумя предприятиями. Если второму предприятию выделена сумма  $x$ , то прирост продукции на нем составит  $f_2(x)$ . Оставшиеся другому предприятию средства  $c-x$  в зависимости от величины  $x$  (а значит, и  $c-x$ ) позволяют увеличить прирост выпуска продукции до максимально возможного значения  $B_1(c-x)$ . При этом условии общий прирост выпуска продукции на двух предприятиях

$$f_2(x) + B_1(c-x). \quad (1)$$

Оптимальному значению  $B_2(c)$  прироста продукции при распределении суммы  $c$  между двумя предприятиями соответствует такое  $x$ , при котором сумма (1) максимальна. Это можно выразить записью

$$B_2(c) = \max_{0 \leq x \leq c} (f_2(x) + B_1(c-x))$$

Значение  $B_3(c)$  можно вычислить, если известны значения  $B_2(c)$ , и т. д.

Функциональное уравнение Беллмана для рассматриваемой задачи запишется в следующем виде:

$$B_n(c) = \max_{0 \leq x \leq c} (f_n(x) + B_{n-1}(c-x)).$$

Итак, максимальный прирост выпуска продукции на  $n$  предприятиях определяется как максимум суммы прироста выпуска на  $n$ -м предприятии и прироста выпуска на остальных  $n-1$  предприятиях при условии, что оставшиеся после  $n$ -го предприятия средства распределяются между остальными предприятиями оптимально.

Имея функциональные уравнения и , можно последовательно найти сначала  $B_1$ , затем  $B_2$ ,  $B_3, \dots$  и, наконец,  $B_{n-1}$  и  $B_n$  для различных значений распределяемой суммы средств.

Для отыскания оптимального распределения средств прежде всего находим величину  $x_n^*(c)$  асигнований  $n$ -му предприятию, которая позволяет достичь полученного нами максимального значения  $B_n$  прироста продукции. По величине

оставшихся средств  $c - x_n^*(c)$  и уже известному нам значению  $B_{n-1}$  устанавливаем  $x_{n-1}^*(c)$  — величину ассигнований  $n-1$ -му предприятию и т.д. и, наконец, находим  $x_2^*(c)$  и  $x_1^*(c)$ .

**Пример.** Пусть имеются четыре предприятия, между которыми распределяется 100 тыс. ден. ед. Значения  $f_i(x)$  прироста выпуска продукции на предприятиях в зависимости от выделенной суммы  $x$  приведены в таблице. Составить план распределения средств, максимизирующий общий прирост выпуска продукции.

Таблица

Средства $c$ , тыс. ден. ед.	Предприятие		
	№ 1	№ 2	№ 3
	Прирост выпуска продукции на предприятиях, $f_i(x)$ , тыс. ден. ед.		
	$f_1(x)$	$f_2(x)$	$f_3(x)$
20	10	12	11
40	31	26	36
60	42	36	45
80	62	54	60
100	76	78	77

**Решение.** В данной задаче  $n=3$ ,  $c=100$  тыс. ден. ед.

Рассмотрим выделение средств первому предприятию:  $n=1$ . В соответствии с формулой в зависимости от начальной суммы  $c$  получаем с учетом таблицы значения  $B_1(c)$ .

$x_1^*(c)$	0	20	40	60	80	100
$B_1(c)$	0	10	31	42	62	76

Предположим теперь, что средства вкладываются в два предприятия.

Тогда в соответствии с формулой

$$B_2(c) = \max_{0 \leq x \leq c} (f_2(x) + B_1(c-x))$$

Очередная задача — найти значения функции для всех допустимых комбинаций  $c$  и  $x$ . Для упрощения расчетов значения  $x$  будем принимать кратными 20 тыс. ден. ед. и для большей наглядности записи оформлять в виде таблиц. Каждому шагу будет соответствовать своя таблица. Рассматриваемому шагу соответствует табл.

Для каждого значения  $(0, 20, 40, 60, 80, 100)$  начальной суммы  $c$  распределяемых средств в табл. предусмотрена отдельная строка, а для каждого возможного значения  $x$   $(0, 20, 40, 60, 80, 100)$  распределяемой суммы — столбец. Некоторые клетки таблицы останутся незаполненными, так как соответствуют недопустимым сочетаниям  $c$  и  $x$ . Такой, например, будет клетка, отвечающая строке  $c=40$  и столбцу

$x = 80$ , так как при наличии 40 тыс. ден. ед. естественно отпадает вариант, при котором одному из предприятий выделяется 80 тыс. ден. ед.

Для заполнения таблицы используем формулу  $B_2(c) = \max_{0 \leq x \leq c} (f_2(x) + B_1(c - x))$ .

Таблица

$c$	$x$	0	20	40	60	80	100	$B_2(c)$	$x_2^*(c)$
0	0+0							0	0
20	0+10	12+0						12	20
40	0+31	12+10	26+0					31	0
60	0+42	12+31	26+10	36+0				43	20
80	0+62	12+42	26+31	36+10	54+0			62	0
100	0+76	12+62	26+42	36+31	54+10	78+0		78	100

В каждую клетку таблицы будем вписывать значение суммы  $f_2(x) + B_1(c - x)$ . Первое слагаемое берем из условий задачи, второе — из табл. 10.9. Так, например, при распределении начальной суммы  $c = 80$  тыс. ден. ед. одним из вариантов может быть следующий: второму предприятию выделяется 60 тыс. ден. ед. ( $x = 60$ ), тогда первому —  $80 - 60 = 20$  тыс. ден. ед. При таком распределении первоначальной суммы на втором предприятии будет обеспечен прирост продукции на сумму в 36 тыс. ден. ед. (см. табл. 10.8), на первом — 10 тыс. ден. ед. (см. таблицу).

Общий прирост составит  $(36 + 10)$  тыс. ден. ед., что и записано в соответствующей клетке табл. 10.10. В двух последних столбцах таблицы проставлены максимальный по строке прирост продукции (в столбце  $B_2(c)$ ) и соответствующая ему оптимальная сумма средств, выделенная второму предприятию (в столбце  $x_2^*(c)$ ). Так, при начальной сумме  $c = 60$  тыс. ден. ед. максимальный прирост выпуска продукции составляет 43 тыс. ден. ед. ( $12 + 31$ ), и это достигается выделением второму предприятию 20, а первому —  $60 - 20 = 40$  тыс. ден. ед.

Расчет значений  $B_3(c)$  приведен в таблице. Здесь использована формула, получающаяся при  $\pi = 3$ :

$$B_3(c) = \max_{0 \leq x \leq c} (f_3(x) + B_2(c - x))$$

$c$	$x$	0	20	40	60	80	100	$B_3(c)$	$x_3^*(c)$
0	0+0							0	0
20	0+12	11+0						12	20
40	0+31	11+12	36+0					36	40
60	0+43	1+31	36+12	45+0				48	40
80	0+62	11+43	36+31	45+12	60+0			67	40
100	0+78	11+62	36+43	45+31	60+12	77+0		79	40

Полученные данные запишем в виде сводной таблицы, составленной на основе расчетных таблиц.

Таблица

$c$	0	20	40	60	80	100
$B_1(c)$	0	10	31	42	62	76
$x_1^*(c)$	0	20	40	60	80	100
$B_2(c)$	0	12	31	43	62	78
$x_2^*(c)$	0	20	0	20	0	100
$B_3(c)$	0	12	36	48	67	79
$x_3^*(c)$	0	0	40	40	40	40

Таблица содержит много ценной информации и позволяет единообразно решать целый ряд задач.

Решением задачи при  $n=3$ ,  $c=100$  тыс. ден. ед. является следующее: наибольший прирост выпуска продукции, который могут дать три предприятия при распределении между ними 100 тыс. ден. ед., составляет 79 тыс. ден. ед. ( $B_3(100)=79$ ).

При этом третьему предприятию должно быть выделено 40 тыс. ден.ед. ( $x_3^*(100)=40$ ), а остальным двум –  $100 - 40 = 60$  тыс. ден. ед. Из той же таблицы видно, что оптимальное распределение оставшихся 60 тыс. ден. ед. между двумя предприятиями обеспечит общий прирост продукции на них на сумму 43 тыс. ден. ед. ( $B_2(60)=43$ ) при условии, что второму предприятию будет выделено 20 тыс. ден. ед. ( $x_2^*(60)=20$ ), а первому –  $60 - 20 = 40$  тыс. ден. ед.

Итак, максимальный прирост выпуска продукции на трех предприятиях при распределении между ними 100 тыс. ден. ед. составляет 79 тыс. ден. ед. и будет получен, если первому предприятию выделить 40 тыс. ден. ед., второму выделить 20 тыс. ден. ед., а третьему также 40 тыс. ден. ед.

Убедиться в оптимальности следующего распределения 80 тыс. ден. ед. между тремя предприятиями:  $x_1^* = 40$ ,  $x_2^* = 0$ ,  $x_3^* = 40$ ; при этом  $B_3(80) = 67$ .

## 1.1. Одномерная задача оптимизации

Рассмотрим задачу поиска минимума одномерной функции  $\Phi(x)$ , определенной на интервале  $[a,b]$ :

$$\min_{x \in [a,b]} \Phi(x) = \Phi(x^*)$$

Как известно из курса математического анализа, внутренние точки локального и глобального минимума функции  $\Phi(x)$  являются стационарными точками критерия оптимальности  $\Phi(x)$  (см. рис. 1) или, что то же самое, решениями уравнения

$$\Phi'(x)=0. \quad (1)$$

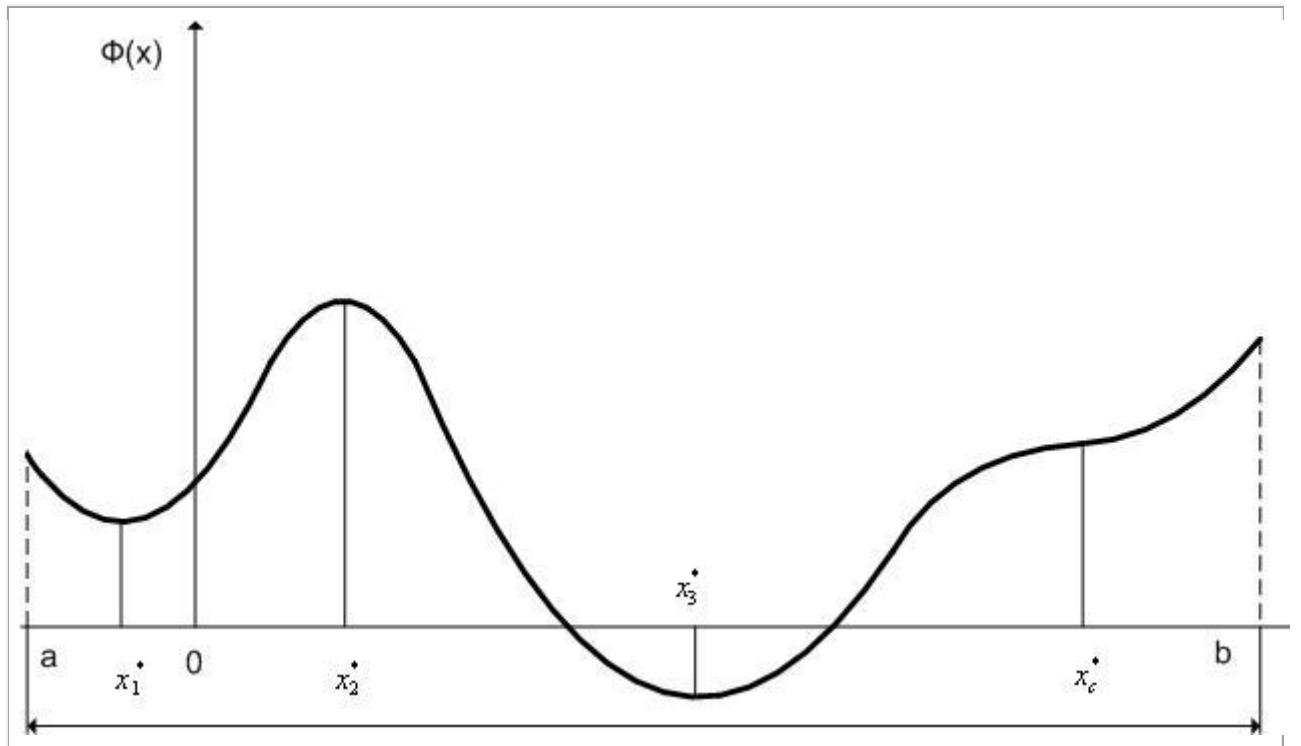


Рис. 1. Локальные минимумы ( $x_1^*$ ,  $x_3^*$ ), локальный максимум ( $x_2^*$ ) и точка перегиба ( $x_c^*$ ) функции  $\Phi(x)$ .

Но, решениями уравнения (1) являются не только точки минимума, но и точки максимума и точки перегиба функции  $\Phi(x)$  (см. рис. 1). Следовательно, уравнение (1) является только необходимым условием минимума, но не является достаточным условием.

Если существует вторая производная функции  $\Phi''(x)$ , то для отыскания достаточных условий минимума  $\Phi(x)$  можно привлечь эту производную. Из курса математического анализа известно, что если в точке  $x_0$  значение первой производной функции  $\Phi(x)$  равно нулю, а второй производной – положительно, то в этой точке функция  $\Phi(x)$  имеет минимум (локальный или глобальный).

Таким образом, имеем следующую теорему:

**Теорема 1.** Если функция  $\Phi(x)$  определена и дважды непрерывно дифференцируема на интервале  $[a,b]$ , то необходимыми и достаточными условиями минимума этой функции в точке  $x_0$  являются условия

$$\Phi'(x_0)=0, \Phi''(x_0)>0.$$

Приведем доказательство справедливости последнего условия. Для этого рассмотрим разложение функции  $\Phi(x)$  в ряд Тейлора в окрестности точки  $x_0$ :

$$\Phi(x_0 + \Delta x) - \Phi(x_0) = \Delta x \Phi'(x_0) + \frac{\Delta x^2}{2!} \Phi''(x_0) + \dots \quad (2)$$

Здесь  $\Delta x$  – некоторая достаточно малая величина.

Для того, что в точке  $x_0$  достигался минимум функции  $\Phi(x)$ , необходимо, чтобы разность  $\Phi(x_0 + \Delta x) - \Phi(x_0)$  была положительной. Поскольку  $\Phi'(x_0) = 0$ , то из (2) следует, что для выполнения этого условия необходимо, чтобы имело место неравенство  $\Phi''(x_0) > 0$ .

Точками, в которых функция  $\Phi(x)$  принимает наименьшее на интервале  $[a, b]$  значение, могут быть либо ее стационарные точки, лежащие внутри интервала  $[a, b]$ , либо ее точки недифференцируемости (критические точки критерия оптимальности), к которым следует отнести также концы интервала  $[a, b]$ .

Поэтому точку, в которой функция  $\Phi(x)$  принимает наименьшее на интервале  $[a, b]$  значение, нужно искать, сравнивая значения этой функции во всех стационарных и критических точках.

## 1.2. Многомерная задача безусловной оптимизации

Многие методы решения многомерной задачи нелинейного программирования основаны на сведении этой задачи к задаче безусловной оптимизации. Поэтому рассмотрим  $n$ -мерную задачу оптимизации без ограничений

$$\min_{\mathbf{X} \in R^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) \quad (1)$$

По аналогии с одномерной задачей, для того, чтобы точка  $\mathbf{X}_0$  являлась минимумом функции  $\Phi(\mathbf{X})$  необходимо выполнение условия стационарности функции  $\Phi(\mathbf{X})$  в точке  $\mathbf{X}_0$  или, что то же самое, необходимо, чтобы точка  $\mathbf{X}_0$  была стационарной точкой функции  $\Phi(\mathbf{X})$ :

$$\nabla \Phi(\mathbf{X}_0) = 0. \quad (2)$$

Положим, что функции  $\Phi(\mathbf{X})$  дважды непрерывно дифференцируема в окрестности точки  $\mathbf{X}_0$ . Для поиска достаточного условия достижения этой функцией в точке  $\mathbf{X}_0$  минимума, разложим  $\Phi(\mathbf{X})$  в окрестности точки  $\mathbf{X}_0$  в ряд Тейлора:

$$\Phi(\mathbf{X}_0 + \Delta \mathbf{X}) - \Phi(\mathbf{X}_0) = \Delta \mathbf{X}^T \nabla \Phi(\mathbf{X}_0) + \frac{1}{2} \Delta \mathbf{X}^T \mathbf{H}(\mathbf{X}_0) \Delta \mathbf{X} + \dots \quad (3)$$

Здесь  $n$ -мерный вектор-столбец достаточно малых величин  $\Delta \mathbf{X} = \{\Delta x_i, i \in [1, n]\}$ ,  $\mathbf{H}(\mathbf{X})$  –  $(n \times n)$ -матрица Гессе.

По аналогии с одномерной задачей, для того, что в точке  $\mathbf{X}_0$  достигался минимум функции  $\Phi(\mathbf{X})$ , необходимо, чтобы разность  $\Phi(\mathbf{X}_0 + \Delta \mathbf{X}) - \Phi(\mathbf{X}_0)$  была положительной. Поскольку  $\nabla \Phi(\mathbf{X}_0) = 0$ , то из (3) следует, что для выполнения этого условия необходимо, чтобы матрица Гессе  $\mathbf{H}(\mathbf{X})$  была положительно определена в точке  $\mathbf{X}_0$ .

Таким образом, справедлива

Теорема 1. Если функция  $\Phi(\mathbf{X})$  дважды непрерывно дифференцируема в окрестности точки  $\mathbf{X}_0 \in R^n$ , то необходимыми и достаточными условиями минимума этой функции в точке  $\mathbf{X}_0$  являются условия:

$$\nabla \Phi(\mathbf{X}_0) = 0; \quad (4)$$

$\mathbf{H}(\mathbf{X}_0)$  – положительно определена •

Таким образом, теорема 1 определяет необходимые и достаточные условия минимума в многомерной задаче безусловной оптимизации.

Заметим, что условие  $\nabla \Phi(\mathbf{X}_0) = 0$  является только необходимым условием минимума в многомерной задаче безусловной оптимизации.

По аналогии с одномерной задачей точками, в которых функция  $\Phi(\mathbf{X})$  достигает своего наименьшего значения, могут быть либо ее стационарные точки функции, либо критические точки функции (точки недифференцируемости).

Поэтому так же, как в одномерной задаче, точку, в которой функция  $\Phi(\mathbf{X})$  принимает наименьшее значение нужно искать, сравнивая значения этой функции во всех стационарных и критических точках.

### 1.3. Задача выпуклого программирования

Рассмотрим  $n$ -мерную задачу выпуклого программирования

$$\min_{\mathbf{X} \in D} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*,$$

$\Phi(\mathbf{X})$  – выпуклая функция,  $D$  – выпуклое не пустое ограниченное и замкнутое множество допустимых значений вектора варьируемых переменных. Напомним, что по определению выпуклая функция является непрерывной.

Во внутренних точках множества допустимых значений  $D$  функция  $\Phi(\mathbf{X})$  достигает минимального значения в точках, которые являются ее либо стационарными точками функции, либо критическими точками функции. Однако функция может достигать своего наименьшего значения и в граничных точках области определения  $D$ .

Важные свойства задачи выпуклого программирования определяют две следующие теоремы.

Теорема 1. Если внутренняя точка  $\mathbf{X}^*$  множества  $D$  является точкой локального минимума в задаче выпуклого программирования, то в этой точке функция  $\Phi(\mathbf{X})$  достигает глобального минимума.

Доказательство. Положим, что в точке  $\mathbf{X}^*$  функция  $\Phi(\mathbf{X})$  не достигает наименьшего во множестве  $D$  значения. Тогда существует точка  $\mathbf{Y} \in D$ , для которой  $\Phi(\mathbf{Y}) < \Phi(\mathbf{X}^*)$ .

Рассмотрим сечение  $\varphi(\lambda) = \Phi(\lambda \mathbf{Y} + (1-\lambda)\mathbf{X}^*)$ ,  $\lambda \in [0,1]$ . Функция  $\varphi(\lambda)$  достигает в точке  $\lambda=0$  наибольшее значение. Действительно, поскольку

$$\varphi(\lambda) = \Phi(\lambda \mathbf{Y} + (1-\lambda)\mathbf{X}^*) \leq \lambda \Phi(\mathbf{Y}) + (1-\lambda)\Phi(\mathbf{X}^*) < \lambda \Phi(\mathbf{X}^*) + (1-\lambda)\Phi(\mathbf{X}^*) = \Phi(\mathbf{X}^*) = \varphi(0).$$

Это значит, что существует окрестность  $\delta D \in D$  точки  $\mathbf{X}^*$  и некоторое  $\tilde{\lambda} \in [0,1]$  такие, что  $\mathbf{X} = \tilde{\lambda} \mathbf{Y} + (1-\tilde{\lambda})\mathbf{X}^* \in \delta D$ . Но тогда  $\Phi(\mathbf{X}) = \varphi(\tilde{\lambda}) < \varphi(0) = \Phi(\mathbf{X}^*)$ , что противоречит условию теоремы •

Из теоремы следует, что во всех точках локального минимума выпуклая функция имеет одинаковые значения.

Пример 1 Рассмотрим не строго выпуклую квадратичную функцию  $\Phi(\mathbf{X}) = \Phi(x_1, x_2) = (x_1 + x_2)^2$ , определенную в области  $D = \{\mathbf{X} | 2 \geq x_1 \geq -2, 2 \geq x_2 \geq -2\}$  (см. рис.

1). Все локальные минимумы этой функции равны нулю и расположены на прямой  $x_1 + x_2 = 0$ .

MATLAB-программа:

```
x=-2:0.06:2;  
y=x;  
[X,Y]=meshgrid(x);  
Z=(X+Y).^2;  
V=[0.025,0.5,1,2,4,8];  
[C,h]=contour(X,Y,Z,V);  
clabel(C,h);
```

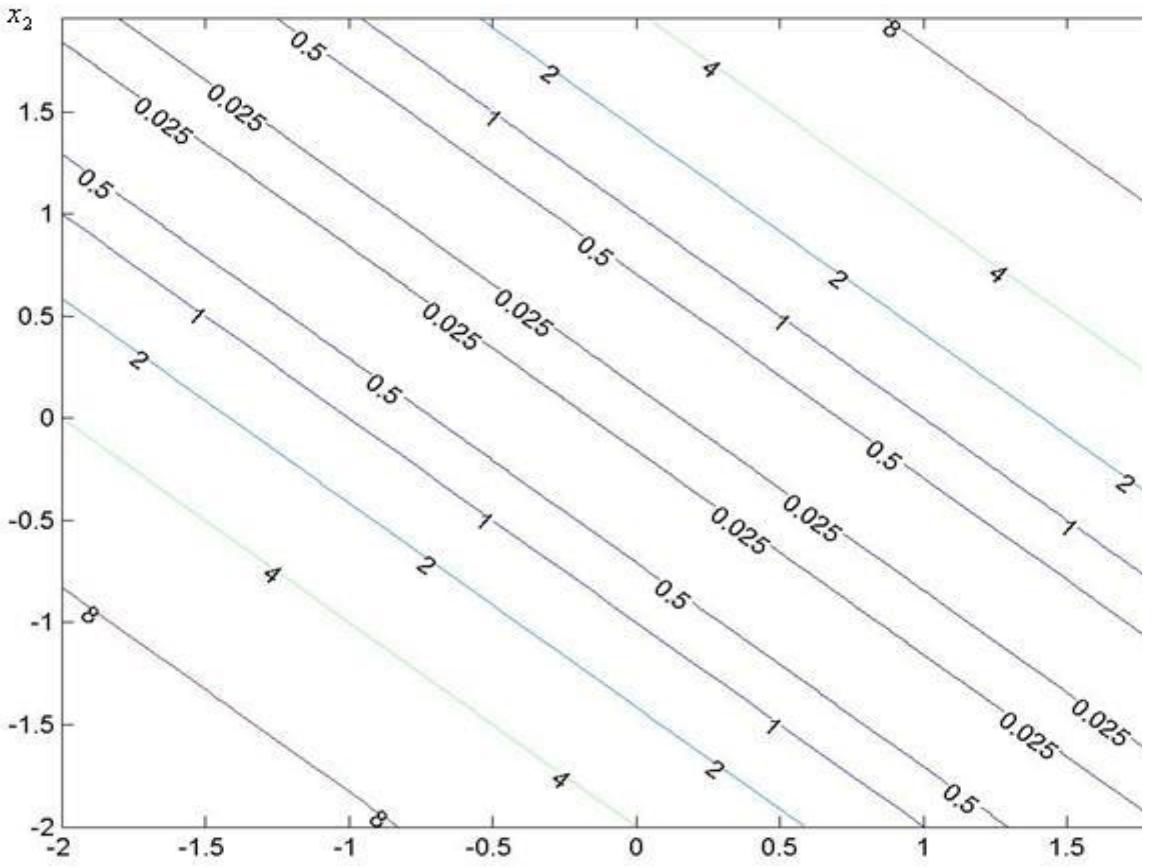


Рис. 1. К прим. 1

Теорема 2. Функция  $\Phi(\mathbf{X})$ , строго выпуклая функция на выпуклом множестве, имеет в этом множестве не более одной точки минимума (глобального) •

Условие существования решения задачи выпуклого программирования дает следующая теорема.

Теорема 3. Пусть функция  $\Phi(\mathbf{X})$  выпукла на выпуклом множестве  $D \subset R^n$  и дифференцируема в точке  $\mathbf{X}^* \in D$ . Тогда для того чтобы эта точка была точкой минимума функции  $\Phi(\mathbf{X})$ , необходимо и достаточно, чтобы для любой точки  $\mathbf{X} \in D$  выполнялось неравенство

$$(\nabla \Phi(\mathbf{X}^*), (\mathbf{X} - \mathbf{X}^*)) \geq 0. \quad (1)$$

Необходимость. Рассмотрим сечение  $\psi(\lambda) = \Phi(\lambda \mathbf{X} + (1-\lambda)\mathbf{X}^*)$  функции  $\Phi(\mathbf{X})$ . Функция  $\psi(\lambda)$  определена на отрезке  $[0,1]$ , имеет в точке  $\lambda=0$  локальный минимум и дифференцируема в этой точке. Следовательно  $\psi'(0) \geq 0$  (равенство нулю имеет место в том случае, когда точка  $\mathbf{X}^*$  является внутренней точкой множества  $D$ ). По правилу дифференцирования сложной функции

$$\psi'(0) = \Phi'_\lambda \Phi(\lambda \mathbf{X} + (1-\lambda)\mathbf{X}^*)|_{\lambda=0} = (\nabla \Phi(\mathbf{X}^*), (\mathbf{X} - \mathbf{X}^*)) \geq 0.$$

Достаточность. Пусть в точке  $\mathbf{X}^* \in D$  выполнено неравенство (1). Рассмотрим сечение  $\psi(\lambda) = \Phi(\lambda \mathbf{X} + (1-\lambda)\mathbf{X}^*)$  функции  $\Phi(\mathbf{X})$ , где  $\mathbf{X}$  – произвольная точка из множества  $D$ .

Поскольку  $\Phi(\mathbf{X})$  выпукла во множестве  $D$ , то функция  $\Phi(\lambda)$  также выпукла на отрезке  $[0,1]$ . Кроме того, из неравенства (1) следует, что  $\Phi(0) \geq 0$ . Это означает, что  $\Phi(\lambda)$  - неубывающая отрезке  $[0,1]$  функция, т.е.  $\Phi(0) \leq \Phi(1)$ . Последнее неравенство означает, что  $\Phi(\mathbf{X}^*) \leq \Phi(\mathbf{X})$  и в точке  $\mathbf{X}^*$  функция  $\Phi(\mathbf{X})$  принимает наименьшее в области  $D$  значение.

Точка  $\mathbf{X}_1^*$  на рис. 2 является точкой локального минимума, поскольку не существует такой точки  $\mathbf{X} \in D$ , что скалярное произведение  $(\nabla \Phi(\mathbf{X}_1^*), (\mathbf{X} - \mathbf{X}_1^*))$  отрицательно. Точка  $\mathbf{X}_4$ , например, не является точкой локального минимума, так как существуют такие точки  $\mathbf{X} \in D$ , что скалярное произведение  $(\nabla \Phi(\mathbf{X}_4^*), (\mathbf{X} - \mathbf{X}_4^*))$  отрицательно.

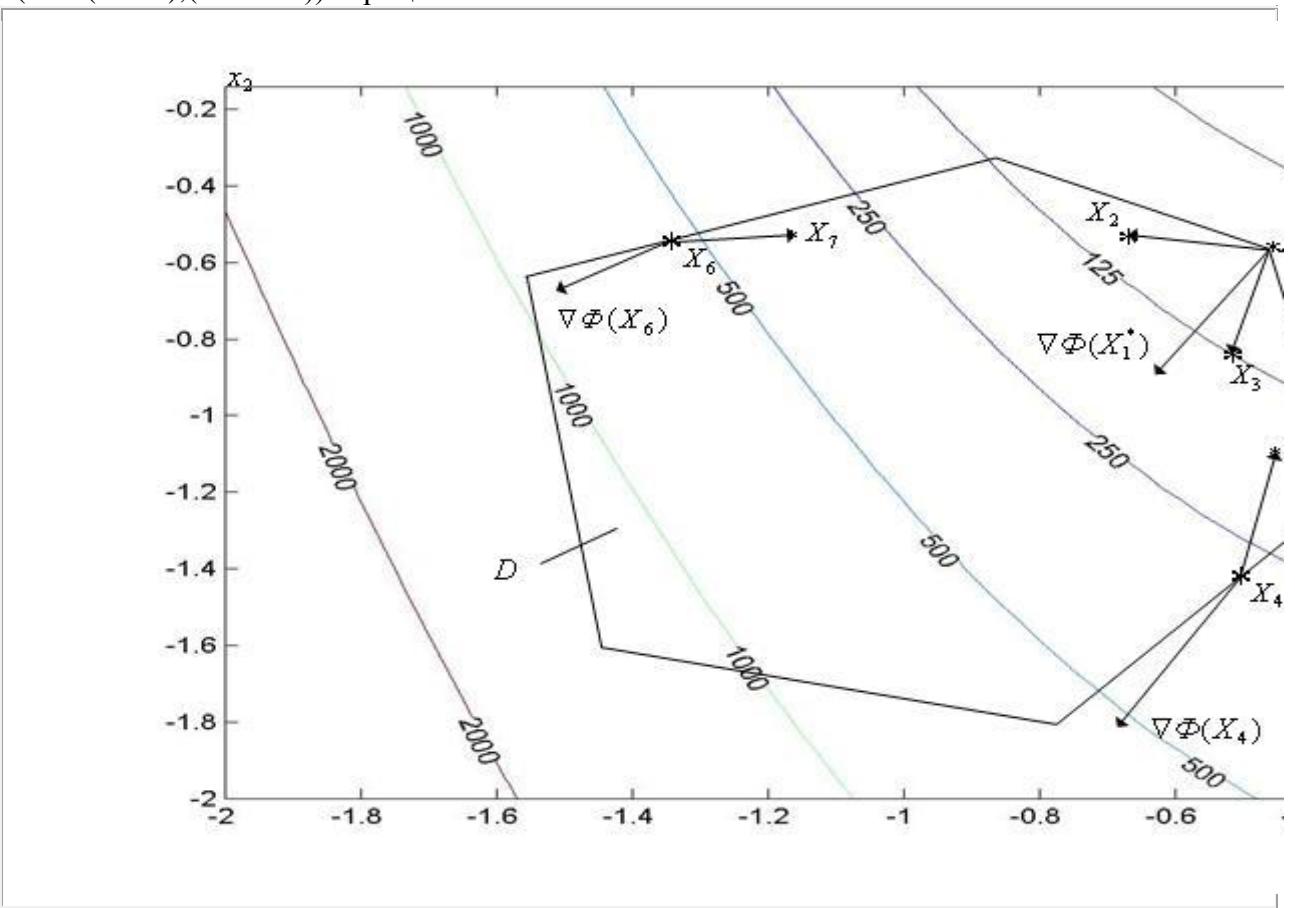


Рис. 2. К теореме 3.

Теорему 3 иллюстрирует рис. 2, линии уровня на котором получены с помощью следующей MATLAB-программы:

$x=-2:0.06:-0.1;$

$y=x;$

$[X,Y]=meshgrid(x);$

$Z=100.* (Y-X.^2).^2+(1-X).^2;$

$V=[2,8,32,125,250,500,1000,2000];$

$contour(X,Y,Z,V);$

$[C,h]=contour(X,Y,Z,V);$

$clabel(C,h);$

Заметим, что если точка  $\mathbf{X}^* \in D$  является внутренней точкой множества  $D$ , то условие (1) эквивалентно условию  $\nabla \Phi(\mathbf{X}^*) = 0$ . Таким образом, условие (1) можно рассматривать как обобщение необходимого условия минимума в многомерной задаче безусловной оптимизации.

#### 1.4. Задача нелинейного программирования с ограничениями типа равенств

Рассмотрим  $n$ -мерную задачу нелинейного программирования

$$\min_{\mathbf{X} \in D} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где

$$D = \{\mathbf{X} | h(\mathbf{X}) = 0\} = \{\mathbf{X} | h_j(\mathbf{X}) = 0, j \in [1, l]\} \subset \mathbb{R}^n. \quad (2)$$

-не пустое, ограниченное замкнутое множество.

Нам понадобятся далее понятия множителей Лагранжа и функции Лагранжа. Функция Лагранжа для задачи (1) с ограничениями (2) определяется формулой

$$L(\mathbf{X}, \lambda) = \Phi(\mathbf{X}) + \sum_{j=1}^l \lambda_j h_j(\mathbf{X}) = \Phi(\mathbf{X}) + \lambda^T h(\mathbf{X}), \quad (3)$$

где  $\lambda = (\lambda_j, j \in [1, l])$  -  $(l \times 1)$ -вектор множителей Лагранжа.

Нам понадобится также понятие условия регулярности ограничивающих функций. Если точка  $\mathbf{X}^* \in D$ , то условие линейной независимости векторов  $\nabla h_j(\mathbf{X}^*), j \in [1, l]$  называется условием регулярности задачи (1), (2) в точке  $\mathbf{X}^*$ . Данное условие означает, в частности, что количество ограничивающих функций, проходящих через точку  $\mathbf{X}^*$ , не может быть больше размерности вектора варьируемых параметров, т.е. должно быть выполнено неравенство  $l \leq n$ . Например, на рис. 1 в ситуации (а) количество ограничивающих функций, проходящих через точку  $\mathbf{X}^*$ , превышает размерность вектора варьируемых параметров, в ситуации (б) в точке  $\mathbf{X}^*$  градиенты  $\nabla h_1(\mathbf{X}^*), \nabla h_2(\mathbf{X}^*)$  ограничивающих функций коллиниарны.

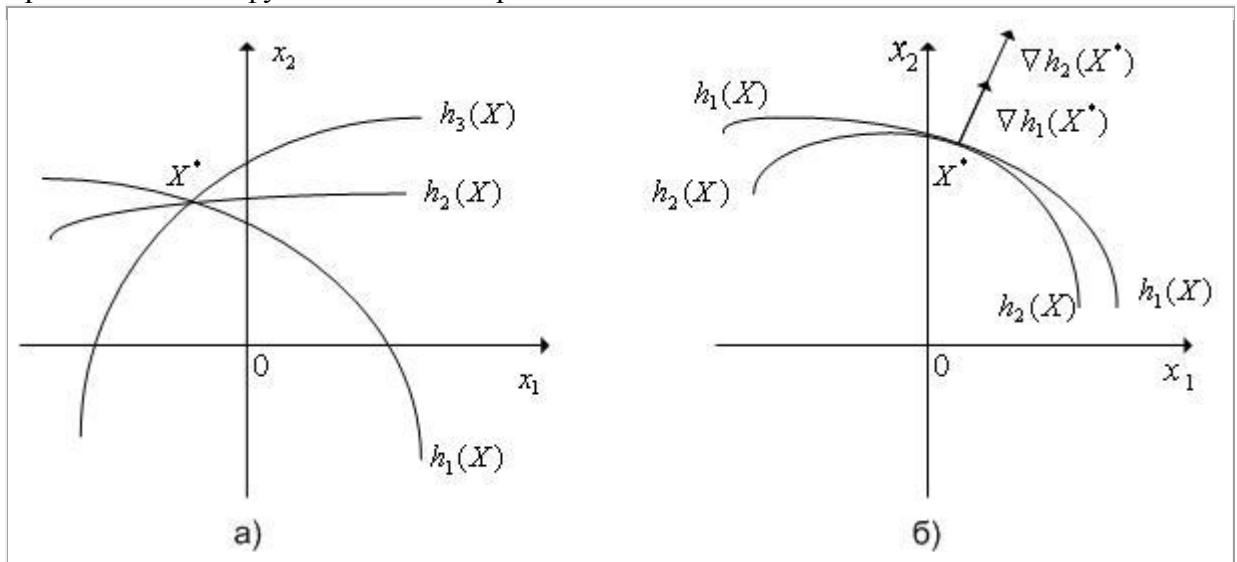


Рис. 1. Ситуации, в которых в двумерном случае ( $n=2$ ) не выполняется условие регулярности системы функций  $h(\mathbf{X})$  в точке  $\mathbf{X}^*$ .

Исключительно важное место в теории и практике решения задач нелинейного программирования с ограничениями типа равенств занимает следующая теорема (правило Лагранжа для задачи оптимизации с ограничениями типа равенств).

Теорема 1. Пусть функция  $\Phi(\mathbf{X})$  и функции  $\nabla h_j(\mathbf{X}), j \in [1, l]$  имеют непрерывные частные производные в некоторой окрестности точки  $\mathbf{X}^*$  и пусть эта точка является точкой локального минимума функции  $\Phi(\mathbf{X})$  при условии  $h(\mathbf{X}^*)=0$ . Пусть, кроме того, выполняется условие регулярности системы функций  $h(\mathbf{X})$  в точке  $\mathbf{X}^*$ . Тогда существуют такие множители Лагранжа  $\lambda_j, j \in [1, l]$ , не все из которых равны нулю одновременно, что для функции Лагранжа  $L(\mathbf{X}, \lambda)$  точка  $\mathbf{X}^*$  является стационарной точкой функции, т.е.

$$\nabla_{\mathbf{X}} L(\mathbf{X}^*, \lambda) = \nabla \Phi(\mathbf{X}^*) + \sum_{j=1}^l \lambda_j \nabla h_j(\mathbf{X}^*) = 0. \quad (4)$$

Доказательство теоремы приведем для одного частного случая. Пусть  $n=3$ , т.е. минимизируемая функция  $\Phi(\mathbf{X})=\Phi(x_1, x_2, x_3)$ , и пусть заданы два ограничения типа равенств

$$h_1(\mathbf{X})=h_1(x_1, x_2, x_3)=0, h_2(\mathbf{X})=h_2(x_1, x_2, x_3)=0. \quad (5)$$

Ограничения (5) определяют область допустимых значений  $D$ , которая представляет собой некоторую кривую в пространстве  $R^3$ , являющуюся результатом пересечения поверхностей  $h_1(\mathbf{X})=0, h_2(\mathbf{X})=0$ . Допустим, что функция  $\Phi(\mathbf{X})$  имеет точку локального минимума  $\mathbf{X}^*=(x_1^*, x_2^*, x_3^*)$  в области  $D$ . Допустим также, что выполнены условия теоремы 1, т.е. функции  $\Phi(\mathbf{X}), h_1(\mathbf{X}), h_2(\mathbf{X})$  имеют непрерывные частные производные в некоторой окрестности точки  $\mathbf{X}^*$  и градиенты функций  $h_1(\mathbf{X}), h_2(\mathbf{X})$  в этой точке линейно независимы. Положим, кроме того, что из равенств (5) переменные  $x_2, x_3$  можно выразить через переменную  $x_1$  в виде

$$x_2=H_1(x_1), x_3=H_2(x_1). \quad (6)$$

Подставив выражения (6) в выражение для функции  $\Phi(\mathbf{X})$ , преобразуем исходную задачу к следующей задаче оптимизации без ограничений, которая содержит только одну переменную  $x_1$ :

$$\min_{\mathbf{X} \in R^1} \Phi(x_1, H_1(x_1), H_2(x_1)). \quad (7)$$

Поскольку функция  $\Phi(\mathbf{X})$  имеет точку минимума  $\mathbf{X}^* \in D$ , производная по  $x_1$  функции  $\Phi(x_1, H_1(x_1), H_2(x_1))$  в точке  $\mathbf{X}^*$  равна нулю:

$$\frac{d\Phi}{dx_1} = \frac{\partial \Phi}{\partial x_1} + \frac{\partial \Phi}{\partial x_2} \frac{dH_1}{dx_1} + \frac{\partial \Phi}{\partial x_3} \frac{dH_2}{dx_1} = 0. \quad (8)$$

Дифференцируя по  $x_1$  выражения (5), получим

$$\frac{dh_i}{dx_1} = \frac{\partial h_i}{\partial x_1} + \frac{\partial h_i}{\partial x_2} \frac{dH_1}{dx_1} + \frac{\partial h_i}{\partial x_3} \frac{dH_2}{dx_1} = 0, \quad i=1,2. \quad (9)$$

Запишем уравнения (8), (9) в виде матричного уравнения

$$\mathbf{A}^* \begin{pmatrix} 1 \\ \frac{dH_1}{dx_1} \\ \frac{dH_2}{dx_1} \end{pmatrix} = 0, \quad \mathbf{A} = \begin{pmatrix} \nabla \Phi(\mathbf{X}^*) \\ \nabla h_1(\mathbf{X}^*) \\ \nabla h_2(\mathbf{X}^*) \end{pmatrix} \quad (10)$$

Поскольку вектор  $\begin{pmatrix} 1 & \frac{dH_1}{dx_1} & \frac{dH_2}{dx_1} \end{pmatrix}^T$  не нулевой, то равенство (10) возможно лишь в том случае, когда  $\det(\mathbf{A})=0$ . Но это возможно лишь в том случае, когда вектора-строки матрицы  $\mathbf{A}$  линейно зависимы. Значит, существуют такие скаляры  $a, b, c$ , не все равные нулю, что

$$a\nabla \Phi(\mathbf{X}^*) + b\nabla h_1(\mathbf{X}^*) + c\nabla h_2(\mathbf{X}^*) = 0. \quad (11)$$

В выражении (11) скаляр  $a$  не может быть равен нулю, поскольку противное означало бы линейную зависимость векторов  $\nabla h_1(\mathbf{X}^*)$ ,  $\nabla h_2(\mathbf{X}^*)$  что противоречит условию теоремы. Поэтому после деления на  $a$  из (11) получим

$$\nabla \Phi(\mathbf{X}^*) + \lambda_1 \nabla h_1(\mathbf{X}^*) + \lambda_2 \nabla h_2(\mathbf{X}^*) = 0.$$

Таким образом, для рассматриваемого частного случая справедливость теоремы доказана

• Отметим, что теорема 1 не требует знакопределенности (т.е. положительности или отрицательности) множителей Лагранжа  $\lambda_j, j \in [1, l]$ . Теорема требуется лишь того, чтобы не все из этих множителей равнялись нулю одновременно.

Пример 1

Рассмотрим в качестве минимизируемой функции функцию Розенброка  $\Phi(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ , ( $n=2$ ). Положим, что имеется только одно ограничение типа равенств, которое задается с помощью функции  $h_1(\mathbf{X}) = x_1 + x_2 + 0.2 = 0$ . Легко видеть, что градиенты функций  $\Phi(\mathbf{X})$ ,  $h_1(\mathbf{X})$  равны, соответственно

$$\nabla \Phi(\mathbf{X}) = (-400(x_2 - x_1^2)x_1 - 2(1 - x_1), 200((x_2 - x_1^2)))^T,$$

$$\nabla h_1(\mathbf{X}) = (1, 1)^T.$$

Задачу иллюстрирует рис. 2, линии уровня функции Розенброка на котором получены с помощью следующей MATLAB-программы:

```
x=-2:0.06:0;
y=x;
[X,Y]=meshgrid(x);
Z=100.* (Y-X.^2).^2+(1-X).^2;
V=[2,8,32,125,250,500,1000,2000];
contour(X,Y,Z,V);
[C,h]=contour(X,Y,Z,V);
clabel(C,h);
```

В точках  $\mathbf{X}_2, \mathbf{X}_3$  векторы градиента функций  $\Phi(\mathbf{X}), h_1(\mathbf{X})$  не коллинеарны. Поэтому для этих точек не существует не равный нулю множитель Лагранжа  $\lambda_1$ , при котором функция Лагранжа равна нулю:  $\nabla\Phi(\mathbf{X}) + \lambda_1\nabla h_1(\mathbf{X}) = 0$ . И поэтому точки  $\mathbf{X}_2, \mathbf{X}_3$  не могут быть точками локального минимума для рассматриваемой задачи. Наоборот, в точке  $\mathbf{X}^*$  векторы градиента функций  $\Phi(\mathbf{X}), h_1(\mathbf{X})$  коллинеарны и поэтому существует не равный нулю множитель  $\lambda_1$ , при котором справедливо равенство  $\nabla\Phi(\mathbf{X}^*) + \lambda_1\nabla h_1(\mathbf{X}^*) = 0$ . Отметим, что, например в точке  $(-1, -1)^T$ , градиент функции Розенброка равен  $(-402, -400)^T$ .

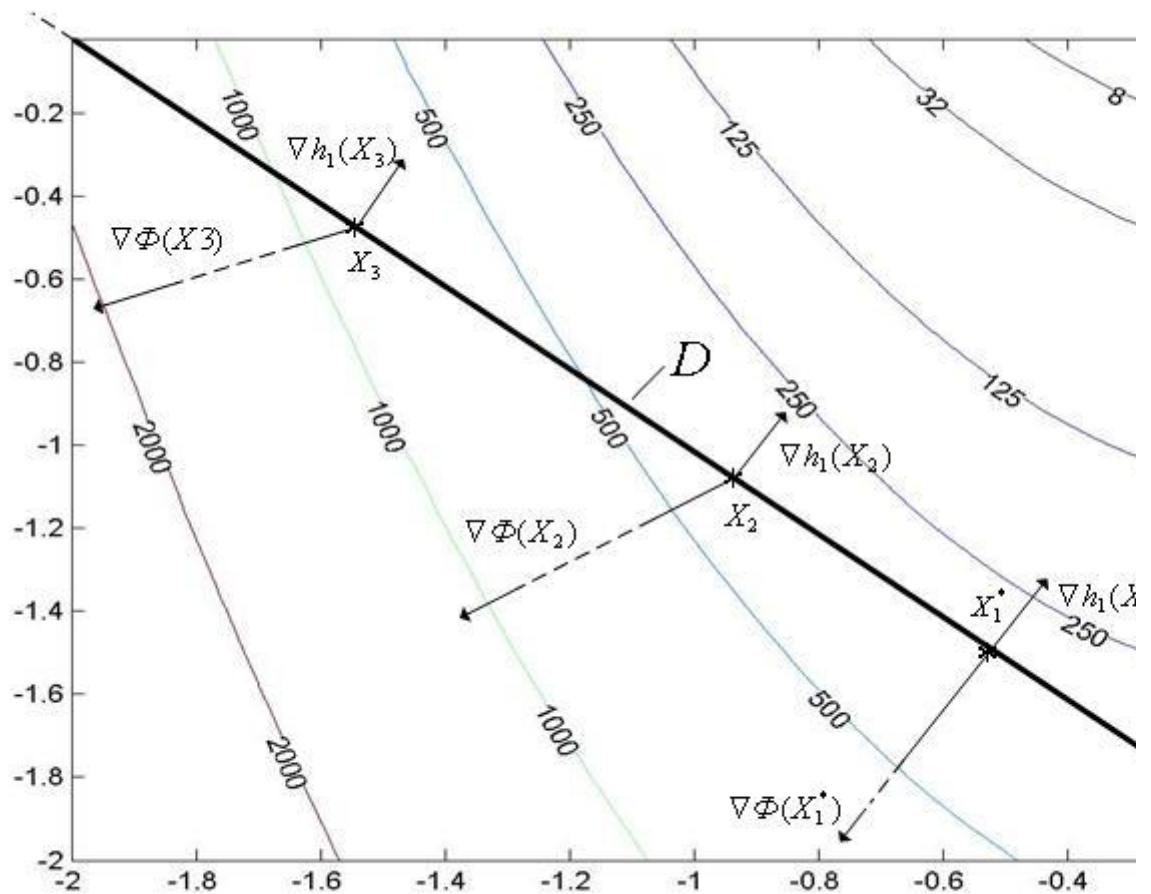


Рис. 2. К прим. 1.

Теорема 1 означает, что в ее условиях вместо задачи условной оптимизации (1), (2) можно решать задачу безусловной оптимизации

$$\min_{\mathbf{X} \in R^n} \left( \Phi(\mathbf{X}) + \sum_{j=1}^l \lambda_j h_j(\mathbf{X}) \right)$$

Необходимым условием существования локального минимума этой задачи в некоторой

точке  $\mathbf{X}^* \in R^n$  является условие  $\nabla_{\mathbf{X}} \left( \Phi(\mathbf{X}) + \sum_{j=1}^l \lambda_j h_j(\mathbf{X}) \right) = 0$  (см. Теорему 2.1).

Широко известна другая форма теоремы 1, которую мы сформулируем в виде следствия этой теоремы.

Следствие. В условиях теоремы 1 существуют такие множители Лагранжа  $\lambda_j, j \in [1, l]$ , не все из которых равные нулю одновременно, что имеют место следующие равенства:

$$\nabla_x L(\mathbf{X}^*, \lambda) = 0, \quad (12)$$

$$\nabla_\lambda L(\mathbf{X}^*, \lambda) = h(\mathbf{X}^*) = 0. \quad (13)$$

Здесь равенство (12) повторяет равенство (4), а справедливость равенства (13) следует из того факта, что по условиям теоремы точка  $\mathbf{X}^*$  удовлетворяет всем ограничениям, т.е.  $h_j(\mathbf{X}^*) = 0, j \in [1, l]$ .

Заметим, что из (13) следует справедливость еще одного полезного равенства

$$\lambda^T \nabla_\lambda L(\mathbf{X}^*, \lambda) = \sum_{j=1}^l \lambda_j h_j(\mathbf{X}^*) = \lambda^T h(\mathbf{X}^*) = 0.$$

## 1.5. Теорема Куна-Таккера для задачи нелинейного программирования с ограничениями типа неравенств

Рассмотрим задачу нелинейного программирования

$$\min_{\mathbf{X} \in D} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где  $\Phi(\mathbf{X})$  – произвольная функция,

$$D = \{\mathbf{X} | g(\mathbf{X}) \geq 0\} = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0, i \in [1, m]\} \subset \mathbb{R}^n.$$

не пустое, ограниченное замкнутое множество.

Нам понадобятся далее понятия множителей Лагранжа и функции Лагранжа для задачи нелинейного программирования с ограничениями типа неравенств. Функция Лагранжа для задачи (1) с ограничениями (2) определяется формулой

$$L(\mathbf{X}, \lambda) = \Phi(\mathbf{X}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{X}) = \Phi(\mathbf{X}) + \lambda^T g(\mathbf{X}), \quad (2)$$

где  $\lambda = (\lambda_i, i \in [1, m])$  –  $(m \times 1)$ -вектор множителей Лагранжа.

Нам понадобятся также понятия активных и неактивных ограничений. В точке локального минимума задачи (1), (2) каждое из ограничений  $g_i(\mathbf{X}^*) \geq 0, i \in [1, m]$  выполняется либо в виде равенства  $g_i(\mathbf{X}^*) = 0$ , либо в виде неравенства  $g_i(\mathbf{X}^*) > 0$ . Ограничения первого вида называются активными ограничениями. Остальные ограничения называются неактивными ограничениями.

Кроме того, нам понадобится также понятие условия регулярности для задачи нелинейного программирования с ограничениями типа неравенств. Если точка  $\mathbf{X}^* \in D$  и

ограничения  $g_{ij}(\mathbf{X}^*) \geq 0, i, j \in [1, s], s \leq m$  активны, то условие линейной независимости векторов называется условием регулярности ограничивающих функций

$\nabla g_{ij}(\mathbf{X}^*) \geq 0, i, j \in [1, s]$  в точке  $\mathbf{X}^*$ . Это условие означает, что, например, при  $n=2$

количество ограничивающих функций, проходящих через точку  $\mathbf{X}^*$ , не должно превышать 2 и в точке  $\mathbf{X}^*$  векторы  $\nabla g_1(\mathbf{X}), \nabla g_2(\mathbf{X})$  не должны быть коллиниарны.

Например, на рис. 1 в ситуации (а) количество ограничивающих функций, проходящих через точку  $\mathbf{X}^*$ , превышает размерность вектора варьируемых параметров, в ситуации (б) в точке  $\mathbf{X}^*$  градиенты  $\nabla g_1(\mathbf{X}), \nabla g_2(\mathbf{X})$  ограничивающих функций коллиниарны.

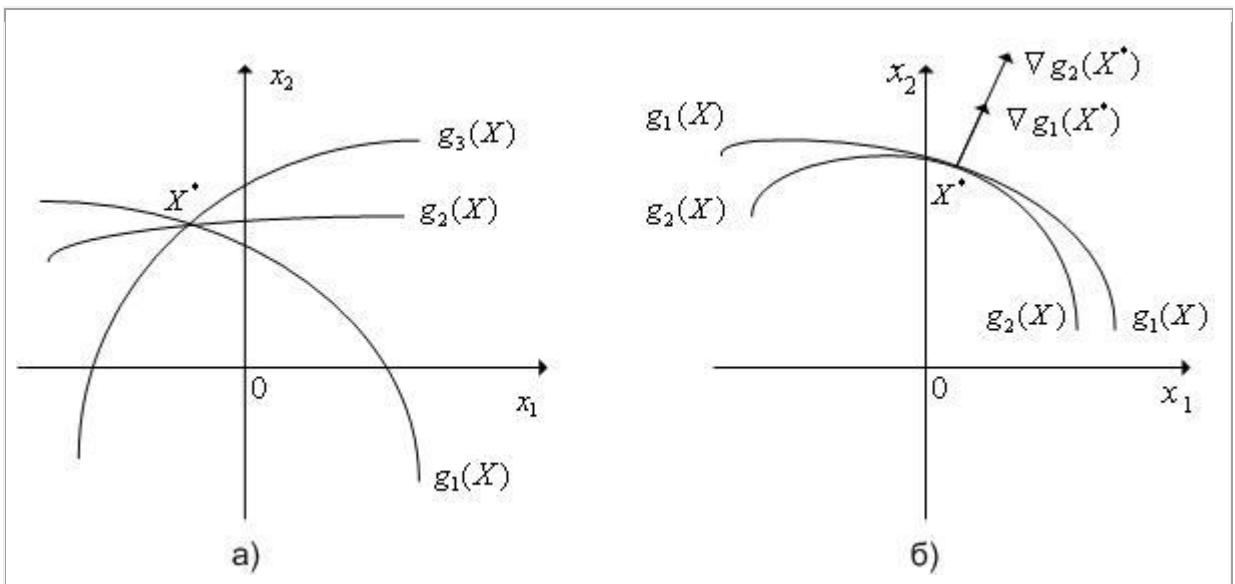


Рис. 1. Ситуации, в которых не выполняется условие регулярности двумерной задачи. Исключительно большое значение в теории и практике решения задач нелинейного программирования имеет следующая теорема (теорема Куна-Таккера для задачи условной оптимизации с ограничениями типа неравенств).

Теорема 1 (Куна-Таккера). Пусть функция  $\Phi(\mathbf{X})$  и функции  $g_i(\mathbf{X}) \geq 0, i \in [1, m]$  имеют непрерывные частные производные в некоторой окрестности точки  $\mathbf{X}^*$  и пусть эта точка является точкой локального минимума функции  $\Phi(\mathbf{X})$  при ограничениях  $g_i(\mathbf{X}^*) \geq 0$ , удовлетворяющих в точке  $\mathbf{X}^*$  условию регулярности ограничивающих функций. Тогда существуют такие неотрицательные множители Лагранжа  $\lambda_i, i \in [1, m]$ , что для функции Лагранжа  $L(\mathbf{X}, \lambda)$  точка  $\mathbf{X}^*$  является стационарной точкой функции, т.е.

$$\nabla_{\mathbf{X}} L(\mathbf{X}^*, \lambda) = \nabla \Phi(\mathbf{X}^*) + \sum_{i=1}^m \lambda_i \nabla g_i(\mathbf{X}^*) = 0 \bullet \quad (3)$$

Заметим, что в отличие от правила множителей Лагранжа, теорема 1 требует знакопределенности множителей Лагранжа  $\lambda_i, i \in [1, m]$ . Отметим также, что теорема не запрещает того, чтобы все множители Лагранжа  $\lambda_i, i \in [1, m]$  были равны нулю.

Поясним смысл теоремы на примере.

Пример 1

Рассмотрим двумерную ( $n=2$ ) задачу нелинейного программирования (1), (2), в которой область допустимых значений  $D$  задается тремя ограничивающими функциями, т.е.  $D = \{\mathbf{X} | g(\mathbf{X}) \geq 0\} = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0, i \in [1, 3]\} \subset \mathbb{R}^2$ . Положим, что множество  $D$  имеет вид, представленный на рис. 2.

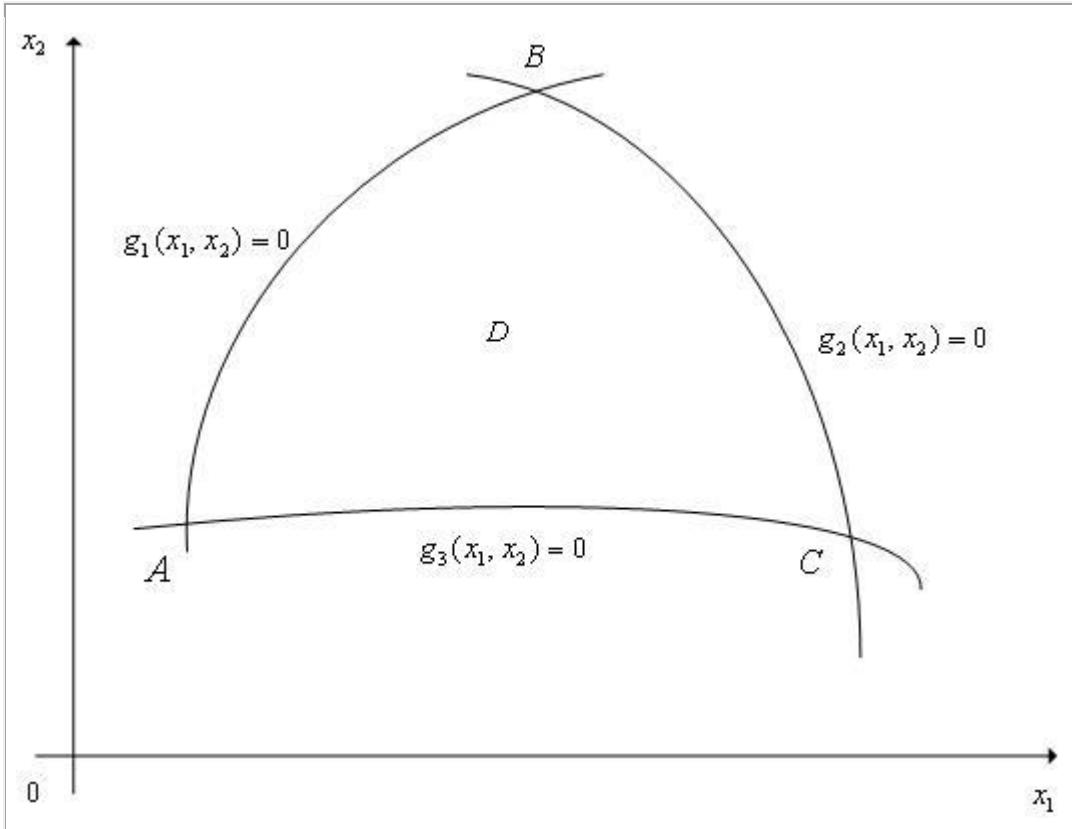


Рис. 2. К прим. 1.

Для всех граничных точек области  $D$ , очевидно, выполняются условия регулярности ограничивающих функций.

Если точка  $\mathbf{X}^*$  находится внутри множества  $D$  (т.е. является стационарной точкой функции  $\Phi(\mathbf{X})$ ), то теорема будет справедлива, если положить все множители Лагранжа  $\lambda_i, i \in [1, m]$  равными нулю.

Пусть теперь точка  $\mathbf{X}^*$  находится на одной из дуг, например, на дуге АВ, т.е. пусть ограничение  $g_1(\mathbf{X}) \geq 0$  является активным ограничением, а остальные ограничения – неактивными ограничениями. Тогда в этой точке  $g_1(\mathbf{X}^*) = 0$  и справедливость теоремы вытекает из правила множителей Лагранжа для задачи с ограничениями типа равенств, если положить  $\lambda_2 = \lambda_3 = 0$ .

Пусть, наконец, точка  $\mathbf{X}^*$  находится в одной из угловых точек множества  $D$ , например, в точке  $B$ , т.е. пусть ограничения  $g_1(\mathbf{X}) \geq 0$ ,  $g_2(\mathbf{X}) \geq 0$  являются активными ограничениями, а ограничение  $g_3(\mathbf{X}) \geq 0$  – неактивным ограничением. Тогда можно положить  $\lambda_3 = 0$  и справедливость теоремы вытекает из правила множителей Лагранжа для задачи с ограничениями типа равенств •

Теорема 1 означает, что в ее условиях вместо задачи условной оптимизации (1), (2) можно решать задачу безусловной оптимизации

$$\min_{\mathbf{X} \in R^n} \left( \Phi(\mathbf{X}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{X}) \right)$$

Необходимым условием существования локального минимума этой задачи в некоторой

точке  $\mathbf{X}^* \in R^n$  является условие  $\nabla_{\mathbf{X}} \left( \Phi(\mathbf{X}^*) + \sum_{i=1}^m \lambda_i g_i(\mathbf{X}^*) \right) = 0$ . (см. Теорему 2.1).

Широко известна другая форма теоремы 1, которую мы сформулируем в виде следствия этой теоремы.

Следствие. В условиях теоремы 1 существуют такие неотрицательные множители Лагранжа  $\lambda_i, i \in [1, m]$ , что имеют место следующие равенства:

$$\nabla_{\mathbf{X}} L(\mathbf{X}^*, \lambda) = 0; \quad (4)$$

$$\nabla_{\mathbf{X}} L(\mathbf{X}^*, \lambda) = g(\mathbf{X}^*) \geq 0 \bullet \quad (5)$$

Здесь равенство (5) повторяет равенство (4), а справедливость равенства (6) следует из того факта, что по условиям теоремы точка  $\mathbf{X}^*$  удовлетворяет всем ограничениям, т.е.  $g_i(\mathbf{X}^*) \geq 0, i \in [1, m]$ ,

Заметим, что из (6) следует справедливость еще одного полезного равенства

$$\lambda^T \nabla_{\lambda} L(\mathbf{X}^*, \lambda) = \sum_{i=1}^m \lambda_i g_i(\mathbf{X}^*) = \lambda^T g(\mathbf{X}^*) \geq 0.$$

## 1.6. Теорема Куна-Таккера для общей задачи нелинейного программирования

Рассмотрим общую задачу нелинейного программирования

$$\min_{\mathbf{X} \in D} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где  $\Phi(\mathbf{X})$  – произвольная функция,

$$D = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0, h_j(\mathbf{X}) = 0, i \in [1, m], j \in [1, l]\} \subset R^n. \quad (2)$$

не пустое ограниченное замкнутое множество.

Нам понадобятся далее понятия множителей Лагранжа и функции Лагранжа для общей задачи нелинейного программирования. Функция Лагранжа для задачи (1) с ограничениями (2) определяется формулой

$$L(\mathbf{X}, \lambda, \mu) = \Phi(\mathbf{X}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{X}) + \sum_{j=1}^l \mu_j h_j(\mathbf{X}) = \Phi(\mathbf{X}) + \lambda^T g(\mathbf{X}) + \mu^T h(\mathbf{X}),$$

где  $\lambda = (\lambda_i, i \in [1, m])$ ,  $\mu = (\mu_j, j \in [1, l])$ ,  $(m \times 1)$ - и  $(l \times 1)$ - векторы множителей Лагранжа, соответственно.

Нам понадобится также понятие условий регулярности для общей задачи нелинейного программирования. Если точка  $\mathbf{X}^* \in D$  и ограничения  $g_{ij}(\mathbf{X}^*) \geq 0, i \in [1, s], j \leq m$  являются активными ограничениями, то условие линейной независимости векторов  $\nabla g_{ij}(\mathbf{X}^*) \geq 0, i \in [1, s]$ , а также условие линейной независимости векторов  $\nabla h_j(\mathbf{X}^*)$ ,  $j \in [1, l]$  называются условиями регулярности ограничивающих функций в точке  $\mathbf{X}^*$ . Смысл условий регулярности раскрыт в предыдущих параграфах.

Теорема 1 (теорема Куна-Таккера). Пусть функции  $\Phi(\mathbf{X})$ ,  $g_i(\mathbf{X}) \geq 0, i \in [1, m]$ ,  $h_j(\mathbf{X}), j \in [1, l]$  имеют непрерывные частные производные в некоторой окрестности точки  $\mathbf{X}^* \in D$  и пусть эта точка является точкой локального минимума функции  $\Phi(\mathbf{X})$ . Пусть, кроме того, выполняются условия регулярности ограничивающих функций  $g(\mathbf{X})$ ,  $h(\mathbf{X})$  в точке  $\mathbf{X}^*$ . Тогда существуют такие множители Лагранжа  $\lambda_i, i \in [1, m]$ ,  $\mu_j \geq 0, j \in [1, l]$ , не все из которых равны нулю одновременно, что для функции Лагранжа  $L(\mathbf{X}, \lambda, \mu)$  точка  $\mathbf{X}^*$  является стационарной точкой функции, т.е.

$$\nabla_{\mathbf{X}} L(\mathbf{X}^*, \lambda, \mu) = \nabla \Phi(\mathbf{X}^*) + \sum_{i=1}^m \lambda_i \nabla g_i(\mathbf{X}^*) + \sum_{j=1}^l \mu_j \nabla h_j(\mathbf{X}^*) = 0. \quad (3)$$

Теорема 1 означает, что в ее условиях вместо задачи условной оптимизации (1), (2) можно решать задачу безусловной оптимизации

$$\min_{\mathbf{X} \in R^n} \left( \Phi(\mathbf{X}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{X}) + \sum_{j=1}^l \mu_j h_j(\mathbf{X}) \right)$$

Необходимым условием существования локального минимума этой задачи в некоторой

$$\text{точке } \mathbf{X}^* \in D \text{ является условие } \Phi(\mathbf{X}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{X}) + \sum_{j=1}^l \mu_j \nabla h_j(\mathbf{X}) = 0 \quad (\text{см. Теорему 2.1}).$$

## 1.7. Аналитическое решение многомерных задач нелинейного программирования

Ограничимся рассмотрением задачи нелинейного программирования с ограничениями типа неравенств

$$\min_{\mathbf{X} \in D} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где  $\Phi(\mathbf{X})$  – произвольная функция,

$$D = \{\mathbf{X} | g_i(\mathbf{X}) \geq 0, i \in [1, m]\} \subset R^n. \quad (2)$$

не пустое, ограниченное замкнутое множество.

Рассмотрим прямое решение задачи (1), (2) (без использования теоремы Куна-Таккера), а также решение этой задачи на основе использования теоремы Куна-Таккера

Прямое решение (без использования теоремы Куна-Таккера).

Общая схема прямого решения задачи нелинейного программирования:

Из условия  $\nabla \Phi(\mathbf{X}^*) = 0$  определяем все стационарные точки функции  $\Phi(\mathbf{X})$  в области  $D$ ;

Определяем все критически точки функции (точки не дифференцируемости) функции  $\Phi(\mathbf{X})$  в области  $D$ ;

Для каждой из границ области  $D$  (ограничивающих функций) решаем соответствующую задачу на условный минимум:

из уравнения  $g_i(\mathbf{X}) \geq 0, i \in [1, m]$  выражаем  $m$  переменных через остальные  $(n-m)$  переменных и подставляем их в выражение для функции  $\Phi(\mathbf{X})$ ;

вместо исходной задачи условной оптимизации получаем задачу безусловной оптимизации с  $(n-m)$  переменными;

решаем эту задачу – находим стационарные точки полученной функции, лежащие на соответствующей границе области  $D$ ;

Решаем задачу, аналогичную задаче, рассмотренной в п.3, для каждого из множеств, которое определяется пересечением границ областей  $D$ ;

Во всех отобранных точках вычисляем значения функции  $\Phi(\mathbf{X})$  и выбираем ту (или те), в которой значение функции наименьшее •

Заметим, что в общем случае такой подход трудно реализовать на практике, поскольку далеко не всегда удается разрешить уравнения  $g_i(\mathbf{X}) \geq 0, i \in [1, m]$  относительно указанных переменных.

Решение с использованием теоремы Куна-Таккера.

Общая схема решения задачи нелинейного программирования с использованием теоремы Куна-Таккера:

Записываем функцию Лагранжа

$$L(\mathbf{X}, \lambda) = \Phi(\mathbf{X}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{X}); \quad (3)$$

Находим градиенты  $\nabla \Phi(\mathbf{X})$ ,  $\nabla g_i(\mathbf{X}), i \in [1, m]$  функций  $\Phi(\mathbf{X})$ ,  $g_i(\mathbf{X}), i \in [1, m]$ ;

Находим стационарные точки функции Лагранжа, т.е. точки, в которых градиент этой функции равен нулю:

$$\nabla_{\mathbf{X}} L(\mathbf{X}^*, \lambda) = \nabla \Phi(\mathbf{X}) + \sum_{i=1}^m \lambda_i \nabla g_i(\mathbf{X}) = 0; \quad (4)$$

Находим точки, в которых нарушаются условия регулярности ограничивающих функций. Во всех стационарных точках функции, а также точках нарушения условий регулярности ограничивающих функций вычисляем значения функции  $\Phi(\mathbf{X})$  и выбираем ту (или те), в которой значение функции наименьшее

## 2.1. Модели управления запасами

Любые предприятия для нормального функционирования должны иметь предмет труда в виде сырья, основных и вспомогательных материалов и полуфабрикатов. Предприятия, фирмы имеют различные запасы: сырье, комплектующие изделия, готовую продукцию, предназначеннную для продажи, и т.д. Совокупность подобных материалов, представляющих временно не используемые экономические ресурсы, называют **запасами предприятия**. Другими словами, под запасом понимаем все на что имеется спрос и что временно выключено из производства.

Рассмотрим простейшие математические модели управления запасами. На рис. 1 представлены возможные графики изменения запаса  $Q$ , имеющегося на складе, во времени  $t$ , для которого рассматривается этот запас.

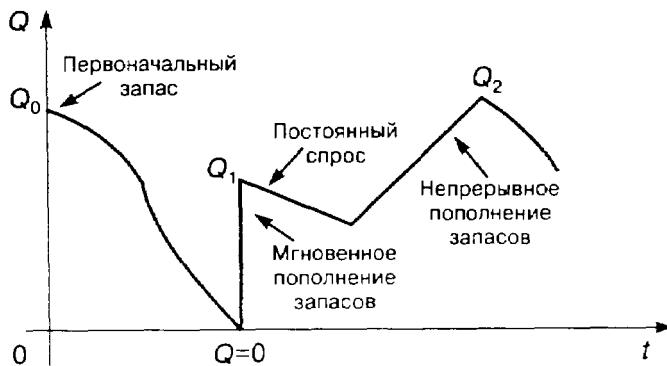


Рис.1

Под  $Q$  будем понимать изделия или материалы (товары) только одного вида. Если на изделие поступает заявка, то оно отпускается и значение  $Q$  падает. Предположим, что величина спроса непрерывна во времени. Если  $Q=0$ , то имеет место дефицит.

Любая математическая модель, которая применяется для изучения определенной ситуации в управлении запасами, должна учитывать факторы, связанные с издержками.

Различают **организационные издержки** — расходы, связанные с оформлением и доставкой товаров, издержки **содержания запасов** — затраты, связанные с хранением. Они возникают из-за амортизации в процессе хранения (изделия могут портиться, устаревать, их количество может уменьшаться и т.д.). Существуют издержки, связанные с **дефицитом**: если поставка со склада не может быть выполнена, то возникают дополнительные издержки, связанные с отказом. Это может быть денежный штраф или ущерб, не осозаемый непосредственно (например, ухудшение бизнеса в будущем и потеря потребителей). Рассматривают также издержки, связанные с приобретением запасов. Их учитывают, если цена единицы продукции зависит от величины партии. Количество товара, поставляемое на склад, называют **размером партии**.

Задача управления запасами состоит в определении объемов поставок и периодичности заказов, при которых издержки (функция затрат) принимают минимальное значение.

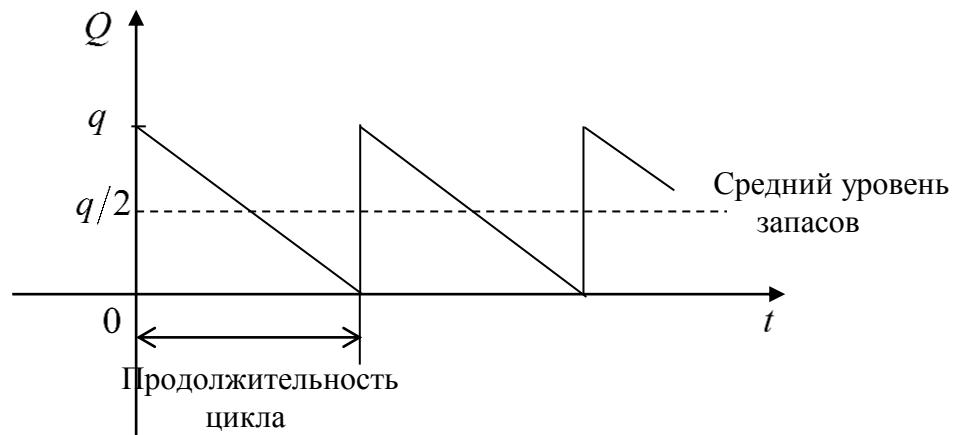
Основная модель управления запасами.

Введем обозначения необходимых для составления модели величин. Данные поместим в таблицу.

Величина	Обозначение	Единица Измерения	Предложения
Интенсивность	$v$	Единиц товара в	Спрос постоянен и непрерывен; весь

спроса		год	спрос удовлетворяется
Организационные издержки	$k$	Рублей за год	Издержки постоянны, не зависят от размера партии
Стоимость товара	$S$	Рублей за год	Цена единицы товара постоянна; рассматривается один вид товара
Издержки содержания запасов	$h$	Рублей за единицу товара в год	Стоимость хранения единицы товара в течение года постоянна
Размер партии	$Q$	Единиц товара в одной партии	Размер партии постоянен; поступление товара происходит мгновенно, как только уровень запаса равен нулю

График изменения запасов представлен на рис. 2.



Чтобы полностью удовлетворить годовой спрос  $V$  при размере поставки  $q$ , необходимо обеспечить  $V/q$  поставок или партий за год. Средний уровень запасов составляет  $q/2$ .

Уравнение издержек будет иметь вид

$$L = L_1 + L_2 + L_3 = kV/q + SV + hq/2,$$

где  $L_1$  — общие организационные издержки;  $L_2$  — стоимость товаров;  $L_3$  — общие издержки содержания запасов.

За исключением  $q$  все величины в правой части уравнения постоянны и известны, т.е.  $L = f(q)$ . Для нахождения минимума  $L$  найдем производную  $\frac{dL}{dq}$  и приравняем ее к нулю:

$$\frac{dL}{dq} = -\frac{kV}{q^2} + \frac{h}{2} = 0,$$

откуда

$$q_{\text{опт}} = \sqrt{\frac{2kV}{h}} \quad (1)$$

где  $q_{\text{опт}}$  — оптимальный размер партии. Равенство (1) называется **формулой Уилсона**.

## 2.2.Модель производственных запасов.

В основной модели предполагали, что поступление товаров на склад происходит мгновенно, например, в течение одного дня. Рассмотрим случай, когда готовые товары поступают на склад непосредственно с производственной линии. Будем считать, что поступление товаров происходит непрерывно. Модель задачи в этом случае называют моделью **производственных поставок**. Обозначим через  $\lambda$  скорость поступающего на склад товара. Эта величина равна количеству товаров, выпускаемых производственной линией за год. Остальные обозначения и предположения те же, что и для основной модели управления запасами.

Определим оптимальный размер партии, минимизирующей общие затраты.

График изменения модели производственных запасов представлен на рис. 3.

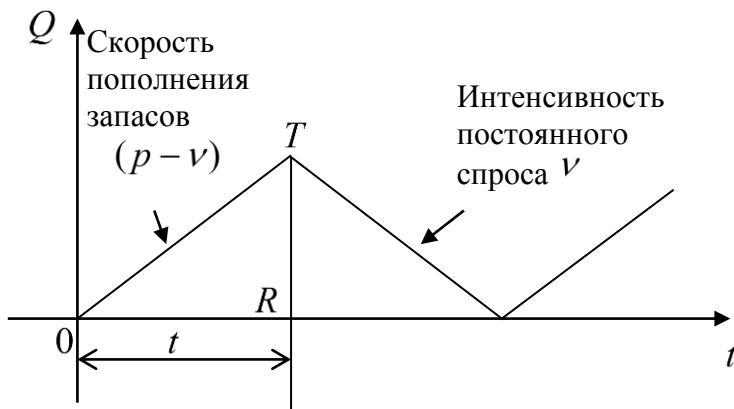


Рис. 3

Общие издержки в течение года, как и для основной модели, составляют

$$L = L_1 + L_2 + L_3,$$

$$L_1 = \frac{k\nu}{q},$$

$$L_2 = s\nu.$$

Для получения среднего уровня запасов следует учесть, что  $RT = (\lambda - \nu)t$  — максимальный уровень запасов,

$q = \lambda t$  — количество товаров в одной производственной поставке.

Тогда средний уровень запасов составляет половину максимального и равен

$$\frac{(\lambda - \nu)q}{2\lambda}.$$

В итоге

$$L = \frac{k\nu}{q} + s\nu + \frac{(\lambda - \nu)}{2\lambda}.$$

Решая уравнение  $\frac{dL}{dq} = 0$ , найдем оптимальный размер партии модели производственных поставок:

$$q_{\text{опт}} = \sqrt{\frac{2\lambda k\nu}{(\lambda - \nu)h}}.$$

### 2.3. Модель запасов, включающая штрафы.

Рассмотрим основную модель, допускающую возможность существования периодов дефицита, который покрывается при последующих поставках, и штрафов за несвоевременную поставку.

Пусть предприятие должно поставить  $q$  ед. товара в течение каждого промежутка времени  $T$  за единицу времени поставляется  $V$  ед. товара ( $q = TV$ ).

Предположим, что в начале каждого периода  $T$  предприятие делает запас; равный  $S$ . Это означает, что в течение периода будет наблюдаться дефицит товара и некоторое время поставки не будут осуществляться. Невыполненные заявки будут накапливаться до максимальной величины  $q - S$  и будут удовлетворены, как только поступит следующая партия товаров в количестве  $q$ .

За то, что товары доставляются предприятием позже необходимого срока, на предприятие налагается штраф, который зависит от того, насколько была задержана поставка. Такая модель целесообразна, поскольку иногда выгоднее заплатить штраф, чем расходовать дополнительные средства на хранение запасов, превышающих величину  $S$ .

Задача управления запасами состоит в том, чтобы выбрать такое значение  $S$ , которое ведет к минимизации всех затрат, включая затраты на хранение и штрафы.

График изменения запасов модели представлен на рис. 4.

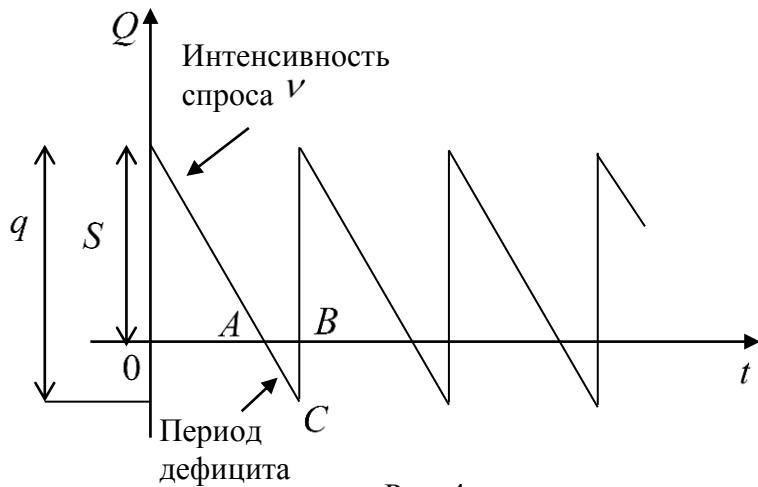


Рис. 4

Для определения оптимального значения  $S$  обозначим:

$h$  — издержки хранения единицы товара за единицу времени;

$\lambda$  — затраты на штраф в расчете на единицу товара за один день отсрочки.

Найдем издержки одного цикла:

$$L = L_1 + L_2,$$

где  $L_1$  — общие издержки содержания запасов;  $L_2$  — общие затраты на штраф.

Так как товары находятся на складе в течение периода OA (см. рис. 5.4), средний уровень запасов за этот период равен  $S/2$ . Если продолжительность периода OA равна  $S/V$ , то

$$L_1 = h \frac{S}{2} \cdot \frac{S}{V} = \frac{hS^2}{2V}.$$

Так как штраф выплачивается в течение периода  $AB = \frac{(q - S)}{v}$  общее число "товародней", на которые налагается штраф, равно площади треугольника  $ABC$ . Площадь составляет

$$\frac{(q - S)}{v} \cdot \frac{(q - S)}{2},$$

откуда

$$L_2 = \frac{\lambda(q - S)^2}{2v}.$$

Окончательно

$$L = \frac{hS^2}{2v} + \frac{\lambda(q - S)^2}{2v}.$$

Найдем  $\frac{dL}{ds}$  и, решив уравнение  $\frac{dL}{ds} = 0$ , получим оптимальное значение:

$$S_{\text{опт}} = \frac{\lambda q}{(h + \lambda)}.$$

Взяв  $S_{\text{опт}}$  в качестве уровня запасов в начале каждого цикла при условии, что невыполненные заявки будут удовлетворены, сведем суммарные расходы  $L$  к минимуму:

$$L_{\min} = \frac{q^2 h \lambda}{2v(h + \lambda)}.$$

## 2.4. Точка заказа.

В реальных задачах следует учитывать время выполнения заказа  $\theta$ . Для бесперебойного снабжения заказ должен подаваться в момент, когда уровень запаса достаточен для удовлетворения потребности на время выполнения заказа. Этот уровень называется **точкой возобновления заказа (точка заказа)** и обозначается  $r$ , т.е. это нижний уровень, по которому мы должны заказывать новую партию. Для систем, в которых дефицит не допускается, заказ должен размещаться в момент когда величина наличного запаса равна

$$r = \theta v - \left[ \frac{\theta}{\tau_{\text{опт}}^*} \right] q_{\text{опт}},$$

где  $[\cdot]$  — целая часть  $(\cdot)$ ,  $\tau_{\text{опт}}^*$  — оптимальный интервал между поставками.

Для бездефицитной работы системы нужно иметь начальный запас  $I_0 = \theta v$ . Если  $I$  — фактический запас, то для непрерывной работы необходимо, чтобы  $I \geq \theta v$ . Время потребления начального запаса  $\frac{I}{v}$ . Чтобы заказанная партия прибыла ко времени полного исчерпания , ее нужно размещать в момент  $t_0 = \frac{I}{v} - \theta$ , а все остальные заказы нужно размещать в моменты

$$t_k = \left( \frac{I}{v} - \theta \right) + k \tau_{\text{опт}}, \quad k = 0, 1, 2, \dots$$

Для систем с дефицитом точка заказа определяется по формуле

$$r = \theta v - \left[ \frac{\theta}{\tau_{\text{опт}}^*} \right] q_{\text{опт}} - S_{\text{опт}}.$$

### 3.1. Классификация методов решения детерминированных задач оптимизации

Особенность задач оптимизации в САПР состоит в том, что вычисление значения критерия оптимальности и значений ограничивающих функций при фиксированных значениях параметров  $\mathbf{X}$  может требовать больших затрат компьютерного времени. В связи с этим возникает проблема решения задачи оптимизации при наименьшем числе испытаний.

Рассмотрим для простоты записи только ограничений типа неравенств:

$$\min_{\mathbf{X} \in D} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*), D = \{\mathbf{X} | g(\mathbf{X}) \geq 0\}.$$

Испытанием называется операция однократного вычисления функций  $\Phi(\mathbf{X}), g(\mathbf{X})$ , и, быть может, их производных, в некоторой точке  $\mathbf{X}$ , вообще говоря, не обязательно принадлежащей множеству  $D$ .

Далее будем говорить, что детерминированная задача оптимизации решается с помощью поискового метода оптимизации, если используется следующая процедура поиска оптимального решения  $\mathbf{X}^*$ :

по очереди при  $r=0,1,2,\dots,N-1$  производятся испытания в точках

$$\mathbf{X}^0, \mathbf{X}^{r+1} = F_{r+1}(\mathbf{X}^0, \Phi(\mathbf{X}^0), g(\mathbf{X}^0), \dots, \mathbf{X}^r, \Phi(\mathbf{X}^r), g(\mathbf{X}^r)), \quad (1)$$

где  $\mathbf{X}^0$  — начальное приближение;

в качестве решения задачи берется вектор  $\mathbf{X}^*$ , который находится из условия

$$\Phi^* = \Phi(\mathbf{X}^*) = \min_{r \in [0, N]} \Phi(\mathbf{X}^r). \quad (2)$$

Классификация по наличию или отсутствию ограничений на вектор варьируемых параметров.

Метод поиска, ориентированный на решение задач безусловной оптимизации, называется методом безусловной оптимизации. Аналогично, метод поиска, ориентированный на решение задач условной оптимизации, называется методом условной оптимизации.

Классификация по размерности вектора  $\mathbf{X}$ .

Если в формулах (1), (2)  $\mathbf{X}^r$  есть скаляр, то метод поиска называется одномерным методом поиска; если  $\mathbf{X}^r$  есть вектор ( $n > 1$ ), то метод поиска называется многопараметрическим методом поиска.

Классификация по характеру искомого решения.

Если метод поиска гарантирует отыскание только локального минимума функции  $\Phi(\mathbf{X})$ , то метод называется методом локального поиска. Если делается попытка отыскать глобальный минимум  $\Phi(\mathbf{X})$ , то метод называется методом глобального поиска. Сразу отметим, что удовлетворительных с точки зрения вычислительной эффективности методов глобального поиска не существует.

Классификация по характеру функции  $F_r$ .

Если функции  $F_r, r \in [1, N]$  являются детерминированными, то метод поиска называется детерминированным методом поиска. Если функции  $F_r$  содержат случайные параметры, то метод поиска называется стохастическим методом поиска.

Методы пассивного и последовательного поиска.

Если все точки  $\mathbf{X}^r, r=0,1,2,\dots,N$  назначаются заранее (до проведения испытаний), то метод поиска называется пассивным методом поиска. Если точка  $\mathbf{X}^{r+1}$  определяется на

основе всей или части информации об испытаниях в предыдущих точках, то метод называется последовательным методом поиска.

Классификация по количеству предыдущих учитываемых шагов.

Если при вычислении координат точки  $X^{r+1}$  учитывается информация только об одном (предыдущем) испытании, то метод поиска называется одношаговым методом поиска.

Схема одношагового последовательного метода поиска:

$$X^{r+1} = F_{r+1}(X^r, \Phi(X^r), g(X^r)).$$

Если при вычислении координат точки  $X^{r+1}$  учитывается информация о  $s > 1$  предыдущих испытаниях, то метод поиска называется многошаговым методом поиска (конкретнее,  $s$ -шаговым).

Классификация по виду функций  $F_r$ .

Если функции  $F_r$  на всех  $N$  шагах одинаковы, то метод поиска называется итерационным методом поиска.

Схема одношагового итерационного метода последовательного поиска:

$$X^{r+1} = F(X^r, \Phi(X^r), g(X^r)).$$

Если функции  $F_r$  различны на различных шагах поиска, то метод называется не итерационным методом поиска.

Классификация по "близости" точек соседних точек, в которых производятся испытания.

Если точка  $X^{r+1}$  принадлежит некоторой малой окрестности точки  $X^r$ , т.е.  $X^{r+1} \in d(X^r)$ , то метод поиска называется локальным методом поиска. Если точка  $X^{r+1}$  может принадлежать любой точки множества  $D$ , т.е.  $X^{r+1} \in D$ , то метод поиска называется нелокальным методом поиска.

Классификация по порядку используемых производных.

Если при вычислении значений функции  $F_r$  производные не используются, то метод поиска называется прямым методом поиска или методом поиска нулевого порядка. Если при этом используются производные  $k$ -го порядка, то метод поиска называется методом поиска k-го порядка. Метод поиска первого порядка называется также градиентным методом поиска.

Способ выбора начальной точки  $X_0$  и конкретная совокупность функций  $\{F_r\}$  называются алгоритмом поисковой оптимизации. Таким образом, понятие алгоритма является более частным по сравнению с понятием метода (одному и тому же методу могут соответствовать различные алгоритмы).

Важной проблемой при построении методов решения задач оптимизации является проблема выбора условия окончания поиска (критерия окончания поиска). Простейшими, но широко используемыми в вычислительной практике, являются следующие критерии окончания поиска:

$$\|X^{r+1} - X^r\| \leq \varepsilon_X, \quad (3)$$

где  $\varepsilon_X$  — константа, определяющая требуемую точность решения по  $X$ ;

$$|\Phi(X^{r+1}) - \Phi(X^r)| \leq \varepsilon_\Phi, \quad (4)$$

где  $\varepsilon_\Phi$  — константа, определяющая требуемую точность решения по  $\Phi$ . Здесь  $\|\bullet\|$  — некоторая векторная норма (например, евклидова).

Будем далее условия окончания поиска (3), (4) называть стандартными условиями окончания поиска (стандартными критериями окончания поиска).

### 3.2. Наилучшие алгоритмы оптимизации

Конкретную детерминированную задачу оптимизации можно решить различными алгоритмами. Отсюда возникают вопросы:

какой алгоритм выбрать?

какой алгоритм является «наилучшим»?

Ответ на эти вопросы возможен только в том случае, когда определен класс функций  $\{\Phi(\mathbf{X})\}$ , которому принадлежит критерий оптимальности  $\Phi(\mathbf{X})$ . Без определения этого класса ответить на поставленные вопросы невозможно – нет алгоритма, наилучшего для всех возможных функций  $\Phi(\mathbf{X})$ .

Примеры классов функций:

класс функций с ограниченным числом минимумов;

класс функций, с ограниченными некоторой величиной первыми частными производными;

класс выпуклых функций.

Множество рассматриваемых алгоритмов оптимизации обозначим  $\{\mathcal{A}\}$ .

Для формальной постановки задачи определения наилучшего алгоритма из множества  $\{\mathcal{A}\}$  на классе функций  $\{\Phi(\mathbf{X})\}$  необходимо еще ввести критерий качества алгоритма оптимизации. Обозначим этот критерий  $W(\Phi, \mathcal{A})$ , где  $\Phi \in \{\Phi(\mathbf{X})\}$ ,  $\mathcal{A} \in \{\mathcal{A}\}$ . Положим, что оптимальным является наименьшее значение этого критерия.

Для построения критерия качества алгоритма на всем классе функций  $\{\Phi(\mathbf{X})\}$  можно воспользоваться

принципом гарантированного результата:

$$W(\mathcal{A}) = \max_{\Phi \in \{\Phi(\mathbf{X})\}} W(\Phi, \mathcal{A});$$

некоторым средним значением критерия качества алгоритма на классе функций  $\{\Phi(\mathbf{X})\}$ .

Если критерий качества алгоритма на классе функций  $\{\Phi(\mathbf{X})\}$  тем или иным образом определен, то задача отыскания наилучшего алгоритма оптимизации на этом классе функций формально может быть записана в следующем виде:

$$W(\mathcal{A}^*) = \max_{\mathcal{A} \in \{\mathcal{A}\}} W(\mathcal{A}); \quad (1)$$

Теоретическое решение задачи (1) возможно только для алгоритмов поиска минимума одномерных унимодальных функций. В более сложных случаях сравнение алгоритмов поисковой оптимизации и выбор из них наилучшего производится на основе экспериментального тестирования.

### 3.3. Экспериментальное тестирование алгоритмов оптимизации

В качестве критерия качества алгоритма оптимизации  $W(\Phi, A)$  обычно рассматривают затраты времени на поиск. Эти затраты складываются из затрат на испытания из затрат на нахождение точек  $\mathbf{X}^r$  по информации о предыдущих испытаниях (можно сказать – из затрат на вычисления значений функции  $F$ ).

Поскольку в задачах САПР последние затраты много меньше первых, в качестве критерия качества алгоритма оптимизации  $A$  можно использовать количество испытаний  $N(\Phi, A, \varepsilon, \mathbf{X}^0)$ , необходимых для нахождения минимума функции  $\Phi(\mathbf{X})_c$  заданной точностью  $\varepsilon$  при начальном приближении  $\mathbf{X}^0$ .

Для корректного сравнения эффективности различных алгоритмов, экспериментальное тестирование алгоритмов оптимизации необходимо выполнять при одинаковых значениях заданной точности решения  $\varepsilon$ . Поэтому будем в качестве критерия качества алгоритма оптимизации  $A$  на классе функций  $\{\Phi(\mathbf{X})\}$  использовать критерий  $N(\Phi, A, \mathbf{X}^0)$ .

При заданной точности решения эффективность любого алгоритма поисковой оптимизации зависит от начального приближения  $\mathbf{X}^0$ . Поэтому при экспериментальном тестировании обычно критерий  $N(\Phi, A, \mathbf{X}^0)$  усредняют по множеству допустимых значений вектора варьируемых параметров  $D$ . Обозначим такой критерий  $\bar{N}(\Phi, A)$ .

Точность решения задачи оптимизации определяется используемым условием окончания поиска. При экспериментальном тестировании в качестве такого условия обычно используют одно из двух следующих условий:

$$\min_{r \in [0, N]} \left| \Phi(\mathbf{X}^r) - \Phi^* \right| = \left| \Phi(\mathbf{X}^*) - \Phi^* \right| \leq \varepsilon_\Phi,$$

где  $\Phi^*$  - значение функции  $\Phi(\mathbf{X})$  в точке истинного минимума;

$$\|(\mathbf{X}^r)^* - \mathbf{X}^*\| \leq \varepsilon_X,$$

где  $\mathbf{X}^*$  - точка истинного минимума функции  $\Phi(\mathbf{X})$ ,  $\|\cdot\|$  - некоторая векторная норма.

Заметим, что эти условия накладывают на класс тестовых функций дополнительное требование: у этих функций априори должно быть известно положение истинного

минимума .

Первоначально в качестве тестовых функций использовали одну или несколько наиболее «трудных» тестовых функций (многоэкстремальных, овражных и т.п.). В настоящее время для оценки качества алгоритмов поисковой оптимизации обычно используют среднее

значение критерия  $\bar{N}(\Phi, A)$  на классе функций  $\{\Phi(\mathbf{X})\}$ . Обозначим такой критерий  $\bar{\bar{N}}(A)$  через .

Тогда формально задачу поиска наилучшего алгоритма оптимизации можно записать в следующем виде:

$$\bar{\bar{N}}(A) = \min_{A \in (A)} \bar{\bar{N}}(A). \quad (1)$$

Особенности задачи (1):

высокая дисперсия оценки  $\bar{N}(A)$ ;

функция  $\bar{N}(A)$  может оказаться многоэкстремальной;

может оказаться, что при некоторых  $X^0 \in D$  решение задачи минимизации затягивается сверх допустимого времени.

Последняя особенность приводит к тому, что кроме критерия  $N(\Phi, A)$  приходится использовать еще вероятность локализации минимума функции  $\Phi(X)$  за допустимое количество испытаний  $N^*$ . Таким образом, вообще говоря, задача экспериментального тестирования является задачей многокритериальной оптимизации (точнее, задачей двухкритериальной оптимизации).

Сделаем следующие предположения:

множество тестируемых алгоритмов  $\{A\}$  состоит из  $n_A$  алгоритмов  $A_i, i \in [1, n_A]$ ;

при тестировании алгоритма  $A_i, i \in [1, n_A]$  используется совокупность  $n_\Phi$  тестовых функций  $\Phi_{i,j}(X) \in \{\Phi(X)\}, j \in [1, n_\Phi]$ ;

при тестировании алгоритма  $A_i, i \in [1, n_A]$  с помощью функции  $\Phi_{i,j}(X) \in \{\Phi(X)\}, j \in [1, n_\Phi]$  используется  $n_X$  начальных приближений вектора варьируемых параметров  $X_{i,j,k}^0 \in D, k \in [1, n_X]$ .

В сделанных предположениях общую схему экспериментального тестирования алгоритмов оптимизации можно представить в следующем виде (см. рис. 1).

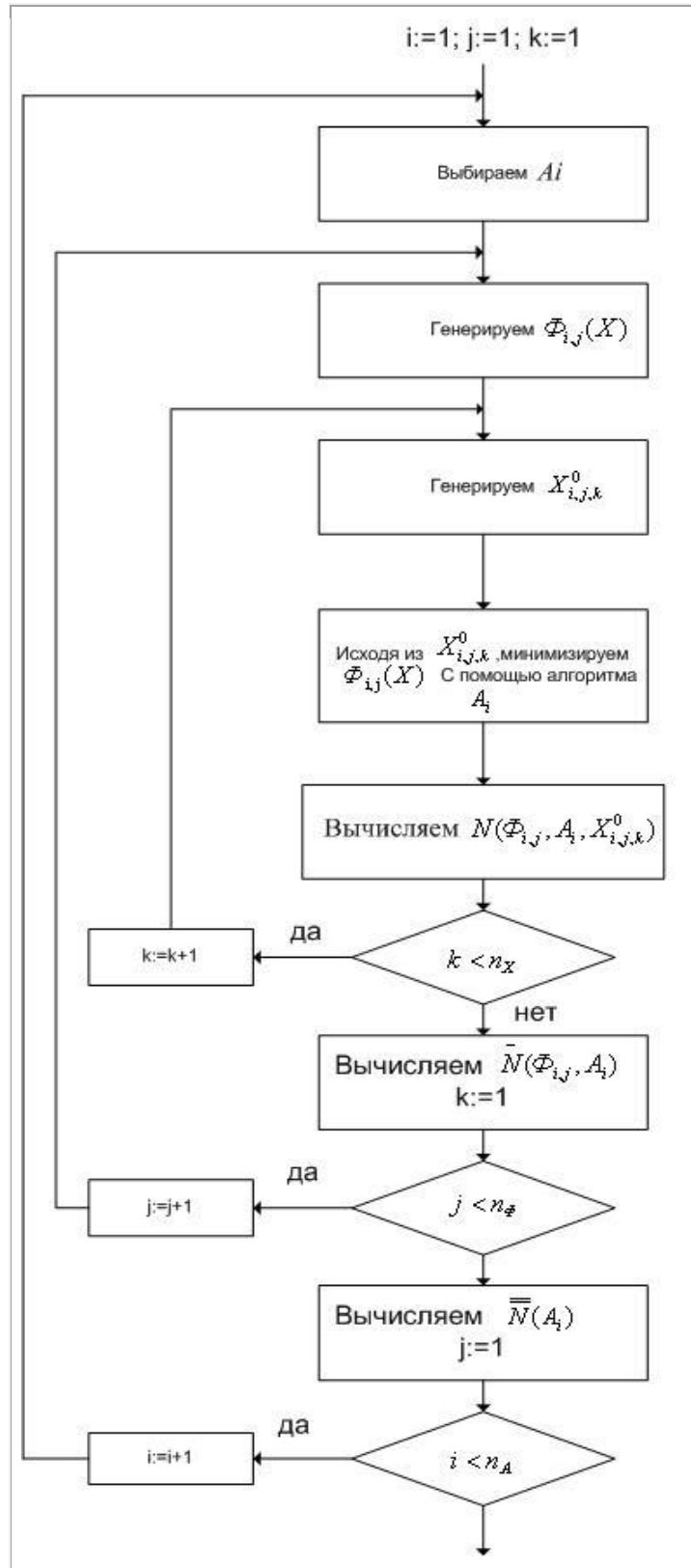


Рис. 1. Общая схема экспериментального тестирования алгоритмов поисковой оптимизации.

Заметим, что для получения статистически достоверных оценок значений  $\bar{N}(\Phi, A)$ ,  $\bar{N}(A)$ , величины  $n_\Phi$ ,  $n_X$  должны быть достаточно велики.

### 3.4. Классы тестовых функций

Класс одномерных унимодальных функций.

Положим, что  $D=[a,b]$ . Покроем интервал  $[a,b]$  сеткой с  $M+1$  узлами и обозначим  $\Phi(x_i)=\Phi_i, i \in [0,M]$ .

Положим, что необходимо построить одномерную унимодальную функцию, минимум которой находится в узле  $i^* \in [0,M]$ , т.е. в точке  $x_{i^*}$ . Тогда для генерации такой функции может быть использован следующий алгоритм построения тестовых одномерных унимодальных функций.

Генерируем дискретную случайную величину  $i^*$ , равномерно распределенную на множестве  $0,1,2,\dots,M$ .

Генерируем случайные числа  $\Phi_i$  по рекуррентной формуле

$$\begin{cases} \Phi_i = \Phi_{i-1} + \beta_i \\ \Phi_0 = \beta_0 \end{cases},$$

где  $\beta_i, i \in [0,M]$  - случайные величины, равномерно распределенные в интервале  $(-\alpha, 0)$ , если  $i \leq i^*$ , и в интервале  $(\alpha, 0)$ , если  $i > i^*$ . Здесь  $\alpha$ - некоторая константа, определяющая максимальную скорость изменения искомой функции.

Для получения функции  $\Phi(\mathbf{X})$  используем линейную интерполяцию полученных значений  $\Phi_i, i \in [0,M]$ . Минимум полученной функции  $\Phi(\mathbf{X})$  достигается в точке  $x_{i^*}=x^*$ .

Класс одномерных многоэкстремальных функций.

Положим, то область допустимых значений  $D$  есть интервал  $[a,b]$ . С помощью линейного преобразования

$$x = \frac{-y+a}{a-b}$$

сведем этот интервал к интервалу  $[0,1]$ . Тогда в качестве тестовых одномерных многоэкстремальных функций можно использовать тригонометрические полиномы некоторой степени  $s$  (отрезки ряда Фурье):

$$\Phi(x) = a_0 + \sum_{k=1}^s \{a_k \cos((k-1)\pi x) + b_k \sin((k-1)\pi x)\} \quad (1)$$

Здесь  $a_0, a_k, b_k, k=[1,s]$  случайные величины с некоторыми законами распределения.

Таким образом, алгоритм генерации тестовых одномерных многоэкстремальных функций состоит в генерации указанных случайных величин и использовании формулы (1).

Класс многоэкстремальных функций нескольких переменных.

В простейшем случае алгоритм генерации многоэкстремальных функций нескольких переменных можно построить на основе использования совокупности с квадратичных функций, каждая из которых в общем случае дает локальный минимум искомой функции:

$$\Phi(\mathbf{X}) = \min_{j \in [1,s]} \Phi_j(\mathbf{X}), \quad (2)$$

где квадратичная функция

$$\Phi_j(x) = \sum_{k=1}^n a_{kj} (x_k - x_{kj}^*)^2 - c_j. \quad (3)$$

Здесь  $\mathcal{N}$  – размерность вектора варьируемых параметров  $\mathbf{X}$ ,  $\mathbf{X}_j^* = (x_{1,j}^*, x_{2,j}^*, \dots, x_{n,j}^*)$  – точка  $j$ -го локального минимума глубиной  $c_j$ ; постоянные  $a_{kj}$ ,  $x_{kj}^*$ ,  $c_j$  – случайные величины с некоторыми законами распределения.

#### 4.1. Алгоритм равномерного поиска

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

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

Идея алгоритмов, относящихся к методу сокращения текущего интервала неопределенности, состоит в исключении в процессе поиска из рассмотрения тех подинтервалов, в которых в силу унимодальности функции  $\Phi(x)$  точка  $x^*$  отсутствует. Текущий интервал неопределенности будем обозначать ТИН, а его длину  $|ТИН|$ . Так что, если  $ТИН = [a, b]$ , то  $|ТИН| = b - a$ .

В алгоритме равномерного поиска испытания проводятся в точках, которые определяются путем равномерного деления интервала  $[a, b]$  на  $N$  одинаковых подинтервалов. Из вычисленных значений функции  $\Phi(x)$  выбирается наименьшее. Пусть это значение достигается в точке  $x_k$ . Тогда в связи с унимодальностью функции  $\Phi(x)$  подинтервалы  $[a, x_{k-1}]$ ,  $[x_{k+1}, b]$  можно исключить из рассмотрения, т.е. сделать очередным интервалом неопределенности интервал  $[x_{k-1}, x_{k+1}]$ . Алгоритм относится к классу пассивных методов поиска.

Более строго описанную схему алгоритма можно записать в нижеследующем виде.

Выполняем присваивания  $r=1$ ,  $a^1=a$ ,  $b^1=b$ ,  $ТИН_1=[a^1, b^1]$ .

На текущем ТИН строим равномерную сетку с  $N+1$  узлами (см. рис. 1).

Вычисляем значения функции  $\Phi(x)$  в узлах построенной сетки  $\Phi(x_0^r), \dots, \Phi(x_N^r)$ .

Находим минимальное из этих значений:

$$\min(\Phi(x_0^r), \dots, \Phi(x_N^r)) = \Phi(x_k^r).$$

Выполняем присваивания  $a^{r+1}=x_{k-1}^r$ ,  $b^{r+1}=x_{k+1}^r$ ,  $ТИН_{r+1}=[a^{r+1}, b^{r+1}]$ .

Если  $|ТИН_{r+1}| \leq \varepsilon_x$ , то заканчиваем вычисления. Иначе - выполняем присваивание  $r=r+1$  и переходим на п.2. Здесь  $\varepsilon_x$  – требуемая точность решения •

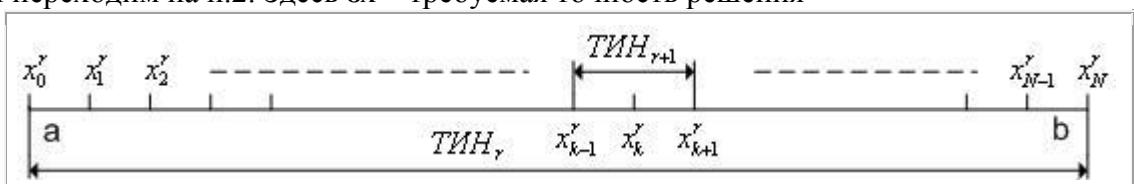


Рис. 1. Построение сетки на текущем интервале неопределенности.

В качестве приближенного значения точки минимума  $x^*$  с равными основаниями может быть принята любая точка последнего текущего интервала неопределенности.

Первую итерацию приведенной схемы алгоритма равномерного поиска иллюстрирует рис. 2.

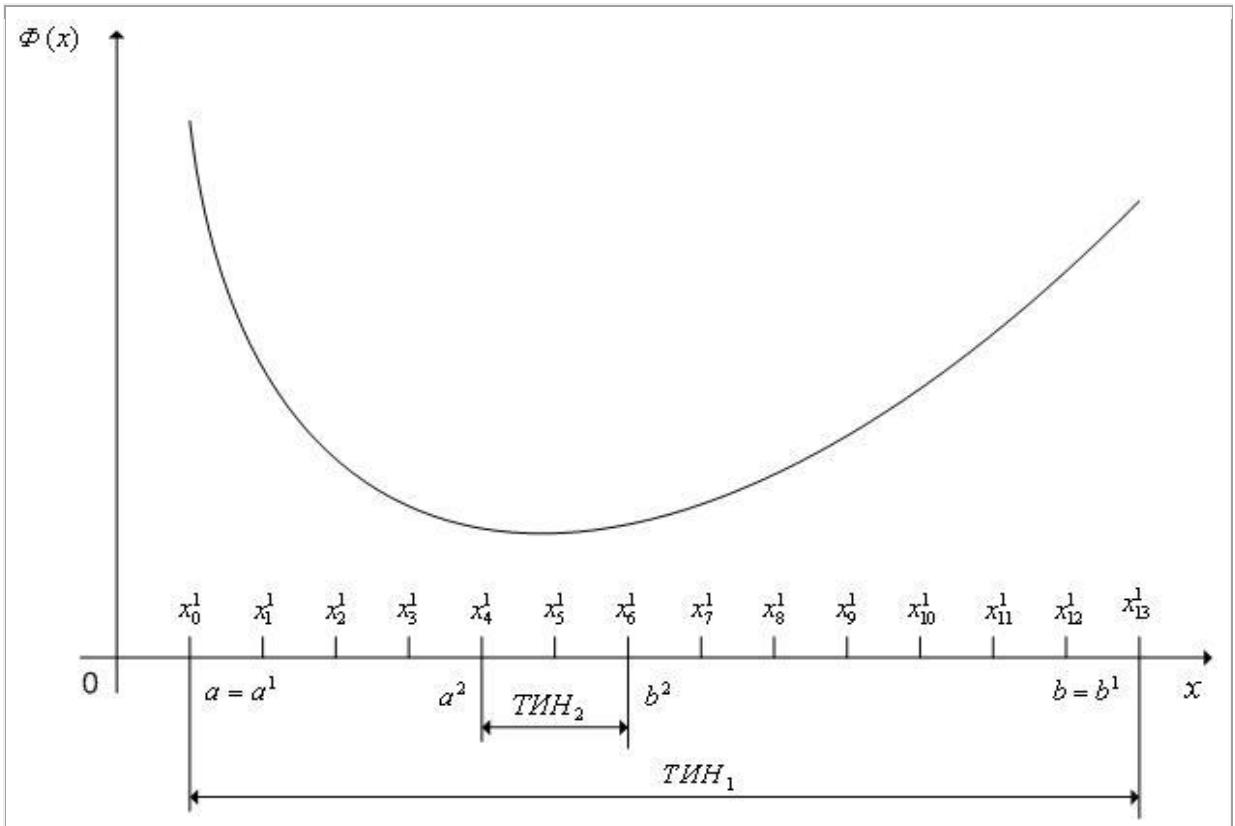


Рис. 2. Первая итерация поиска минимума одномерной унимодальной функции  $\Phi(x)$  с помощью алгоритма равномерного поиска:  $N=13$ .

Легко видеть, что после одной итерации алгоритма равномерного поиска ТИН  $\frac{N}{2}$  уменьшается в  $\frac{1}{2}$  раз. Поэтому количество итераций  $r$ , необходимых для нахождения минимума функции с точностью  $\epsilon_x$ , может быть найдено из условия  $\left(\frac{2}{N}\right)^r(b-a) \leq \epsilon_x$ .

## 4.2. Алгоритм деления пополам

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

$$\min_{X \in [a, b]} \Phi(x) = \Phi(x^*).$$

В алгоритм деления пополам или алгоритме равномерного дихотомического поиска испытания проводятся парами. Координаты каждой последующей пары испытаний разнесены между собой на величину  $\delta_x < \varepsilon_x$ , где  $\varepsilon_x$  - требуемая точность решения. Испытания производятся в середине ТИН. По значениям  $\Phi(x)$ , полученным в этих точках, одна половина ТИН в силу унимодальности функции  $\Phi(x)$  исключается из дальнейшего рассмотрения. Величина  $\delta_x$  определяется требуемой точностью решения. Алгоритм относится к классу методов последовательного поиска.

Более строго описанную схему алгоритма можно записать в нижеследующем виде.

Выполняем присваивания  $r=1$ ,  $a^1=a$ ,  $b^1=b$ ,  $\text{ТИН}_1=[a^1, b^1]$ .

Вычисляем величины (см. рис. 1)

$$x_0^r = \frac{a^r + b^r}{2}, x_1^r = x_0^r - \frac{\delta_x}{2}, x_2^r = x_0^r + \frac{\delta_x}{2}.$$

Вычисляем значения  $\Phi(x_1^r), \Phi(x_2^r)$  функции  $\Phi(x)$ .

Если  $\Phi(x_1^r) < \Phi(x_2^r)$ , то выполняем присваивания  $a^{r+1} = a^r$ ,  $b^{r+1} = x_0^r$ ,  $\text{ТИН}_{r+1} = [a^{r+1}, b^{r+1}]$ . Иначе - выполняем присваивания  $a^{r+1} = x_0^r$ ,  $b^{r+1} = b^r$ ,  $\text{ТИН}_{r+1} = [a^{r+1}, b^{r+1}]$

Если  $|\text{ТИН}_{r+1}| \leq \varepsilon_x$ , то заканчиваем вычисления. Иначе - выполняем присваивание  $r=r+1$  и переходим на п.2. ●

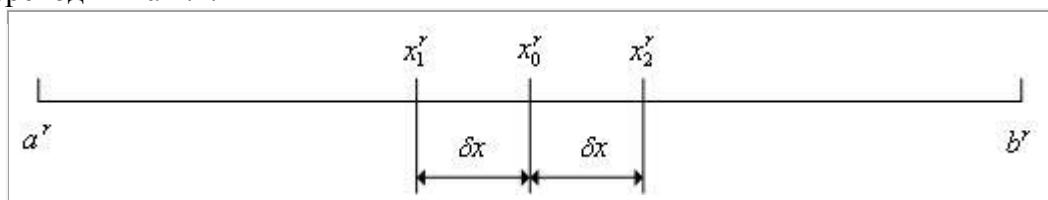


Рис. 1. К определению величин  $x_0^r, x_1^r, x_2^r$ .

В качестве приближенного значения точки минимума  $x^*$  с равными основаниями может быть принята любая точка последнего текущего интервала неопределенности.

Приведенную схему алгоритма равномерного дихотомического поиска иллюстрирует рис. 2.

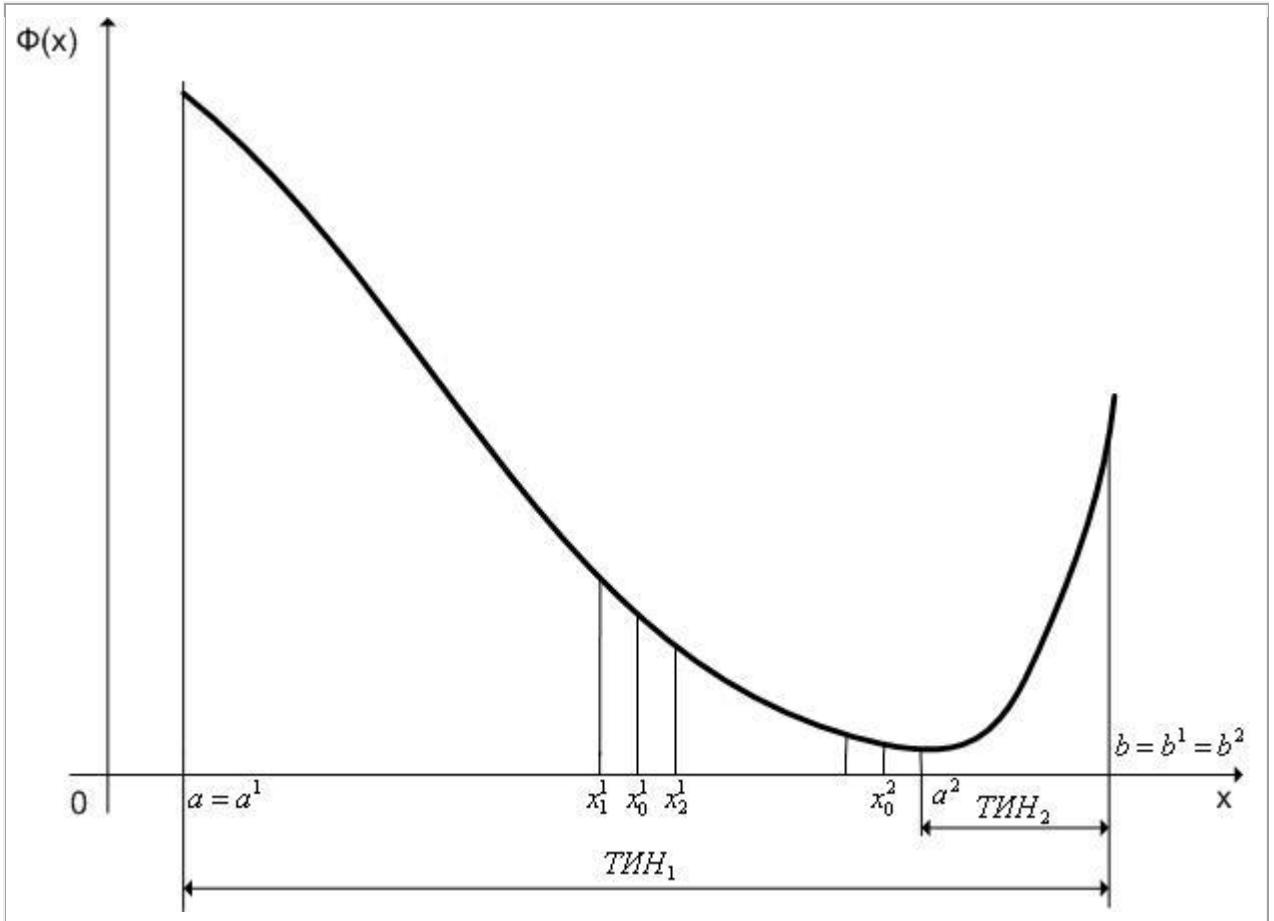


Рис. 2. Первые две итерации поиска минимума одномерной унимодальной функции с помощью алгоритма равномерного дихотомического поиска.

Легко видеть, что после одной итерации алгоритма равномерного поиска ТИН уменьшается в 2 раза. Поэтому количество итераций  $r$ , необходимых для нахождения минимума функции с точностью  $\epsilon_x$ , находится из условия

$$\frac{b-a}{2^r} \leq \epsilon_x.$$

### 4.3. Алгоритм Фибоначчи

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

$$\min_{X \in [a, b]} \Phi(x) = \Phi(x^*).$$

Числа Фибоначчи и их некоторые свойства.

Числа Фибоначчи задаются следующим рекуррентным уравнением:

$$i_N = i_{N-1} + i_{N-2}, \quad N \geq 2, \quad i_0 = i_1 = 1. \quad (1)$$

Числа Фибоначчи  $i_0, \dots, i_9$  приведены в нижеследующей табл. 1.

Таблица 1

$N$	0	1	2	3	4	5	6	7	8	9	...
$i_N$	1	1	2	3	5	8	13	21	34	55	...

Общее выражение для  $N$ -го числа Фибоначчи можно получить из решения уравнения (1):

$$i_N = \frac{\left(\frac{1}{\tau}\right)^{N+1} - (-\tau)^{N+1}}{\sqrt{5}}, \text{ где } \tau = \frac{\sqrt{5}-1}{2} \approx 0,618.$$

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

$$i_N \approx \frac{\left(\frac{1}{\tau}\right)^{N+1}}{\sqrt{5}}. \quad (2)$$

Отсюда следует, что  $\frac{i_{N-1}}{i_N} \approx \tau$ . Т.е. отношение двух соседних чисел Фибоначчи примерно постоянно и равно  $\tau$ .

Алгоритм Фибоначчи.

Алгоритм Фибоначчи относится к классу поисковых методов оптимизации и включает в себя два этапа.

Первый этап состоит из  $(N-1)$ -й итерации для  $r=1, 2, \dots, N-1$ . Рассмотрим схему  $r$ -й итерации, когда  $TIN_r = [a^r, b^r]$ :

Вычисляем

$$x_1^r = a^r + |TIN_r| \frac{i_{N-1-r}}{i_{N+1-r}},$$

$$x_2^r = a^r + |TIN_r| \frac{i_{N-r}}{i_{N+1-r}}.$$

Вычисляем значения  $\Phi(x_1^r), \Phi(x_2^r)$  функции  $\Phi(x)$ .

Если  $\Phi(x_1^r) < \Phi(x_2^r)$ , то выполняем присваивания  $a^{r+1} = a^r, b^{r+1} = x_2^r$ ,  $TIN_{r+1} = [a^{r+1}, b^{r+1}]$ . Иначе - выполняем присваивания  $a^{r+1} = x_1^r, b^{r+1} = b^r$ ,  $TIN_{r+1} = [a^{r+1}, b^{r+1}]$ .

Алгоритм Фибоначчи обладает тем свойством, что после выполнения  $(N-1)$ -й итерации имеет место следующая ситуация:  $x_1^{N-1} = x_2^{N-1} = x^{N-1}$ . Т.е. в результате  $(N-1)$ -й итерации

сужение текущего интервала неопределенности не происходит:  
 $\text{ТИН}_{N-1} = [a^{N-1}, b^{N-1}] \Rightarrow \text{ТИН}_{N-2} = [a^{N-2}, b^{N-2}]$ .

Второй этап призван решить по какую сторону от точки  $x^{N-1}$  лежит точка минимума функции  $\Phi(x)$ .

Второй этап выполняется по следующей схеме:

Находим точку  $x^N = x^{N-1} + \delta_x$ , где  $\delta_x \ll |\text{ТИН}_{N-1}|$  - свободный параметр алгоритма.

Вычисляем значение функции  $\Phi(x^N)$ .

Если  $\Phi(x^N) > \Phi(x^{N-1})$ , то выполняем присваивания  $\text{ТИН}_N = [a^{N-1}, x^N]$ . Иначе выполняем присваивания  $\text{ТИН}_N = [x^N, b_N]$ .

В качестве приближенного значения точки минимума  $x^*$  с равными основаниями может быть принята любая точка ТИН.

Некоторые свойства алгоритм Фибоначчи.

Утверждение 1. Для любого  $r \in [1, N-2]$  алгоритм Фибоначчи обладает следующим свойством: одна из точек  $x_1^{r+1}, x_2^{r+1}$  совпадает с одной из точек  $x_1^r, x_2^r$  (см. рис. 1).

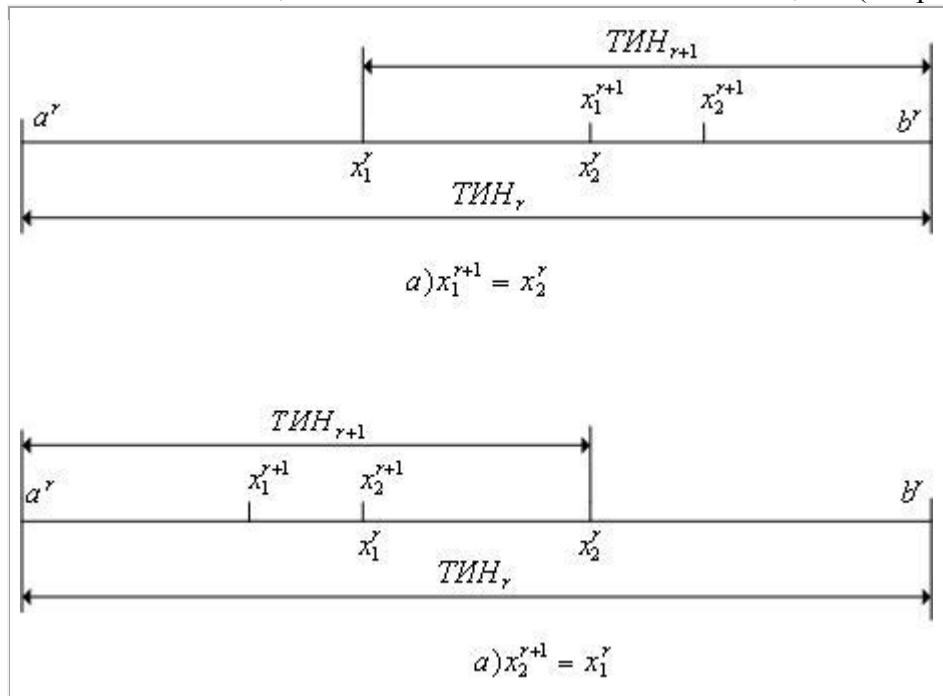


Рис. 1. К утверждению 1.

Доказательство. Пусть на  $r$ -й итерации  $\Phi(x_1^r) \leq \Phi(x_2^r)$  - ситуация б на рис. 1. В соответствии с алгоритмом Фибоначчи  $\text{ТИН}_{r+1} = [a^r, x_2^r]$  причем, очевидно,  $x_1^r \in \text{ТИН}_{r+1}$ . Рассмотрим точку

$$x_2^{r+1} = a^r + |\text{ТИН}_{r+1}| \frac{i_{N-1-r}}{i_{N-r}} = a^r + (x_2^r - a^r) \frac{i_{N-1-r}}{i_{N-r}}.$$

Подставим сюда значение координаты точки

$$x_2^r = a^r + |\text{ТИН}_r| \frac{i_{N-r}}{i_{N+1-r}},$$

$$x_2^{r+1} = a^r + \left( a^r + |\text{ТИН}_r| \frac{i_{N-r}}{i_{N+1-r}} - a^r \right) \frac{i_{N-1-r}}{i_{N-r}} = a^r + |\text{ТИН}_r| \frac{i_{N-1-r}}{i_{N-r}} = x_1^r.$$

Аналогично проводится доказательство для случая  $\Phi(x_1^r) > \Phi(x_2^r)$ . ситуация а на рис. 1 • Указанное свойство алгоритма Фибоначчи позволяет на каждой итерации (кроме первой) производить испытания только в одной точке.

Утверждение 2. Точки  $x_1^r, x_2^r$  расположены симметрично относительно концов текущего интервала неопределенности  $\text{ТИН}_r = [a^r, b^r]$ , т.е. расстояние точки  $x_1^r$  до точки  $a^r$  равно расстоянию точки  $x_2^r$  до точки  $b^r$  или, что то же самое,  $x_2^r - a^r = b^r - x_1^r$ .

Доказательство. В соответствии с алгоритмом Фибоначчи имеем:

$$x_1^r - a^r = a^r + (b^r - a^r) \frac{i_{N-1-r}}{i_{N+1-r}} - a^r = (b^r - a^r) \frac{i_{N-1-r}}{i_{N+1-r}},$$

$$b^r - x_2^r = b^r - a^r - (b^r - a^r) \frac{i_{N-r}}{i_{N+1-r}} - a^r = (b^r - a^r) \frac{i_{N+1-r} - i_{N-r}}{i_{N+1-r}}.$$

Но из формулы (1) следует, что  $i_{N+1-r} = i_{N-r} + i_{N-1-r}$ . Подставляя это в предыдущую формулу,

$$b^r - x_2^r = (b^r - a^r) \frac{i_{N-1-r}}{i_{N+1-r}} = x_1^r - a^r •$$

Утверждение 3. В результате любой итерации  $r \in [1, N-2]$  алгоритма Фибоначчи длина текущего интервала неопределенности уменьшается в  $\frac{i_{N-r}}{i_{N+1-r}}$  раз.

Доказательство. Поскольку  $x_2^r - a^r = b^r - x_1^r$  (см. утверждения 2), достаточно рассмотреть только один из интервалов  $[x_2^r - a^r], [b^r - x_1^r]$ . Рассмотрим первый из указанных интервалов:

$$x_2^r - a^r = a^r + (b^r - a^r) \frac{i_{N-r}}{i_{N+1-r}} - a^r = (b^r - a^r) \frac{i_{N-r}}{i_{N+1-r}} •$$

Утверждение 4. При достаточно больших  $N$  в результате одной итерации алгоритма Фибоначчи длина текущего интервала неопределенности уменьшается примерно в  $\tau$  раз.

Доказательство. Справедливость утверждения следует из утверждения 3 и из того факта, что при достаточно больших  $N$  имеем (см. (2)):

$$\frac{i_{N-r}}{i_{N+1-r}} \approx \frac{(1/\tau)^{N+1-r}}{(1/\tau)^{N+2-r}} = \tau •$$

Из утверждения 4 следует, что при достаточно больших  $N$  алгоритм Фибоначчи практически идентичен алгоритму золотого сечения (см. следующий параграф).

Утверждение 5. В результате  $N$  итераций алгоритма Фибоначчи длина текущего интервала неопределенности становится равной

$$|\text{ТИН}_N| = \frac{b-a}{i_N}. \quad (3)$$

Доказательство. Из утверждения 3 следует, что:

после первой итерации длина ТИН равна  $(b-a) \frac{i_{N-1}}{i_N}$ ;

после второй итерации -  $(b-a) \frac{i_{N-1}}{i_N} \frac{i_{N-2}}{i_{N-1}} = (b-a) \frac{i_{N-2}}{i_N}$ ;

...

после итерации номер  $(N-2)$  длина ТИН равна  $(b-a) \frac{i_2}{i_N} = (b-a) \frac{2}{i_N}$ ;

после итерации номер  $(N-1)$  длина ТИН не меняется;

после итерации номер  $N$  длина ТИН уменьшается в два раза и становится равной  $\frac{b-a}{i_N}$  •  
Поэтому количество итераций  $N$ , необходимых для нахождения минимума функции с точностью  $\varepsilon_x$ , находится из условия

$$\varepsilon_x \leq \frac{b-a}{i_N}.$$

#### 4.4. Алгоритм золотого сечения

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

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

Свойства золотого сечения.

Рассмотрим интервал  $[a, b]$  (см. рис. 1). Говорят, что точка  $c$  выполняет золотое сечение интервала  $[a, b]$ , если

$$\frac{c-a}{b-a} = \tau, \quad (1)$$

где  $\tau = \frac{\sqrt{5}-1}{2} \approx 0,618$  - решение квадратного уравнения

$$\tau^2 + \tau - 1 = 0. \quad (2)$$



Рис. 1. К определению золотого сечения отрезка.

Из определения золотого сечения следует, что  $\frac{b-c}{b-a} = 1-\tau$ . Действительно,  $\frac{b-c}{b-a} = \frac{(b-a)-(c-a)}{b-a} = 1 - \frac{c-a}{b-a} = 1-\tau$ .

Алгоритм золотого сечения.

Алгоритм золотого сечения относится к классу последовательных методов поиска.

Выполняем присваивания  $r=1, a^1=a, b^1=b$ , ТИН<sub>1</sub>=[ $a^1, b^1$ ].

Вычисляем величины (см. Рис. 2)

$$x_1^r = b^r - (b^r - a^r)\tau, \quad x_2^r = a^r + (b^r - a^r)\tau. \quad (3)$$

Вычисляем значения  $\Phi(x_1^r), \Phi(x_2^r)$  функции  $\Phi(x)$ .

Если  $\Phi(x_1^r) < \Phi(x_2^r)$ , то выполняем присваивания  $a^{r+1}=a^r, b^{r+1}=x_2^r$ , ТИН<sub>r+1</sub>=[ $a^{r+1}, b^{r+1}$ ]. Иначе - выполняем присваивания  $a^{r+1}=x_1^r, b^{r+1}=b^r$ , ТИН<sub>r+1</sub>=[ $a^{r+1}, b^{r+1}$ ]

Если  $|ТИН_{r+1}| \leq \varepsilon_x$ , то заканчиваем вычисления. Иначе - выполняем присваивание  $r=r+1$  и переходим на п.2. Здесь  $\varepsilon_x$  – требуемая точность решения •

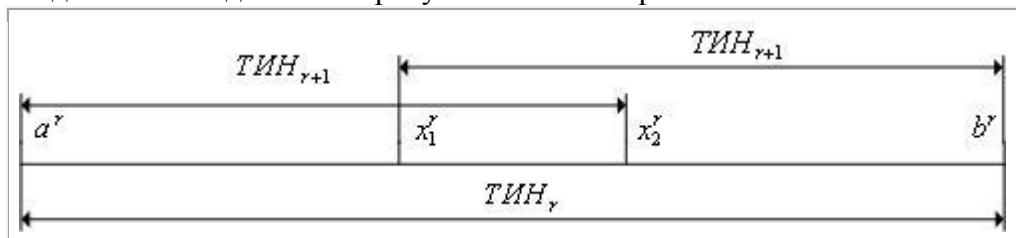


Рис. 2. К определению величин  $x_{1r}, x_{2r}$ .

В качестве приближенного значения точки минимума  $\hat{x}^*$  с равными основаниями может быть принята любая точка последнего текущего интервала неопределенности. Некоторые свойства алгоритм золотого сечения.

Утверждение 1. Точки  $x_1^r, x_2^r$  расположены симметрично относительно концов текущего интервала неопределенности.

Действительно, из (3) следует, что точка  $x_1^r$  отстоит от точки  $b^r$  на величину  $(b^r - a^r)\tau$ ; точка  $x_2^{r+1}$  отстоит от точки  $a^r$  на ту же величину •

Утверждение 2. Для любого  $r \geq 1$  алгоритм золотого сечения обладает следующим свойством: одна из точек  $x_1^{r+1}, x_2^{r+1}$  совпадает с одной из точек  $x_1^r, x_2^r$ .

Доказательство. Пусть на  $r$ -й итерации  $\Phi(x_1^r) < \Phi(x_2^r)$ . В соответствии с алгоритмом золотого сечения  $TIN_{r+1} = [a^r, x_2^r]$  причем, очевидно,  $x_1^r \in TIN_{r+1}$ . Для того, чтобы доказать справедливость утверждения достаточно показать, что верно отношение

$$\frac{x_1^r - a^r}{x_2^r - a^r} = \tau. \quad (4)$$

Из соотношений (3) следует, что

$$(b^r - x_1^r) = (b^r - a^r)\tau \Rightarrow b^r - x_1^r - a^r + a^r = (b^r - a^r)\tau \Rightarrow (b^r - a^r) - (x_1^r - a^r) = (b^r - a^r)\tau \Rightarrow x_1^r - a^r = (b^r - a^r)(1 - \tau).$$

Аналогично имеем  $x_2^r - a^r = (b^r - a^r)\tau$ .

Разделив первый из этих результатов на второй, получим

$$\frac{x_1^r - a^r}{x_2^r - a^r} = \frac{1 - \tau}{\tau}. \quad (5)$$

Из уравнения (2) следует, что  $1 - \tau = \tau^2$ . Отсюда и из (5) следует справедливость (4).

Аналогично проводится доказательство для случая  $\Phi(x_1^r) > \Phi(x_2^r)$  •

Указанное свойство алгоритма золотого сечения позволяет на каждой итерации (кроме первой) производить испытания только в одной точке.

Из схемы алгоритма золотого сечения имеем.

Утверждение 3. В результате одной итерации алгоритма золотого сечения длина текущего интервала неопределенности сокращается в  $\tau$  раз •

Поэтому количество итераций  $N$ , необходимых для нахождения минимума функции с точностью  $\varepsilon_x$ , находится из условия

$$\varepsilon_x \leq (b - a)\tau^N.$$

Из утверждения 3 и результатов предыдущего параграфа следует, что при достаточно больших  $N$  алгоритм Фибоначчи практически идентичен алгоритму золотого сечения.

#### 4.5. Сравнение эффективности алгоритмов одномерной условной оптимизации

Аналитическое вычисление критерия качества алгоритмов оптимизации удаётся только для следующей одномерной задачи условной оптимизации: найти минимум одномерной унимодальной функции  $\Phi(x)$ , определенной в замкнутой области допустимых значений  $D = [a, b]$ ,

$$\min_{x \in [a, b]} \Phi(x) = \Phi(x^*). \quad (1)$$

Обозначим через  $\Psi$  класс одномерных унимодальных функций. Пусть множество рассматриваемых алгоритмов решения задачи (1) есть  $\{A_1, A_2, A_3, A_4\}$ , где

$A_1$  - алгоритм равномерного поиска,

$A_2$  - алгоритм равномерного дихотомического поиска,

$A_3$  - алгоритм Фибоначчи,

$A_4$  - алгоритм золотого сечения.

В качестве критерия оптимальности алгоритмов  $A \in \{A_1, A_2, A_3, A_4\}$  на классе функций  $\Psi$  используем максимальную длину текущего интервала неопределенности после  $N$  испытаний

$$W(A) = \max_{\Phi \in \Psi} |\text{ТИН}_N(\Phi, A)|.$$

В параграфе 1 мы показали, что если количество узлов сетки равно  $N+1$ , то после одной итерации алгоритма равномерного поиска текущий интервал неопределенности уменьшается в  $\frac{N}{2}$  раз. Поэтому

$$W(A_1) = \frac{2(b-a)}{N-1}.$$

Из результатов параграфа 2 следует, что после одной итерации (двух испытаний) алгоритма равномерного поиска текущий интервал неопределенности уменьшается в 2 раза. Поэтому

$$W(A_2) = \frac{(b-a)}{2^{N/2}}.$$

В параграфе 3 показано, что в результате  $N$  итераций ( $N+1$  испытаний) алгоритма Фибоначчи длина текущего интервала неопределенности становится равной  $\frac{b-a}{\tau^N}$ . Отсюда следует, что

$$W(A_3) = \frac{b-a}{\tau^{N-1}}.$$

Наконец, в параграфе 4 показано, что в результате  $N$  итераций ( $N+1$  испытаний) алгоритма золотого сечения длина текущего интервала неопределенности становится равной  $(b-a)^{\tau^N}$ . Поэтому

$$W(A_4) = (b-a)\tau^{N-1}.$$

Сравним эффективности алгоритма деления пополам и алгоритма Фибоначчи при  $N=14$ :

$$\frac{W(A_2)}{W(A_3)} = \frac{\frac{(b-a)}{2^{N/2}}}{\frac{b-a}{i_{N-1}}} = \frac{i_{N-1}}{2^{N/2}} = \frac{377}{128} \approx 3.$$

Таким образом, при  $N=14$  алгоритм Фибоначчи почти в 3 раз эффективнее алгоритма деления пополам.

При  $N=14$  сравним также эффективности алгоритма Фибоначчи и алгоритма золотого сечения:

$$\frac{W(A_3)}{W(A_4)} = \frac{1}{i_{N-1}\tau^{N-1}} \approx \frac{\tau^{N\sqrt{5}}}{\tau^{N-1}} = \tau^N \sqrt{5} \approx 1.4.$$

$$i_N \approx \frac{(1/\tau)^{N+1}}{\sqrt{5}}.$$

Здесь учтено, что Таким образом, при  $N=14$  алгоритм золотого сечения примерно на 40 процентов эффективнее алгоритма Фибоначчи.

#### 4.6. Метод квадратичной аппроксимации

Метод квадратичной аппроксимации относится к семейству методов полиномиальной аппроксимации. Идея метода полиномиальной аппроксимации состоит в том, что в некоторой окрестности минимума функции  $\Phi(x)$  она аппроксимируется полиномом достаточно высокого порядка и в качестве точки минимума функции  $\Phi(x)$  (или в качестве очередного приближения к этой точке) принимается точка минимума аппроксимирующего полинома. В силу того, что аппроксимирующая функция является полиномом, этот минимум находится легко.

В качестве аппроксимирующих полиномов чаще всего используются полиномы второго и третьего порядков, т.е. квадратичная и кубическая аппроксимации.

Квадратичная аппроксимация.

Рассмотрим квадратичную аппроксимацию (см. рис. 1). Пусть в процессе решения задачи поиска минимума функции  $\Phi(x)$  тем или иным образом получены попарно не совпадающие точки  $x_1, x_2, x_3$ , принадлежащие области допустимых значений  $D$  (не обязательно упорядоченные слева направо!).

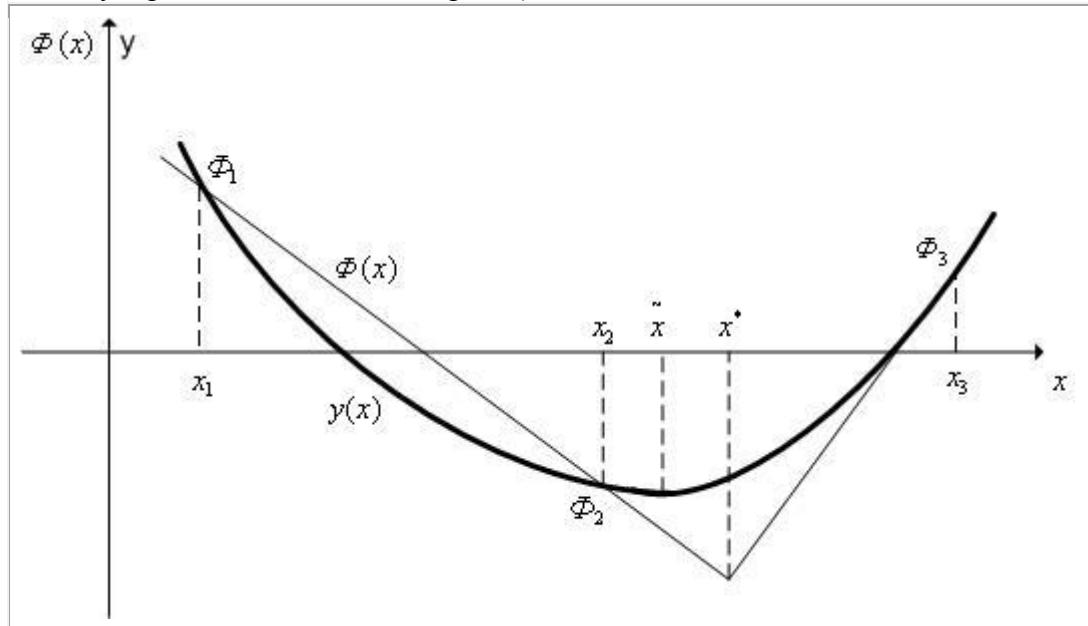


Рис. 1. Квадратичная аппроксимация.

Построим квадратичную функцию

$$y = \alpha x^2 + \beta x + \gamma, \quad (1)$$

проходящую через точки  $(x_i, \Phi_i)$ ,  $i=1,2,3$ , где  $\Phi_i = \Phi(x_i)$ .

Коэффициенты  $\alpha, \beta, \gamma$  функции (1) удовлетворяют системе линейных алгебраических уравнений (СЛАУ)

$$\begin{cases} \alpha x_1^2 + \beta x_1 + \gamma = \Phi_1 \\ \alpha x_2^2 + \beta x_2 + \gamma = \Phi_2 \\ \alpha x_3^2 + \beta x_3 + \gamma = \Phi_3 \end{cases} \quad (2)$$

Определитель СЛАУ (2) является определителем Вандермонда, который отличен от нуля, если величины  $x_1, x_2, x_3$  попарно различны.

Таким образом, в сделанных предположениях СЛАУ (2) имеет решение и, притом единственное. Поскольку определитель СЛАУ (2) равен  $(x_1-x_2)(x_1-x_3)(x_2-x_3)$ , это решение имеет вид

$$\begin{cases} \alpha = \frac{\Phi_1}{r_{12}r_{13}} - \frac{\Phi_2}{r_{12}r_{23}} + \frac{\Phi_3}{r_{13}r_{23}} \\ \beta = \frac{\Phi_1 s_{23}}{r_{12}r_{13}} + \frac{\Phi_2 s_{13}}{r_{12}r_{23}} - \frac{\Phi_3 s_{12}}{r_{13}r_{23}} \\ \gamma = \frac{\Phi_1 p_{23}}{r_{12}r_{13}} - \frac{\Phi_2 p_{13}}{r_{12}r_{23}} + \frac{\Phi_3 p_{12}}{r_{13}r_{23}} \end{cases},$$

где  $r_{ij}=x_i-x_j$ ,  $s_{ij}=x_i+x_j$ ,  $p_{ij}=x_i x_j$ .

Подставим найденные значения коэффициентов  $\alpha, \beta, \gamma$  в необходимое условие  $y=2\alpha x+\beta=0$  минимума квадратичной функции (1), получим стационарную точку этой функции

$$\tilde{x} = -\frac{\beta}{2\alpha} = \frac{\Phi_1 e_{23} - \Phi_2 e_{13} + \Phi_3 e_{12}}{2(\Phi_1 r_{23} - \Phi_2 r_{13} + \Phi_3 r_{12})}, \quad (3)$$

где  $e_{ij}=x_i^2-x_j^2$ .

Метод квадратичной аппроксимации.

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

$$\min_{x \in [a, b]} \Phi(x) = \Phi(x^*). \quad (4)$$

Метод квадратичной аппроксимации относится к классу методов сокращения текущего интервала неопределенности. Пусть при решении задачи (4) каким-либо методом получены три точки  $x_1 < x_2 < x_3$ , принадлежащие области допустимых значений, такие, что  $\Phi(x_1) > \Phi(x_2) < \Phi(x_3)$ .

Схема метода квадратичной аппроксимации:

Выполняем присваивания  $r=1, x_1^r=x_1, x_2^r=x_2, x_3^r=x_3$ , ТИН $_r=[x_1^r, x_3^r]$ .

Вычисляем значения  $\Phi_1^r, \Phi_2^r, \Phi_3^r$  функции  $\Phi(x)$  в точках  $x_1^r, x_2^r, x_3^r$ , соответственно.

По формуле (3) вычисляем величину  $\tilde{x}^r$  и находим значение функции  $\Phi(x)$  в этой точке  $\tilde{\Phi}^r = \Phi(\tilde{x}^r)$ .

Находим следующие три точки:

случай (а) - если  $\tilde{x}^r \in [x_2^r, x_3^r]$ , то  $x_1^{r+1}=x_2^r, x_2^{r+1}=\tilde{x}^r, x_3^{r+1}=x_3^r$  (см. рис. 2);

случай (б) - если  $\tilde{x}^r \in [x_1^r, x_2^r]$ , то  $x_1^{r+1}=x_1^r, x_2^{r+1}=\tilde{x}^r, x_3^{r+1}=x_2^r$  (см. рис. 3).

В качестве следующего текущего интервала неопределенности принимаем  $TIN_{r+1} = [x_1^{r+1}, x_3^{r+1}]$ .

Если  $|TIN_{r+1}| \leq \varepsilon_x$ , то заканчиваем вычисления. Иначе - выполняем присваивание  $r = r + 1$  и переходим на п.2. Здесь  $\varepsilon_x$  – требуемая точность решения •

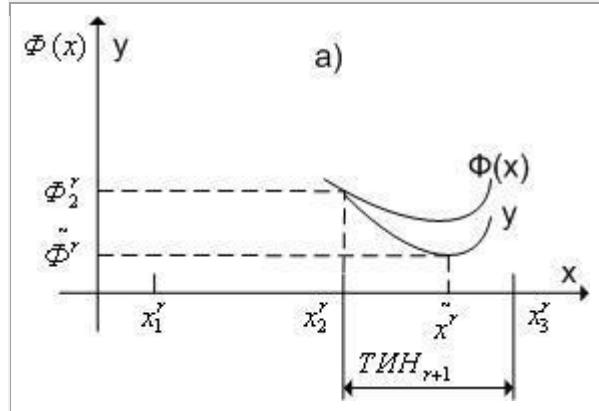


Рис. 2. К методу квадратичной аппроксимации (случай а).

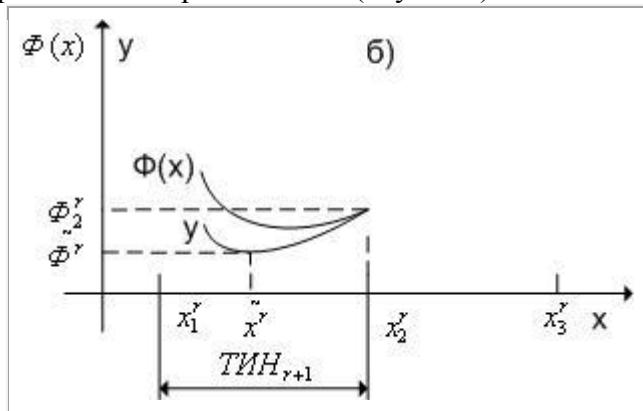


Рис. 3. К методу квадратичной аппроксимации (случай б).

В качестве приближенного значения точки минимума  $\tilde{x}^*$  с равными основаниями может быть принята любая точка последнего текущего интервала неопределенности.

Замечание. В силу условий  $x_1^r < x_2^r < x_3^r$ ,  $\Phi_1^r > \Phi_2^r < \Phi_3^r$  точка  $\tilde{x}^r$  всегда принадлежит текущему интервалу неопределенности  $TIN_r = [x_1^r, x_3^r]$  •

#### 4.7. Метод Паулла

Напомним, что методы, использующие квадратичную аппроксимацию, относятся к семейству методов полиномиальной аппроксимации.

В параграфе 6 показано, что если имеются попарно не совпадающие точки  $x_1, x_2, x_3$  и соответствующие значения  $\Phi_1, \Phi_2, \Phi_3$  минимизируемой функции, то аппроксимирующий квадратичный полином  $y = \alpha x^2 + \beta x + \gamma$  достигает минимального значения в точке

$$\tilde{x} = -\frac{\beta}{2\alpha} = \frac{\Phi_1 e_{23} - \Phi_2 e_{13} + \Phi_3 e_{12}}{2(\Phi_1 r_{23} - \Phi_2 r_{13} + \Phi_3 r_{12})}, \quad (1)$$

где  $r_{ij} = x_i - x_j$ ,  $e_{ij} = x_i^2 - x_j^2$ .

Рассмотрим следующую задачу безусловной оптимизации: найти минимум одномерной унимодальной функции  $\Phi(x)$ , определенной в незамкнутой области допустимых значений  $D = (-\infty, \infty)$ ,

$$\min_{x \in [-\infty, \infty]} \Phi(x) = \Phi(x^*). \quad (2)$$

Метод Паулла решения задачи (2) может быть весьма полезен при решении задач безусловной оптимизации многопараметрических функций (см. далее).

Схема метода Паулла:

Полагаем  $r=1$  и задаем начальную точку  $x_1^r$ .

Находим точку  $x_2^r = x_1^r + \Delta$ , где  $\Delta$  – длина шага, которая должна быть величиной того же порядка, что и расстояние точки  $x_1^r$  до точки минимума функции  $\Phi(x)$  (оценка этого расстояния является самостоятельной проблемой).

Вычисляем значения  $\Phi_1^r, \Phi_2^r$  функции  $\Phi(x)$  в точках  $x_1^r, x_2^r$ .

Находим точку  $x_3^r$ :

$$x_3^r = \begin{cases} x_1^r + 2\Delta, & \Phi_1^r > \Phi_2^r \\ x_1^r - \Delta, & \Phi_1^r < \Phi_2^r \end{cases}$$

Вычисляем значение  $\Phi_3^r$  функции  $\Phi(x)$  в точке  $x_3^r$  (см. рис. 1).

По формуле (1) вычисляем величину  $\tilde{x}^r$  и находим значение функции  $\Phi(x)$  в этой точке  $\tilde{\Phi}^r = \Phi(\tilde{x}^r)$ .

Находим следующие три точки:

- а) если  $\tilde{x}^r \in [x_2^r, x_3^r]$ , то  $x_1^{r+1} = x_2^r, x_2^{r+1} = \tilde{x}^r, x_3^{r+1} = x_3^r$ ;
- б) если  $\tilde{x}^r \in [x_1^r, x_2^r]$ , то  $x_1^{r+1} = x_1^r, x_2^{r+1} = \tilde{x}^r, x_3^{r+1} = x_2^r$ ;
- в) если  $\tilde{x}^r > x_3^r$ , то  $x_1^{r+1} = x_2^r, x_2^{r+1} = x_3^r, x_3^{r+1} = \tilde{x}^r$  (см. рис. 2);
- г) если  $\tilde{x}^r < x_1^r$ , то  $x_1^{r+1} = \tilde{x}^r, x_2^{r+1} = x_3^r, x_3^{r+1} = x_1^r$  (см. рис. 3).

В качестве следующего текущего интервала неопределенности принимаем  $TIN_{r+1} = [x_1^{r+1}, x_3^{r+1}]$ .

Если  $|TIN_{r+1}| \leq \varepsilon_x$ , то заканчиваем вычисления. Иначе - выполняем присваивание  $r = r + 1$  и переходим на п.6. Здесь  $\varepsilon_x$  – требуемая точность решения •

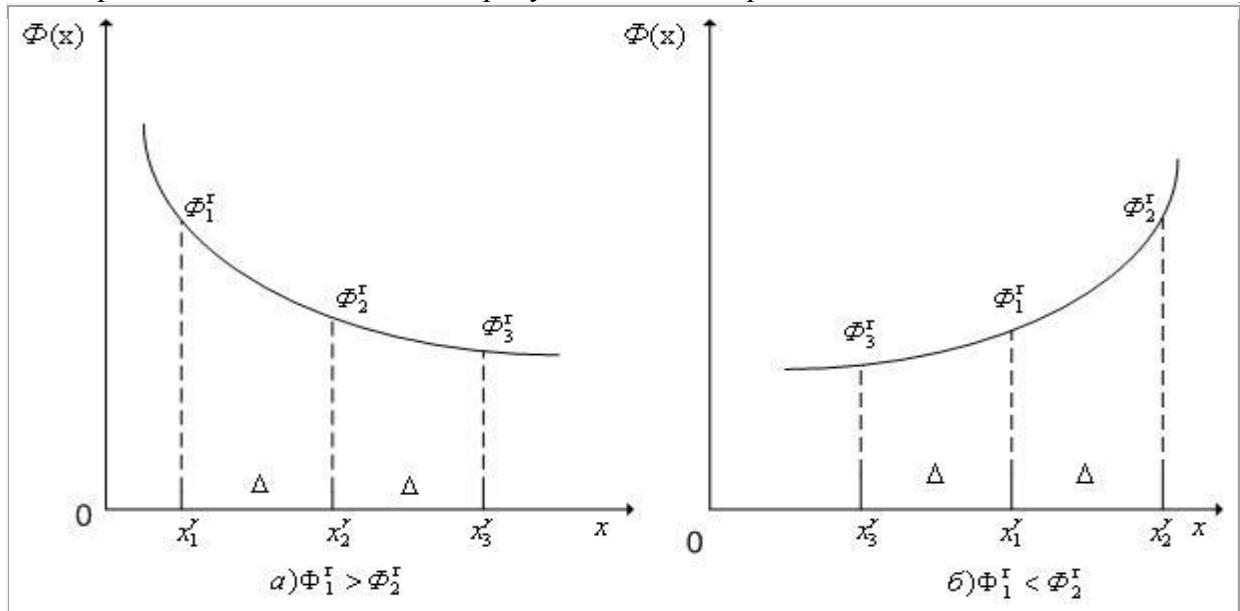


Рис. 1. К определению точки  $x_{3r}$ .

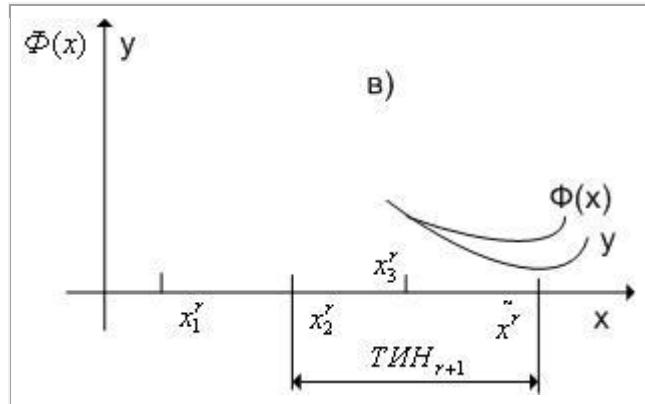


Рис. 2. Случай  $x \sim r > x_{3r}$ .

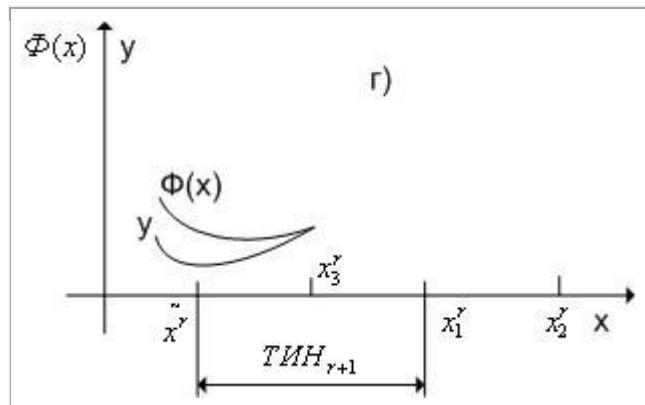


Рис. 3. Случай  $x \sim r < x_{3r}$

В качестве приближенного значения точки минимума  $\tilde{x}^*$  с равными основаниями может быть принята любая точка последнего текущего интервала неопределенности.

Рассмотренный метод Паулла может быть модифицирован и для замкнутой области допустимых значений  $D$ :

#### 4.8. Методы на основе поиска стационарной точки критерия оптимальности

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

$$\min_{x \in [a, b]} \Phi(x) = \Phi(x^*). \quad (1)$$

В параграфе 2.1 мы показали, что в этих предположениях необходимым условием минимума функции  $\Phi(x)$  является условие

$$\Phi'(x) = 0, x \in [a, b]. \quad (2)$$

Рассматриваемые в данном параграфе методы первого порядка решения задачи (1) основаны на поиске стационарной точки функции  $\Phi(x)$ , т.е. на решении задачи (2), которая представляет собой задачу нахождения корней функции  $\Phi'(x)$ , принадлежащих интервалу  $[a, b]$ .

Аналитическое решение задачи (2) возможно лишь в простейших случаях. Обычно для решения этой задачи приходится использовать численные методы нахождения корней нелинейных уравнений.

Широко известны следующие методы нахождения корней нелинейных уравнений:  
метод хорд (метод секущих);

метод касательных (метод Ньютона решения нелинейных уравнений) •

Поиск стационарной точки минимизируемой функции методом хорд.

Метод хорд ориентирован на нахождение корня уравнения (2) в случае, когда на границах интервала  $[a, b]$  знаки производной  $\Phi'(x)$  различны. Такая ситуация, очевидно, возможна, если точка минимума функции  $\Phi(x)$  является внутренней точкой интервала  $[a, b]$  — см. рис. 1.

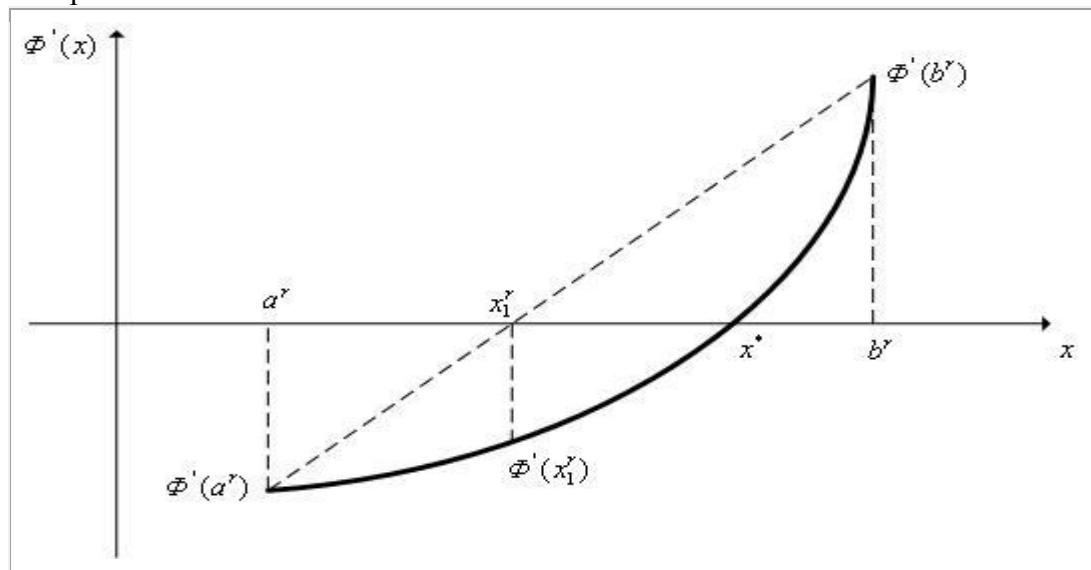


Рис. 1. К схеме метода хорд.

Нам далее понадобится значение  $x_1^*$ . Из подобия треугольника  $a^*, x_1^*, \Phi'(a^*)$  и треугольника  $b^*, x_1^*, \Phi'(b^*)$  имеем

$$\frac{\Phi'(b^r)}{\Phi'(a^r)} = \frac{b^r - x_1^r}{x_1^r - a^r}.$$

Отсюда следует, что

$$x_1^r = b^r - \frac{\Phi'(b^r)(b^r - a^r)}{\Phi'(b^r) - \Phi'(a^r)}. \quad (3)$$

Схема поиска стационарной точки минимизируемой функции методом хорд (см. рис. 1):

Выполняем присваивания  $r=1, a^1=a, b^1=b$ .

Вычисляем значения производных  $\Phi'(a^r), \Phi'(b^r)$ .

Если производные  $\Phi'(a^r), \Phi'(b^r)$  имеют одинаковые знаки – завершаем вычисления (точки  $a, b$  выбраны неверно).

По формуле (3) вычисляем приближение  $x_1^r$  к стационарной точке функции  $\Phi(x)$  в значение производной  $\Phi'(x_1^r)$ .

Если  $|\Phi'(x_1^r)| \leq \varepsilon$ , где  $\varepsilon$  – требуемая точность решения, то принимаем  $x^* \approx x_1^r$  и заканчиваем вычисления.

Если производные  $\Phi'(a^r), \Phi'(x_1^r)$  имеют разные знаки, то выполняем присваивания  $a^{r+1}=a^r, b^{r+1}=x_1^r, r=r+1$  и переходим на п.4.

Если производные  $\Phi'(x_1^r), \Phi'(b^r)$  имеют разные знаки (как на рис. 1), то выполняем присваивания  $a^{r+1}=x_1^r, b^{r+1}=b^r, r=r+1$  и переходим на п.4 •

В случае квадратичной функции  $\Phi(x)$  производная этой функции  $\Phi'(x)$  линейна. Поэтому метод хорд гарантирует нахождение стационарной точки функции  $\Phi(x)$  всего за одну итерацию.

Поскольку поиск заканчивается при выполнении условия  $|\Phi'(x_1^r)| \leq \varepsilon$ , возможно появление ложных корней. Например, для уравнения  $x^2 + 0.0001 = 0$  ложный корень  $x=0$  появляется в том случае, если  $\varepsilon > 0.0001$ . В подобных случаях увеличивая точность поиска, можно избавить от ложных корней. Однако, возможны уравнения, для которых такой подход не приводит к успеху. Например, уравнение  $\frac{1}{x} = 0$  не имеет действительных корней, однако для сколь угодно малого  $\varepsilon$  найдется точка, удовлетворяющая условию окончания поиска.

Возможна модификация метода хорд, когда значения производной  $\Phi'(x)$  вычисляются приближенно с использованием первых разностей. В этом случае, очевидно, метод становится прямым (нулевого порядка).

Поиск стационарной точки минимизируемой функции методом касательных.

Метод касательных ориентирован на нахождение корня уравнения (2) в случае, когда на границах интервала  $[a, b]$  знаки производной  $\Phi'(x)$  различны. Такая ситуация, очевидно, возможна, если точка минимума функции  $\Phi(x)$  является внутренней точкой интервала  $[a, b]$ .

, $b]$  — см. рис. 2. метод требует, чтобы функция  $\Phi(x)$  была определена и дважды дифференцируема в области допустимых значений  $D=[a,b]$ .

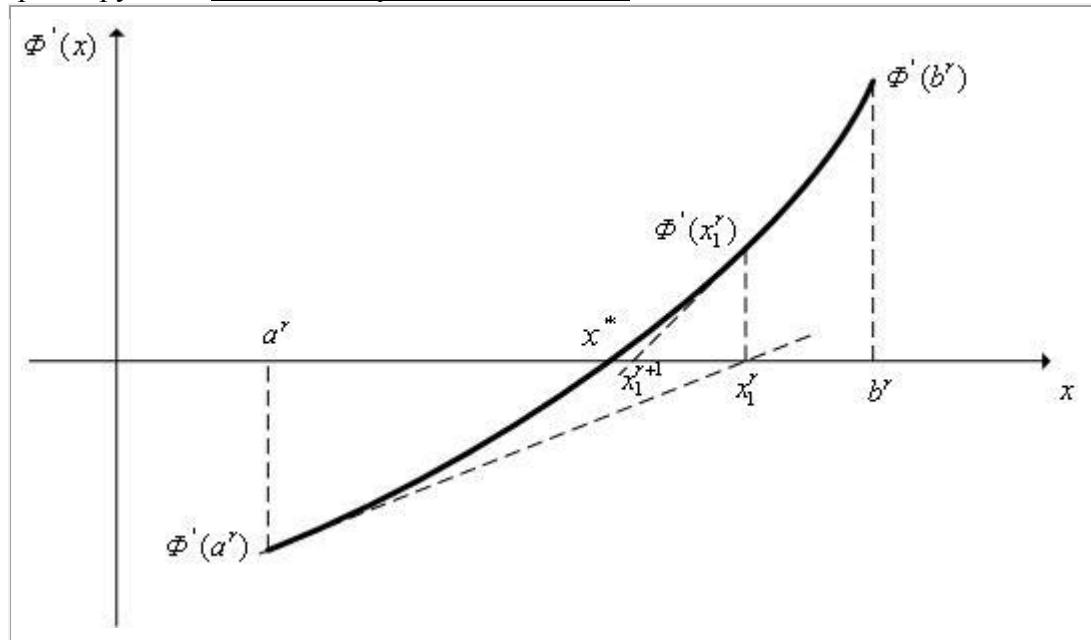


Рис. 2. К схеме метода касательных.

Нам далее понадобится значение  $x_1^r$ . Линейная функция, аппроксимирующая функцию  $\Phi'(x)$  в точке  $a^r$ , записывается в виде

$$\tilde{\Phi}'(x, a^r) = \Phi'(a^r) + \Phi''(a^r)(x - a^r). \quad (4)$$

Приравняв правую часть уравнения (4) к нулю, получим

$$x_1^r = a^r - \frac{\Phi'(a^r)}{\Phi''(a^r)}. \quad (5)$$

Схема поиска стационарной точки минимизируемой функции методом касательных (см. рис. 2):

Выполняем присваивания  $r=1$ ,  $a^1=a$ ,  $b^1=b$ .

Вычисляем значения производных  $\Phi'(a^r)$ ,  $\Phi'(b^r)$ .

Если производные  $\Phi'(a^r)$ ,  $\Phi'(b^r)$  имеют одинаковые знаки — завершаем вычисления (точки  $a, b$  выбраны неверно).

По формуле (5) вычисляем приближение  $x_1^r$  к стационарной точке функции  $\Phi(x)$  и значение  $\Phi(x_1^r)$ .

Если  $|\Phi'(x_1^r)| \leq \varepsilon$ , где  $\varepsilon$  — требуемая точность решения, то принимаем  $x^* \approx x_1^r$  и заканчиваем вычисления.

Если разные знаки имеют производные  $\Phi'(a^r)$ ,  $\Phi'(x_1^r)$  (как на рис. 2), то выполняем присваивания  $a^{r+1}=a^r$ ,  $b^{r+1}=x_1^r$ ,  $r=r+1$  и переходим на п.4.

Если разные знаки имеют производные  $\Phi'(x_1^r)$ ,  $\Phi'(b^r)$ , то выполняем присваивания  $a^{r+1}=x_1^r$ ,  $b^{r+1}=b^r$ ,  $r=r+1$  и переходим на п.4. •

В случае квадратичной функции  $\Phi(x)$  производная этой функции  $\Phi'(x)$  линейна. Поэтому метод касательных гарантирует нахождение стационарной точки функции  $\Phi(x)$  всего за одну итерацию.

Также, как в методе хорд, возможна модификация метода касательных, когда значения производной  $\Phi'(x)$  вычисляются приближенно с использованием первых разностей. В этом случае, очевидно, метод становится прямым (нулевого порядка).

#### 4.9. Повышение эффективности поиска на основе дополнительной информации о свойствах критерия оптимальности

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

$$\min_{x \in [a, b]} \Phi(x) = \Phi(x^*). \quad (1)$$

Всякая априорная информация о свойствах минимизируемой функции  $\Phi(x)$  может быть использована для повышения эффективности решения задачи (1).

Положим, что имеется следующая априорная информация о минимизируемой функции:  $\Phi(x)$  является липшицевой функцией, т.е. принадлежит классу функций, которые на интервале  $[a, b]$  удовлетворяют условию Липшица с константой Липшица  $K$

$$|\Phi(x_1) - \Phi(x_2)| \leq K|x_1 - x_2|, \forall x_1, x_2 \in [a, b]$$

Покажем, как можно использовать данную априорную информацию для сокращения текущего интервала неопределенности без проведения дополнительных испытаний.

Пусть после проведения  $r$  итераций каким-либо методом сокращения текущего интервала неопределенности имеет место ситуация:

$$a^r = x_1^r < x_2^r < b^r = x_3^r, \Phi(a^r) > \Phi(x_1^r) < \Phi(b^r), \text{ТИН}_r = [a^r, b^r].$$

Обозначим  $\Phi(x_j^r) = \Phi_j^r, j=1, 2, 3$  и проведем через точки  $(x_j^r, \Phi_j^r), j=1, 3$  прямые  $L_1, L_3$  с тангенсами угла наклона к оси  $x$ , равными  $K$  и  $(-K)$ , соответственно (см. рис. 1).

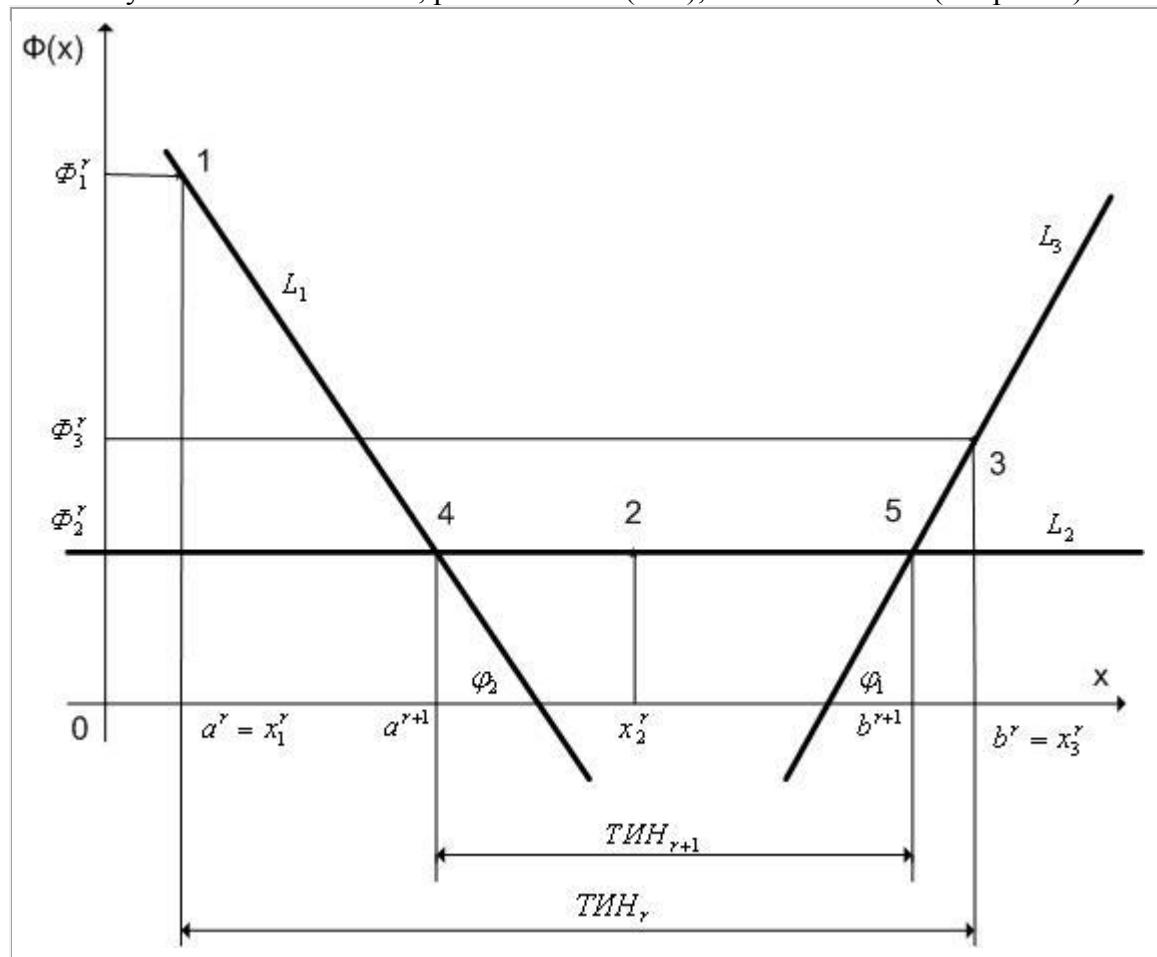


Рис. 1. Сужение интервала неопределенности за счет априорной информации о липшицевости минимизируемой функции.  $\operatorname{tg}(\varphi_1)=K$ ,  $\operatorname{tg}(\varphi_2)=-K$ ,  $\varphi_2=180^\circ-\varphi_1$ .

Проведем, кроме того, через точку  $(x_2^r, \Phi_2^r)$  прямую  $L_2$ , параллельную оси  $Ox$ , до пересечения с прямыми  $L_1, L_3$  и обозначим абсциссы точек пересечения  $a^{r+1}, b^{r+1}$ .

Утверждение 1. В сделанных предположениях, точки  $a^{r+1}, b^{r+1}$  могут быть использованы в качестве границ интервала неопределенности  $\text{ТИН}_{r+1} \in \text{ТИН}_r$ . Другими словами, в сделанных предположениях, точка  $x^*$  минимума функции  $\Phi(x)$  не может лежать вне интервала  $[a^{r+1}, b^{r+1}] \in [a^r, b^r]$ .

Найдем величины  $a^{r+1}, b^{r+1}$ .

Прямая  $L_1$  имеет уравнение  $y_1 = -Kx + c$ , где константа  $c$  определяется из условия прохождения этой прямой через точку  $(x_1^r, \Phi_1^r)$

$$\Phi_1^r = -Kx_1^r + c \Rightarrow c = \Phi_1^r + Kx_1^r.$$

Таким образом,  $y_1 = -Kx + \Phi_1^r + Kx_1^r$ .

В точке (4) имеет место равенство  $(-K)a^{r+1} + \Phi_1^r + Kx_1^r = \Phi_2^r$ , из которого следует, что

$$a^{r+1} = x_1^r + \frac{\Phi_1^r - \Phi_2^r}{K}. \quad \text{Аналогично для прямой } L_2 \quad b^{r+1} = x_3^r - \frac{\Phi_3^r - \Phi_2^r}{K}.$$

## 5.1. Метод перебора. Одномерный метод Монте-Карло

Некоторые методы решения многомерных задач оптимизации требуют решения одномерной задачи глобальной оптимизации (или совокупности таких задач).

Рассматривается следующая одномерная задача условной глобальной оптимизации: найти минимум, вообще говоря, многоэкстремальной функции  $\Phi(x)$ , определенной в замкнутой области допустимых значений  $D=[a,b]$ ,

$$\min_{x \in [a,b]} \Phi(x) = \Phi(x^*). \quad (1)$$

Метод перебора

Схема метода перебора (см. рис. 1):

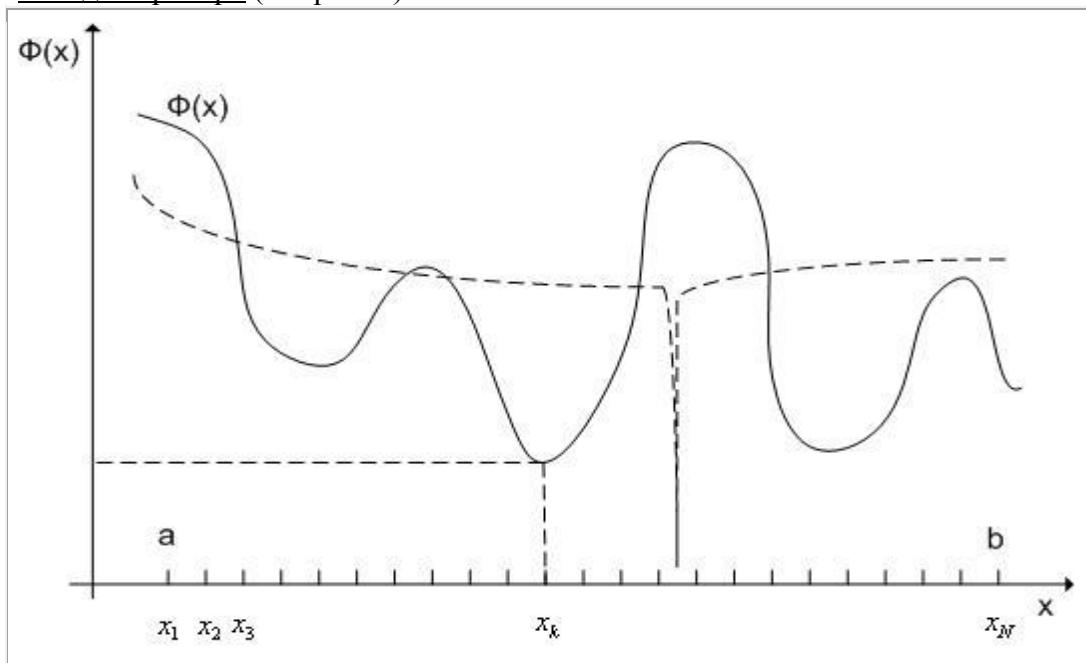


Рис. 1. К схеме метода перебора.

Покрываем интервал  $[a,b]$  некоторой сеткой с узлами  $x_i, i \in [1, N]$ .

Производим испытание в точке  $x_1 = \tilde{x}^*$ , т.е. вычисляем значения  $\tilde{\Phi}^*$  функции  $\Phi(x)$  в этой точке.

Полагаем  $r=2$ .

Производим испытание в точке  $x_r$  - вычисляем значение  $\Phi_r = \tilde{\Phi}(x_r)$  функции  $\Phi(x)$  в этой точке.

Если  $\Phi_r < \tilde{\Phi}^*$ , то выполняем присваивания  $\tilde{x}^* = x_r$ ,  $\tilde{\Phi}^* = \Phi_r$ .

Если  $r < N$ , то выполняем присваивание  $r=r+1$  и переходим на п.4). Иначе - заканчиваем вычисления.

Принимаем  $\tilde{x}^*$  в качестве приближенного значения точки глобального минимума функции  $\Phi(x)$  на интервале  $[a,b]$  или каким-либо из рассмотренных одномерных методов локальной оптимизации организуем в окрестности точки  $\tilde{x}^*$  поиск локального минимума этой функции •

При выборе количества узлов сетки  $x_i, i \in [1, N]$  можно исходить из требуемой точности решения  $\varepsilon_x$  – максимальный шаг сетки принять равным этой величине. Отметим, что метод перебора, как и любой другой метод глобальной оптимизации, при отсутствии априорной информации о свойствах минимизируемой функции не гарантирует нахождение глобального минимума (см. пунктирный график на рис. 1).

Одномерный метод Монте-Карло

Схема одномерного метода Монте-Карло:

Генерируем с помощью какого-либо программного генератора случайных чисел, равномерно распределенных в интервале  $[a, b]$ , случайное число  $x_1 = \tilde{x}^*$ .

Производим испытание в точке  $x_1 = \tilde{x}^*$  - вычисляем значения  $\tilde{\Phi}^*$  функции  $\Phi(x)$  в этой точке.

Полагаем  $r=2$ .

Аналогично п. 1) генерируем случайное число  $x_r \in [a, b]$ .

Производим испытание в точке  $x_r$  - вычисляем значение  $\Phi_r = \Phi(x_r)$  функции  $\Phi(x)$  в этой точке.

Если  $\Phi_r < \tilde{\Phi}^*$ , то выполняем присваивания  $\tilde{x}^* = x_r$ ,  $\tilde{\Phi}^* = \Phi_r$ .

Если  $r < N$ , то выполняем присваивание  $r=r+1$  и переходим на п. 4). Иначе - заканчиваем вычисления. Здесь  $N$  – количество испытаний.

Принимаем  $\tilde{x}^*$  в качестве приближенного значения точки глобального минимума функции  $\Phi(x)$  на интервале  $[a, b]$  или каким-либо из рассмотренных одномерных методов локальной оптимизации организуем в окрестности точки  $\tilde{x}^*$  поиск локального минимума этой функции •

При достаточно большом  $N$  метод гарантирует нахождение глобального минимума с высокой вероятностью.

## 5.2. Метод выделения интервалов унимодальности

Рассмотрим одномерную задачу условной глобальной оптимизации: найти минимум одномерной многоэкстремальной функции  $\Phi(x)$ , определенной в замкнутой области допустимых значений  $D=[a,b]$  и имеющей в этой области конечное число минимумов,

$$\min_{x \in [a,b]} \Phi(x) = \Phi(x^*). \quad (1)$$

Метод выделения интервалов унимодальности функции  $\Phi(x)$  требует априорного знания оценки  $d > 0$  минимального расстояния между локальными минимумами этой функции. Схема метода (см. рис. 1):

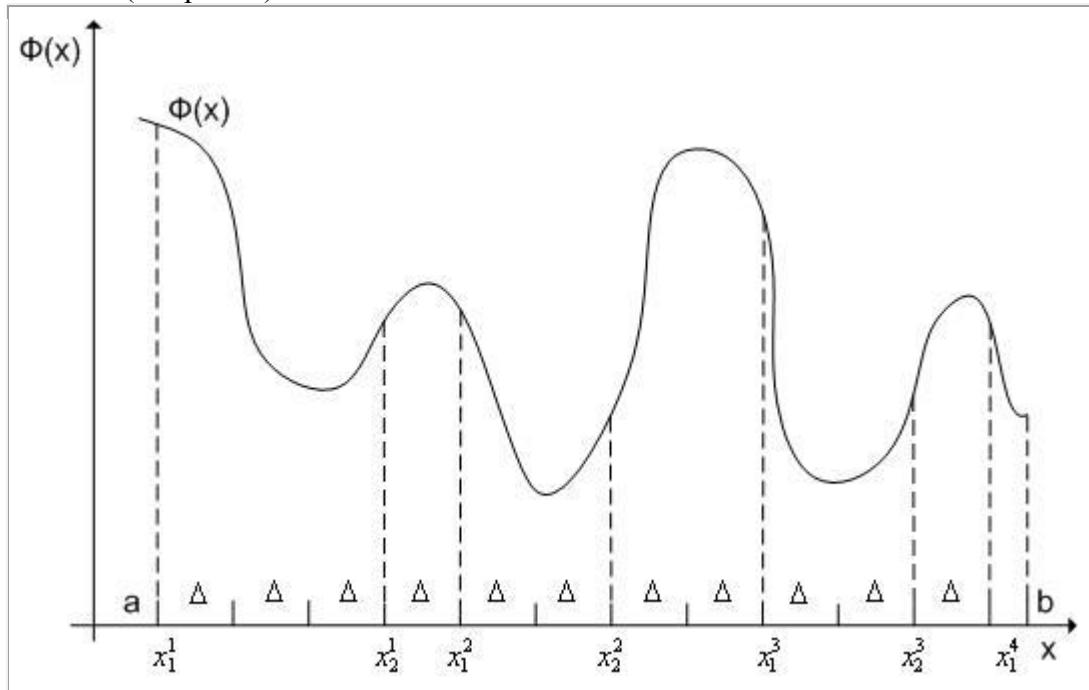


Рис. 1. К схеме метода выделения интервалов унимодальности.

Полагаем  $r=1$ .

Если функция  $\Phi(x)$  в точке  $a$  возрастает, то полагаем  $x_0^r=a$  и переходим на п. 4).

Если функция  $\Phi(x)$  в точке  $a$  убывает, то полагаем  $x_1^r=a$  и переходим на п. 5).

Последовательно увеличивая  $x$  на величину  $\Delta$  по формуле  $x=x_0^r+i\Delta \leq b$ ,  $i=1,2,\dots$ , находим первую точку  $x_1^r$ , в которой функция  $\Phi(x)$  убывает. Здесь и далее  $\Delta$  - величина в несколько раз меньшая величины  $d$ .

Последовательно увеличивая  $x$  на величину  $\Delta$  по формуле  $x=x_1^r+i\Delta \leq b$ ,  $i=1,2,\dots$ , находим первую точку  $x_2^r$ , в которой функция  $\Phi(x)$  возрастает.

В качестве  $r$ -го интервала унимодальности принимаем интервал  $[x_1^r, x_2^r]$ .

Если интервал  $[a, b]$  исчерпан, переходим на п.8), иначе полагаем  $x_0^{r+1}=x_2^r$ ,  $r=r+1$  и переходим на п.4).

Положим, что общее количество найденных интервалов унимодальности функции  $\Phi(x)$  равно  $N-1$ . Каким-либо одномерным методом локальной оптимизации находим локальный минимум функции  $\Phi(x)$  в каждом из  $N-1$  интервалов унимодальности.

Обозначим точки этих минимумов  $\tilde{x}_i^*, i \in [1, N-1]$ . Соответствующие значения функции  $\Phi(x)$  обозначим  $\tilde{\Phi}_i^*, i \in [1, N-1]$ . Добавим к точкам  $\tilde{x}_i^*$  точки  $\tilde{x}_0^*, \tilde{x}_N^*$  и вычислим соответствующие значения  $\tilde{\Phi}_0^*, \tilde{\Phi}_N^*$  функции  $\Phi(x)$ .

Найдем минимальную из величин  $\tilde{\Phi}_i^*, i \in [0, N]$  и соответствующее значение аргумента:

$$\min_{i \in [0, N]} \tilde{\Phi}_i^* = \tilde{\Phi}_k^* = \Phi(\tilde{x}_k^*)$$

В качестве решения задачи глобальной оптимизации (1) примем точку  $(\tilde{x}_k^*, \tilde{\Phi}_k^*)$  •

На рис. 1 интервалами унимодальности являются интервалы  $[\tilde{x}_1^1, \tilde{x}_2^1], [\tilde{x}_1^2, \tilde{x}_2^2], [\tilde{x}_1^3, \tilde{x}_2^3]$ .

Для определения того, возрастает или убывает в данной точке функция  $\Phi(x)$ , может использоваться ее первая разность в этой точке  $\Delta\Phi_x = \Phi(x) - \Phi(x + \Delta x)$ , где  $\Delta x$  – некоторая малая величина. А именно, если  $\Delta\Phi_x > 0$ , то функция возрастает в точке  $x$ ; иначе – убывает. Заметим, что при этом в каждой точке требуется выполнить дополнительное испытание функции  $\Phi(x)$ .

Если функция  $\Phi(x)$  непрерывно дифференцируема в интервале  $[a, b]$ , то для определения того, возрастает или убывает в данной точке эта функция, можно, очевидно, использовать значения первой производной функции  $\Phi'(x)$  в этой точке. А именно, если  $\Phi'(x) > 0$ , то в точке  $x$  функция  $\Phi(x)$  возрастает; в противном случае – убывает.

Замечание 1. Если априорная оценка  $d$  минимального расстояния между локальными минимумами функции  $\Phi(x)$  отсутствует, то никаких оснований полагать, что в интервалах, выделенных с помощью рассмотренного алгоритма, функция  $\Phi(x)$  является унимодальной функцией. Пусть, например, функция  $\Phi(x)$  на интервале  $[c, d]$  постоянна (см. рис. 2). Если к такой функции применить алгоритм выделения интервалов унимодальности с любым  $\Delta > 0$ , то в качестве интервала унимодальности будет выделен интервал  $[\tilde{x}_1^1, \tilde{x}_2^1] \supset [c, d]$ , на котором функция  $\Phi(x)$  имеет бесконечное количество минимумов, т.е. не является унимодальной функцией •

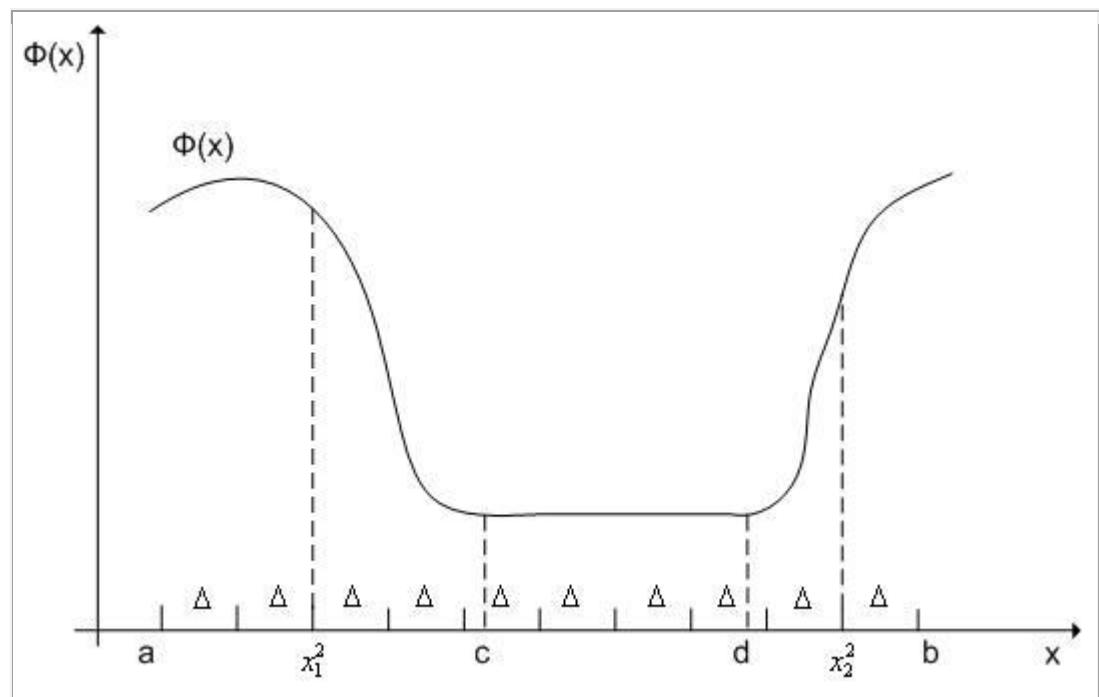


Рис. 2. К замечанию 1.

### 5.3. Метод аппроксимирующих моделей

Рассмотрим одномерную задачу условной глобальной оптимизации: найти минимум одномерной мультимодальной функции  $\Phi(x)$ , определенной в замкнутой области допустимых значений  $D=[a,b]$ :

$$\min_{x \in [a,b]} \Phi(x) = \Phi(x^*). \quad (1)$$

Схема метода

Схема метода аппроксимирующих моделей (см. рис. 1):

Покрываем интервал  $[a,b]$  некоторой сеткой с узлами  $x_i \in [a,b], x_i \neq x_j, i,j \in [0,M]$  и производим испытания в точках  $x_i, i \in [0,M]$ , т.е. вычисляем значения функции  $\Phi(x)$  в этих точках  $\Phi(x_i) = \Phi_i, i \in [0,M]$ .

Строим аппроксимирующую функцию  $\Psi(x)$ , проходящую через точки  $(x_i, \Phi_i), i \in [0,M]$ . Эту функцию принято называть математической моделью минимизируемой функции  $\Phi(x)$  или модельной функцией.

Оцениваем адекватность построенной модели  $\Psi(x)$ . Для этого:

производим дополнительные испытания функции  $\Phi(x)$  в некоторых точках  $x_k \in [a,b], k \in [1,M]$ ;

вычисляем значения модельной функции  $\Psi(x)$  и функции  $\Phi(x)$  в этих точках  $\Psi(x_k), \Phi(x_k), k \in [1,M]$ ;

вычисляем погрешность аппроксимации, например,  $\max_{k \in [1,M]} |\Psi(x_k) - \Phi(x_k)|$ .  
Если погрешность аппроксимации превышает заданную, то по результатам всех предшествующих испытаний строим новую модельную функцию  $\Psi(x)$  и переходим на п. 3).

Определяем положение глобального минимума модельной функции  $\Psi(x)$ , который или принимается в качестве глобального минимума функции  $\Phi(x)$ , или уточняется с помощью какого-либо метода локальной оптимизации •

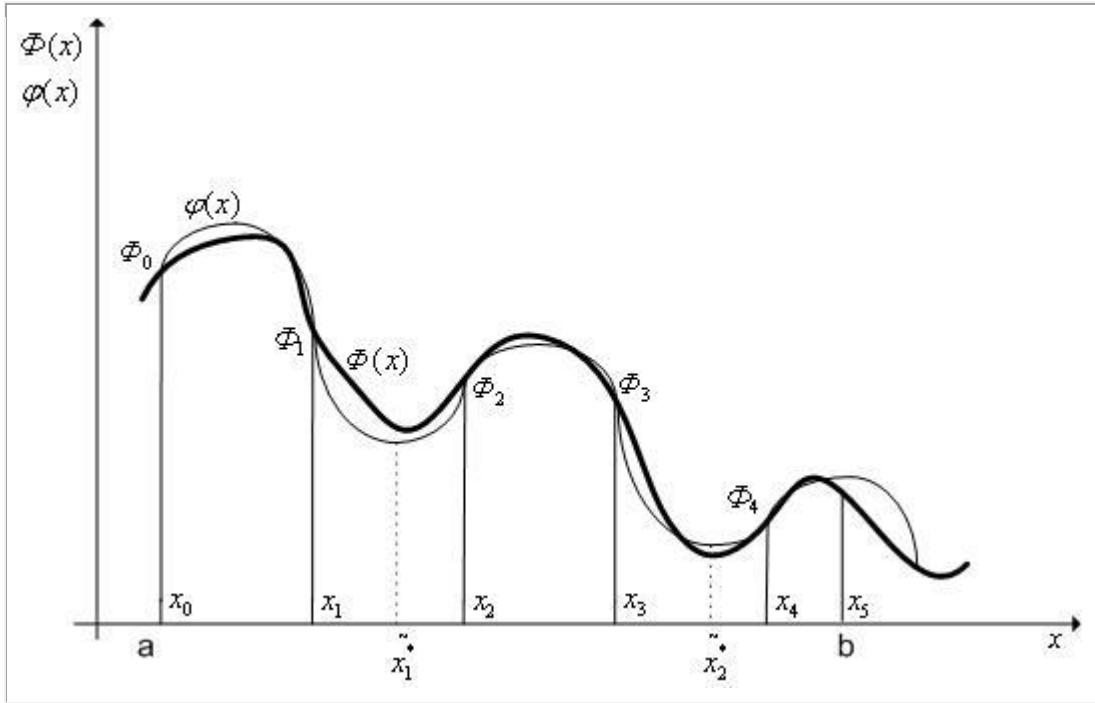


Рис. 1. К схеме метода аппроксимирующих моделей.  $N=5$ ;

На рис. 1  $x_1^*$ ,  $x_2^*$ - точки локального минимума модельной функции  $\Phi(x)$ ; точка  $\tilde{x}_2^*$ - приближенное значение точки глобального минимума функции  $\Phi(x)$  на интервале  $[a,b]$ .

В качестве модельных функций  $\Phi(x)$  чаще всего используют полиномы и сплайны.  
Рассмотрим использование в качестве модельной функции полиномов.

Аппроксимирующий полином Лагранжа

Будем искать аппроксимирующий полином в виде

$$\varphi(x)=\Phi(x_0)\Psi_0(x)+\Phi(x_1)\Psi_1(x)+\Phi(x_2)\Psi_2(x)+\dots+\Phi(x_N)\Psi_N(x)=\sum_{i=0}^N \Phi(x_i)\Psi_i(x)$$

где  $\Psi_i(x)$ ,  $i \in [0, N]$ - неизвестные полиномы от  $x$ , независящие от аппроксимируемой функции  $\Phi(x)$ .

Из того условия, что модельная функция  $\Phi(x)$  должна совпадать с аппроксимируемой функцией  $\Phi(x)$  в узлах сетки  $x_i$ ,  $i \in [0, N]$ , имеем систему из  $N+1$  равенств

$$\varphi(x_i)=\Phi(x_i)=\Phi(x_0)\Psi_0(x_i)+\Phi(x_1)\Psi_1(x_i)+\Phi(x_2)\Psi_2(x_i)+\dots+\Phi(x_N)\Psi_N(x_i), i \in [0, N]$$

Для выполнения равенств (3) полиномы  $\Psi_i(x)$ ,  $i \in [0, N]$ , очевидно, должны удовлетворять условиям

$$\Psi_i(x_j)=\begin{cases} 0, & i \neq j \\ 1, & i=j \end{cases} \quad (4)$$

или, другими словами, полином  $\Psi_i(x)$ ,  $i \in [0, N]$  должен иметь в качестве корней все числа  $x_i$ ,  $i \in [0, N]$ , кроме числа  $x_j$ , а при  $x=x_j$  должен иметь значение, равное единице.

Условию (4) удовлетворяют только полиномы вида  
 $\Psi_i(x)=A_i(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_N)$ , где  $A$  – неизвестная константа.

Найдем эту константу из условия  $\Psi_i(x_i)=1$ :

$$1=A_i(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_N);$$

$$A_i=\frac{1}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_N)}.$$

Таким образом,

$$\Psi_i(x)=\frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_N)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_N)}=\prod_{j=0, j \neq i}^N \frac{(x-x_j)}{(x_i-x_j)} \quad (5)$$

и искомый аппроксимирующий полином определяют выражением

$$\varphi(x)=\sum_{i=0}^N \Phi(x_i) \Psi_i(x)=L_N(x). \quad (6)$$

Полином (6) называется аппроксимирующим полиномом Лагранжа.

Использование аппроксимирующего полинома Лагранжа (6) в качестве модельной функции идеально очень просто, но обладает существенным недостатком. Пусть после построения этого полинома на сетке  $x_i, i \in [0, N]$  и проверке его адекватности выясняется, что погрешность аппроксимации превышает заданную. Тогда, в соответствии с рассмотренной выше схемой метода, необходимо построить новый полином Лагранжа на сетке, полученной объединением сеток  $x_i, i \in [0, N], x_k, k \in [0, M]$ , что требует пересчета всех посчитанных ранее функций  $\Psi_i(x), i \in [0, N]$ . От этого недостатка свободна модификация аппроксимирующего полинома Лагранжа – аппроксимирующий полином Ньютона.

Нахождение стационарных точек аппроксимирующего полинома.

После построения аппроксимирующего полинома возникает задача нахождения стационарных точек функции  $\Psi(x)$  (см. схему метода). Поскольку аппроксимирующий полином непрерывен и, по крайней мере, один раз непрерывно дифференцируем, его стационарные точки удобно искать как нули первой производной – т.е. как корни уравнения

$$\varphi'(x)=0. \quad (7)$$

Для поиска корней уравнения чаще всего используют метод хорд и метод касательных, использующие линейную интерполяцию функции  $\Psi(x)$  (см. параграф 4.8), а также другие методы.

## 6.1. Метод Гаусса-Зейделя

При решении задач САПР чаще всего приходится иметь дело с математическими моделями, в которых нет аналитических зависимостей для первых производных минимизируемой функции  $\Phi(\mathbf{X})$ . Поэтому поиск локального минимума в этом случае приходится вести по результатам вычислений только значений функции  $\Phi(\mathbf{X})$  – с помощью прямых методов оптимизации.

Рассматривается следующая многомерная задача безусловной оптимизации (точнее говоря, задача многомерной локальной безусловной оптимизации): найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного в  $n$ -мерном евклидовом пространстве  $R^n$ ,

$$\min_{\mathbf{X} \in R^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*. \quad (1)$$

При решении задачи (1) методом Гаусса-Зейделя (методом покоординатного спуска, методом циклического покоординатного спуска) используются следующие итерационные формулы

$$\begin{aligned} \mathbf{X}_1^{r+1} &= \mathbf{X}^r + \lambda_1^r \mathbf{L}_1, \\ \mathbf{X}_2^{r+1} &= \mathbf{X}_1^{r+1} + \lambda_2^r \mathbf{L}_2, \\ &\dots \\ \mathbf{X}_n^{r+1} &= \mathbf{X}_{n-1}^{r+1} + \lambda_n^r \mathbf{L}_n = \mathbf{X}^{r+1}, \end{aligned} \quad (2)$$

где вектор  $\mathbf{L}_i$  определяет направление вдоль  $i$ -й координатной оси и представляет собой

$n$ -мерный вектор с компонентами

$$l_{ij} = \begin{cases} 1, & \text{если } j=i, \\ 0, & \text{если } j \neq i, j=1,2,\dots,n, \end{cases}$$

а величины  $\lambda_1^r, \lambda_2^r, \dots, \lambda_n^r$  – определяются из условий

$$\begin{aligned} \min_{\lambda \in (-\infty, \infty)} \Phi(\mathbf{X}^r + \lambda \mathbf{L}_1) &= \Phi(\mathbf{X}^r + \lambda_1^r \mathbf{L}_1) = \Phi(\mathbf{X}_1^{r+1}), \\ \min_{\lambda \in (-\infty, \infty)} \Phi(\mathbf{X}_1^{r+1} + \lambda \mathbf{L}_2) &= \Phi(\mathbf{X}_1^{r+1} + \lambda_2^r \mathbf{L}_2) = \Phi(\mathbf{X}_2^{r+1}), \\ &\dots \\ \min_{\lambda \in (-\infty, \infty)} \Phi(\mathbf{X}_{n-1}^{r+1} + \lambda \mathbf{L}_n) &= \Phi(\mathbf{X}_{n-1}^{r+1} + \lambda_n^r \mathbf{L}_n) = \Phi(\mathbf{X}_n^{r+1}). \end{aligned} \quad (3)$$

Другими словами, величина  $\lambda_i^r, i=1,2,\dots,n$  представляет собой длину шага, минимизирующего функцию  $\Phi(\mathbf{X})$  в направлении  $\mathbf{L}_i$  на итерации номер  $r$ , исходя из точки, полученной на предыдущем шаге.

Если положить  $\mathbf{X}_0^{r+1} = \mathbf{X}^r, \mathbf{X}_n^{r+1} = \mathbf{X}^{r+1}$ , то формулы (2), (3) можно записать в виде

$$\mathbf{X}_i^{r+1} = \mathbf{X}_{i-1}^{r+1} + \lambda_i^r \mathbf{L}_i, \quad i=1,2,\dots,n, \quad (4)$$

$$\min_{\lambda \in (-\infty, \infty)} \Phi(\mathbf{X}_{i-1}^{r+1} + \lambda \mathbf{L}_i) = \Phi(\mathbf{X}_{i-1}^{r+1} + \lambda_i^r \mathbf{L}_i) = \Phi(\mathbf{X}_i^{r+1}), \quad i=1,2,\dots,n. \quad (5)$$

Таким образом, каждая итерация по методу Гаусса-Зейделя включает в себя  $n$  шагов. Каждая последующая итерация начинается из точки, полученной на последнем шаге предыдущей итерации. Поиск заканчивается при выполнении одного из стандартных условий окончания итераций:

$$\|\mathbf{X}^{r+1} - \mathbf{X}^r\| \leq \varepsilon_X; \quad (6)$$

$$|\Phi(\mathbf{X}^{r+1}) - \Phi(\mathbf{X}^r)| \leq \varepsilon_\Phi \quad (6')$$

Заметим, что задачи (5) даже в случае одноэкстремальной функции  $\Phi(\mathbf{X})$  могут быть задачами многоэкстремальной оптимизации и могут быть решены рассмотренными в главе 4 методами решения задач одномерной оптимизации.

Схема метода Гаусса-Зейделя:

Задаем начальную точку  $\mathbf{X}^0$  и полагаем  $r=0, i=1$ .

Последовательно для  $i=1,2,\dots,n$  решаем задачи (5), т.е. исходя из предыдущей точки, отыскиваем минимум функции  $\Phi(\mathbf{X})$  вдоль  $i$ -го координатного направления;

Если условие окончания поиска (6) или (6') выполнено, то полагаем  $\mathbf{X}^* \approx \mathbf{X}^{r+1}$  и заканчиваем вычисления. Иначе - полагаем  $r=r+1$  и переходим к п. 2.

Метод Гаусса-Зейделя иллюстрирует рис. 1, на котором показан фрагмент линий уровня функции Химмельблау  $\Phi(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$ . На рис. 1 точка  $\mathbf{X}_1^0$  представляет собой локальный минимум функции  $\Phi(\mathbf{X})$  вдоль оси  $X_1$  при исходной точке  $\mathbf{X}^0$ . Точка  $\mathbf{X}_2^1$  представляет собой локальный минимум функции  $\Phi(\mathbf{X})$  вдоль оси  $X_2$  при исходной точке  $\mathbf{X}_1^0$ . Отыскание точки  $\mathbf{X}_2^1$  завершает первую итерацию. Следующая итерация начинается из точки  $\mathbf{X}_1^1 = \mathbf{X}_2^1$ . И т.д.

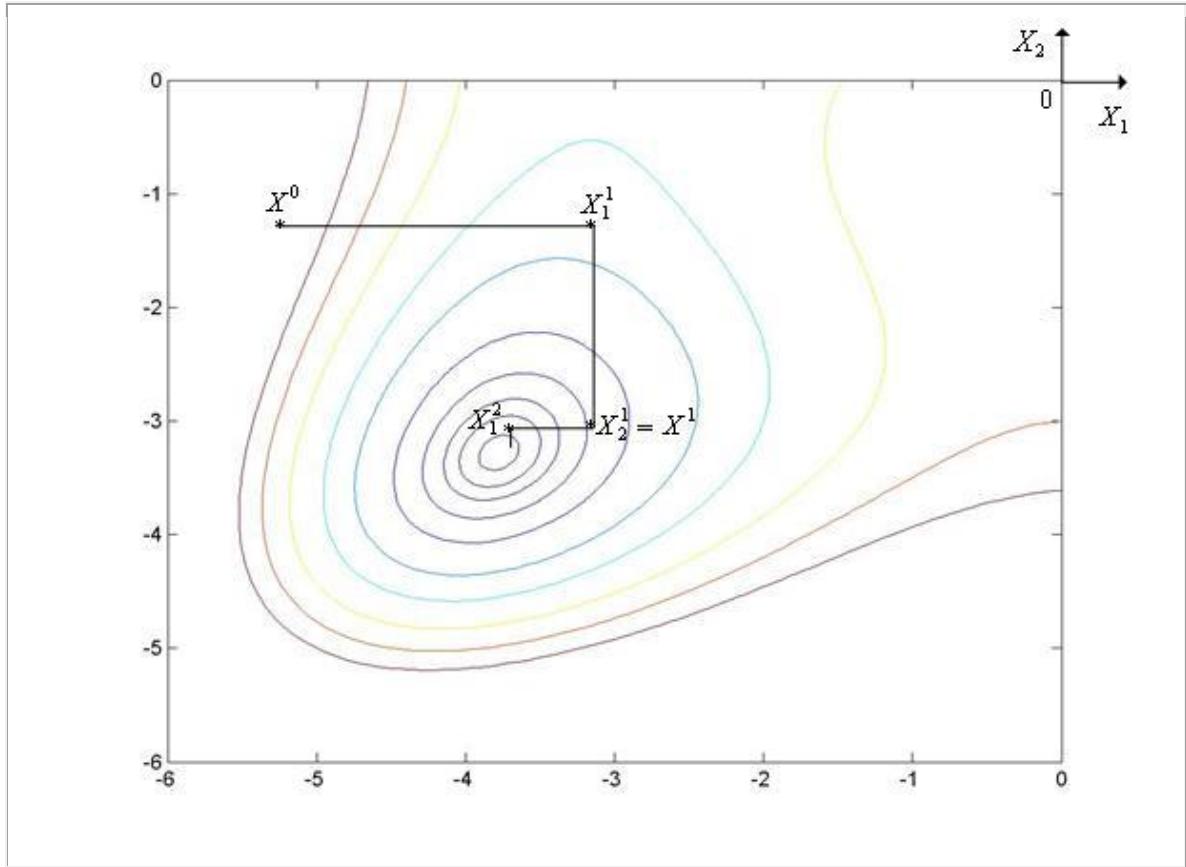


Рис. 1. Траектория поиска минимума не овражной функции Химмельблау методом Гаусса-Зейделя.

Линии уровня на рис. 1 получены с помощью следующей MATLAB-программы:

$x=-6:0.05:0;$

$y=x;$

$[X,Y]=meshgrid(x);$

$Z=(X.^2+Y-11).^2+(X+Y.^2-7).^2;$

$V=[1,4,8,16,32,64,100,150,200,250];$

$contour(X,Y,Z,V);$

Метод Гаусса-Зейделя медленно сходится на овражных функциях, в которых овраг не ориентирован в направлении какой-либо из координатных осей (см. рис. 2). На рисунке показаны линии уровня функции Розенброка ( $n=2$ ). Линии уровня получены с помощью следующей MATLAB-программы:

$x=-2:0.06:2;$

$y=x;$

$[X,Y]=meshgrid(x);$

$Z=100.*(Y-X.^2).^2+(1-X).^2;$

$V=[1,5,50,500];$

$[C,h]=contour(X,Y,Z,V);$

$clabel(C,h);$

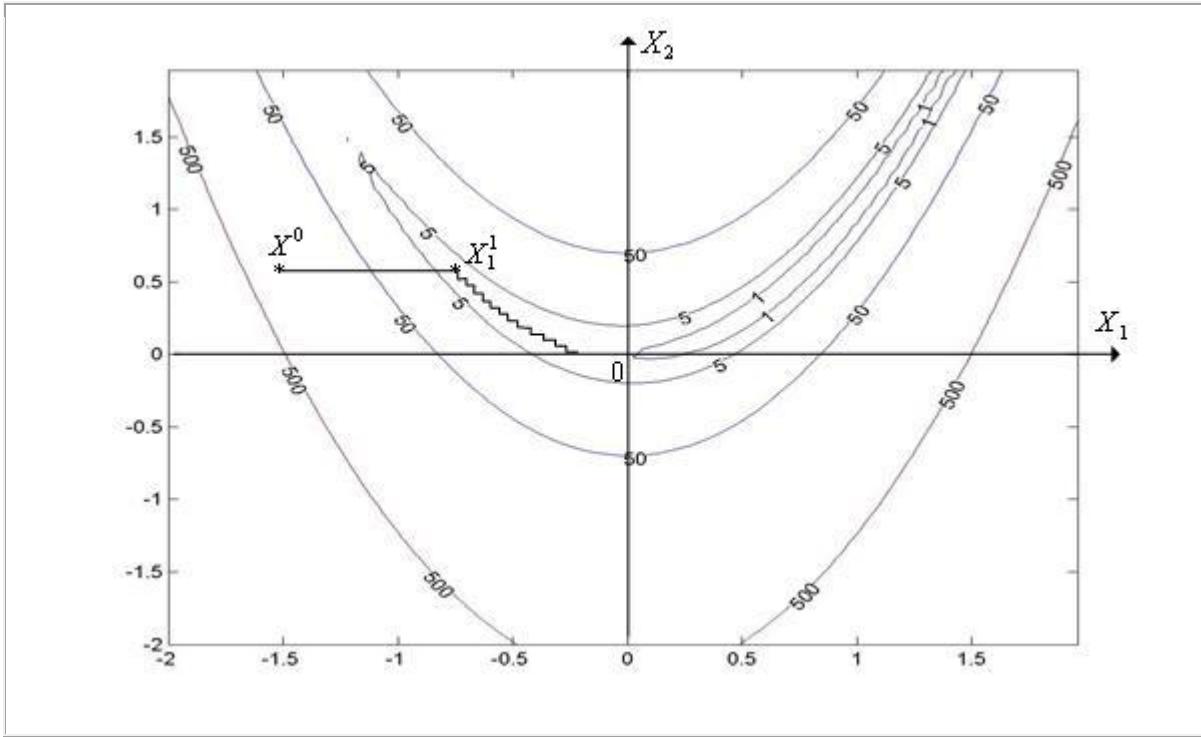


Рис. 2. Траектория поиска минимума овражной функции Розенброка методом Гаусса-Зейделя. Текущая точка быстро (в данном случае – за один шаг) «скатывается» на дно оврага и очень медленно движется по дну оврага к минимуму функции  $\Phi(X)$ .

## 6.2. Метод Хука-Дживса

Рассмотрим следующую многомерную задачу локальной безусловной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного в  $n$ -мерном евклидовом пространстве  $R^n$ ,

$$\min_{\mathbf{X} \in R^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*. \quad (1)$$

При решении задачи (1) методом Хука-Дживса (методом конфигураций, методом пробных шагов) используются итерационные формулы, аналогичные формулам, используемым в методе Гаусса-Зейделя

$$\tilde{\mathbf{X}}_i^{r+1} = \tilde{\mathbf{X}}_{i-1}^{r+1} + \lambda_i^r \mathbf{L}_i, \quad i=1,2,\dots,n, \quad (2)$$

$$\tilde{\mathbf{X}}_0^{r+1} = \mathbf{X}^r, \quad \tilde{\mathbf{X}}_n^{r+1} = \tilde{\mathbf{X}}^{r+1}$$

где принято  $\tilde{\mathbf{X}}_0^{r+1} = \mathbf{X}^r$ ,  $\tilde{\mathbf{X}}_n^{r+1} = \tilde{\mathbf{X}}^{r+1}$ , вектор  $\mathbf{L}_i$  определяет направление вдоль  $i$ -й координатной оси и представляет собой  $n$ -мерный вектор с компонентами

$$l_{ij} = \begin{cases} 1, & \text{если } j=i, \\ 0, & \text{если } j \neq i, \quad j=1,2,\dots,n, \end{cases}$$

величины  $\lambda_1^r, \lambda_2^r, \dots, \lambda_n^r$  – определяются из условий

$$\lambda_i^r = \begin{cases} \Delta_i^r, & \text{если } \Phi(\mathbf{X}_{i-1}^{r+1} + \Delta_i^r \mathbf{L}_i) < \Phi(\mathbf{X}_{i-1}^{r+1}), \\ -\Delta_i^r, & \text{если } \Phi(\mathbf{X}_{i-1}^{r+1} - \Delta_i^r \mathbf{L}_i) < \Phi(\mathbf{X}_{i-1}^{r+1}) < \Phi(\mathbf{X}_{i-1}^{r+1} + \Delta_i^r \mathbf{L}_i), \\ 0, & \text{если } \min[\Phi(\mathbf{X}_{i-1}^{r+1} + \Delta_i^r \mathbf{L}_i), \Phi(\mathbf{X}_{i-1}^{r+1} - \Delta_i^r \mathbf{L}_i)] > \Phi(\mathbf{X}_{i-1}^{r+1}). \end{cases} \quad (3)$$

После завершения  $n$  шагов выполняется спуск в направлении вектора  $(\tilde{\mathbf{X}}^{r+1} - \mathbf{X}^r)$  по формуле

$$\mathbf{X}^{r+1} = \mathbf{X}^r + \alpha^r (\tilde{\mathbf{X}}^{r+1} - \mathbf{X}^r), \quad (4)$$

где  $\alpha^r$  – ускоряющий множитель. В различных модификациях метода Хука-Дживса множитель  $\alpha^r$  может

приниматься постоянным (обычно, равным 2),

выбираться из условия  $\Phi(\mathbf{X}^{r+1}) < \Phi(\mathbf{X}^r)$ ,

находиться из условия локального минимума функции  $\Phi(\mathbf{X})$  при движении из точки  $\mathbf{X}^r$

в направлении вектора  $(\tilde{\mathbf{X}}^{r+1} - \mathbf{X}^r)$ :

$$\min_{\alpha \in (-\infty, \infty)} \Phi(\mathbf{X}^r + \alpha (\tilde{\mathbf{X}}^{r+1} - \mathbf{X}^r)) = \Phi(\mathbf{X}^r + \alpha^r (\tilde{\mathbf{X}}^{r+1} - \mathbf{X}^r)) = \Phi(\mathbf{X}^{r+1}). \quad (5)$$

Заметим, что задачи (5) даже в случае одноэкстремальной функции  $\Phi(\mathbf{X})$  могут быть многоэкстремальными задачами оптимизации и могут быть решены рассмотренными в главе 4 методами решения задач одномерной оптимизации.

Итерации заканчиваются при выполнении одного из стандартных условий окончания итераций:

$$\|\mathbf{X}^{r+1} - \mathbf{X}^r\| \leq \varepsilon_{\Phi}, \quad (6)$$

$$|\Phi(\mathbf{X}^{r+1}) - \Phi(\mathbf{X}^r)| \leq \varepsilon_{\Phi} \quad (6')$$

$$\Delta^0 = (\Delta_1^0, \Delta_2^0, \dots, \Delta_n^0)$$

Вектор  $\Delta^0 = (\Delta_1^0, \Delta_2^0, \dots, \Delta_n^0)$  является вектором свободных параметров метода - вектором «пробных шагов» по всем  $n$  координатным осям.

$$\tilde{\mathbf{X}}^{r+1}$$

Известна модификация метода Хука-Дживса, в которой точка  $\tilde{\mathbf{X}}^{r+1}$  определяется не процедурами (2), (3), а методом Гаусса-Зейделя.

Схема метода Хука-Дживса:

Задаем начальную точку  $\mathbf{X}^0$ , вектор «пробных» шагов  $\Delta^0 = (\Delta_1^0, \Delta_2^0, \dots, \Delta_n^0)$  и полагаем  $r = 0$ .

Последовательно для  $i = 1, 2, \dots, n$  по формулам (2), (3) находим точки  $\tilde{\mathbf{X}}_1^{r+1}, \tilde{\mathbf{X}}_2^{r+1}, \dots, \tilde{\mathbf{X}}_n^{r+1} = \tilde{\mathbf{X}}^{r+1}$

$$\tilde{\mathbf{X}}^{r+1} \neq \tilde{\mathbf{X}}^r$$

Если  $\Delta_1^0, \Delta_2^0, \dots, \Delta_n^0$ , то переходим к п. 4). Иначе уменьшаем длины «пробных» шагов  $\Delta_1^0, \Delta_2^0, \dots, \Delta_n^0$ , например, вдвое и переходим к п.2).

Если условие окончания поиска (6) или (6') выполнено, то полагаем  $\mathbf{X}^* \approx \mathbf{X}^{r+1}$  и  $(\tilde{\mathbf{X}}^{r+1} - \mathbf{X}^r)$  заканчиваем вычисления. Иначе выполняем спуск в направлении вектора  $\alpha^r$  по формуле (4), в которой ускоряющий множитель находится, например, из условия (5). Полагаем  $r=r+1$  и переходим к п. 2 •

Метод Хука-Дживса иллюстрирует рис. 1, на котором показаны линии уровня функции Химмельблау ( $n=2$ ). Ускоряющий множитель  $\alpha^r$  находится из условия локального минимума функции  $\Phi(\mathbf{X})$  при движении из точки  $\mathbf{X}^r$  в направлении вектора  $(\tilde{\mathbf{X}}^{r+1} - \mathbf{X}^r)$

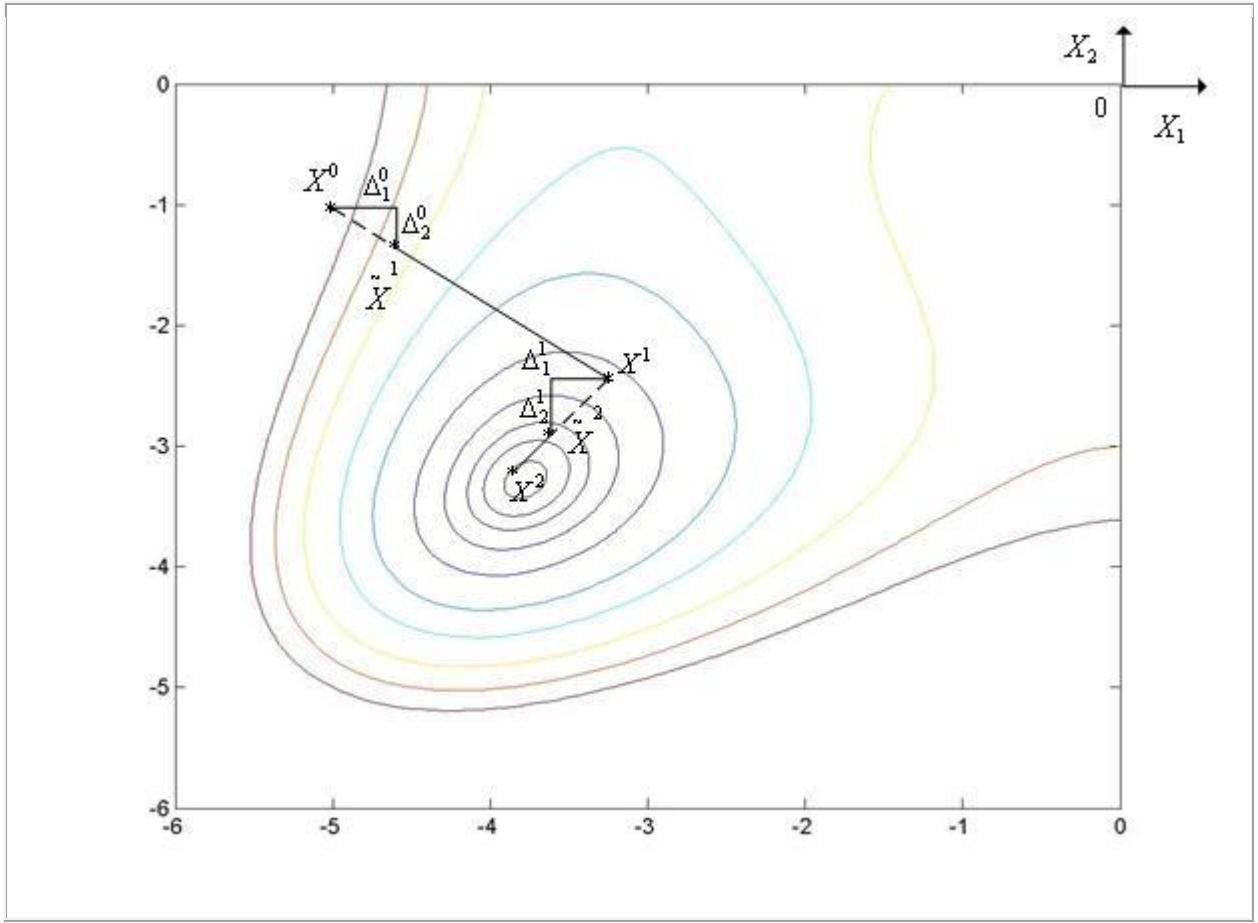


Рис. 1. Траектория поиска минимума не овражной функции Химмельблау методом Хука-Дживса.

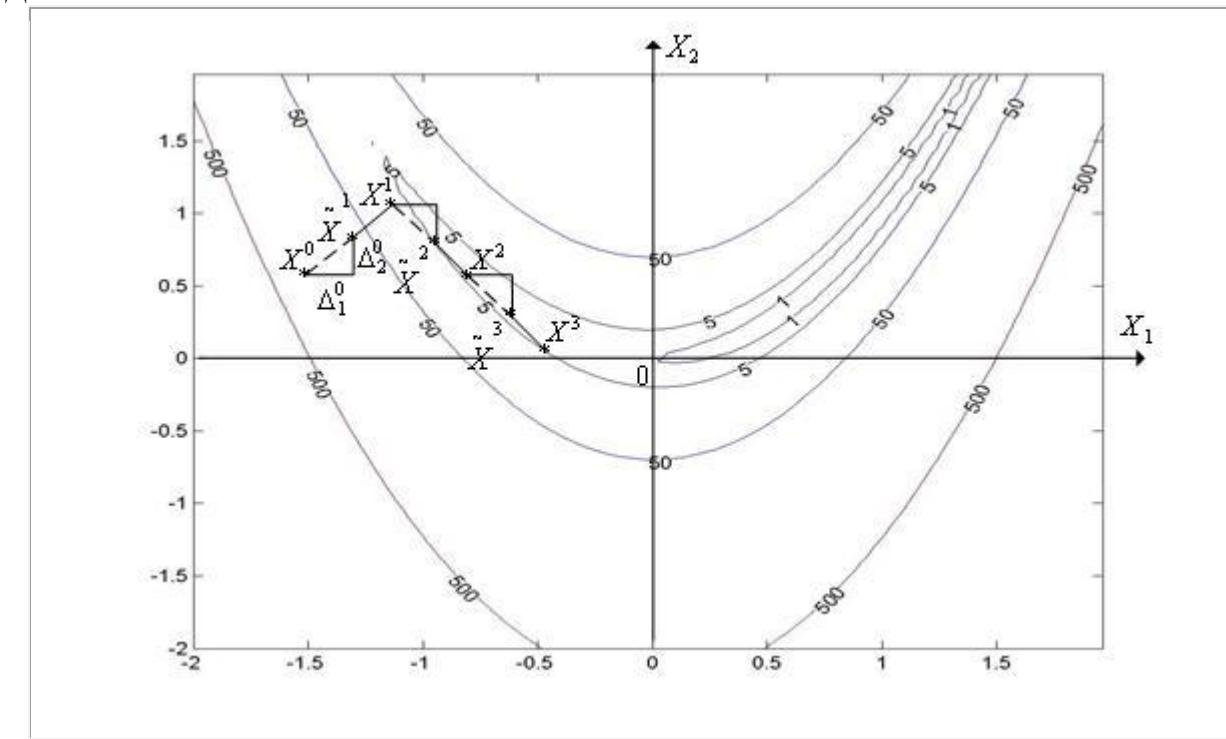


Рис. 2. Траектория поиска минимума овражной функции Розенброка методом Хука-Дживса. Ускоряющий множитель  $\alpha$  принят равным двум.

Метод Хука-Дживса имеет высокую эффективность в случае, если функция  $\Phi(\mathbf{X})$  имеет прямолинейный овраг (не обязательно ориентированный вдоль одного из координатных направлений, как в методе Гаусса-Зейделя). При минимизации "овражных" функций,

имеющих не прямолинейный овраг, процесс поиска может сильно замедлиться и закончиться далеко от точки истинного минимума (см. рис. 2). На рисунке показаны линии уровня функции Розенброка ( $n=2$ ).

### 6.3. Метод Розенброка

Рассмотрим следующую задачу локальной безусловной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного в  $n$ -мерном евклидовом пространстве  $R^n$ ,

$$\min_{\mathbf{X} \in R^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*. \quad (1)$$

Ортогонализация Грамма-Шмидта.

При решении задачи (1) методом Розенброка (методом вращающихся координат) используется преобразование на каждой итерации системы координат таким образом, чтобы в новой системе координат одна из осей совпадала с направлением предыдущего шага. Остальные оси новой системы координат обычно находят с помощью процедуры ортогонализации Грамма-Шмидта.

Рассмотрим произвольный набор векторов  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$  пространства  $R^n$ . Поставим задачу построить на основе этих векторов ортонормированный набор векторов  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$  того же пространства  $R^n$ .

Напомним, что набор векторов  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$  называется ортонормированным, если для любых двух векторов из этого набора выполняется условие

$$\langle \mathbf{e}_i, \mathbf{e}_j \rangle = \begin{cases} 0, & i \neq j, \\ 1, & i = j. \end{cases} \quad (2)$$

Или, другими словами, набор векторов  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$  ортонормирован, если эти векторы линейно независимы и скалярное произведение любых двух из них равно единице.

Для построения векторов  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$  применим индуктивный подход. Положим, что

$$\mathbf{e}_1 = \frac{\mathbf{p}_1}{\|\mathbf{p}_1\|}, \quad (3)$$

где  $\|\cdot\|$  - символ евклидовой нормы. Полагая векторы  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{k-1}$  уже построенными будем искать вектор  $\mathbf{e}_k$  в виде

$$\mathbf{p}_{kk} \mathbf{e}_k = \mathbf{p}_k - \sum_{i=1}^{k-1} p_{ik} \mathbf{e}_i. \quad (4)$$

Для отыскания неизвестных множителей  $p_{ik}$  умножим (4) скалярно на вектор  $\mathbf{e}_j, j=1, 2, \dots, k-1$ :

$$\begin{aligned}
\rho_{kk}(\mathbf{e}_k, \mathbf{e}_1) &= 0 = (\mathbf{p}_k, \mathbf{e}_1) - \sum_{i=1}^{k-1} \rho_{ik}(\mathbf{e}_i, \mathbf{e}_1) = (\mathbf{p}_k, \mathbf{e}_1) - \rho_{1k}(\mathbf{e}_1, \mathbf{e}_1), \\
\rho_{kk}(\mathbf{e}_k, \mathbf{e}_2) &= 0 = (\mathbf{p}_k, \mathbf{e}_2) - \sum_{i=1}^{k-1} \rho_{ik}(\mathbf{e}_i, \mathbf{e}_2) = (\mathbf{p}_k, \mathbf{e}_2) - \rho_{2k}(\mathbf{e}_2, \mathbf{e}_2), \\
&\dots \\
\rho_{kk}(\mathbf{e}_k, \mathbf{e}_{k-1}) &= 0 = (\mathbf{p}_k, \mathbf{e}_{k-1}) - \sum_{i=1}^{k-1} \rho_{ik}(\mathbf{e}_i, \mathbf{e}_{k-1}) = (\mathbf{p}_k, \mathbf{e}_{k-1}) - \rho_{k-1,k}(\mathbf{e}_{k-1}, \mathbf{e}_{k-1}).
\end{aligned}$$

Поскольку  $\langle \mathbf{e}_j, \mathbf{e}_j \rangle = 1$ , имеем

$$\rho_{jk} = (\mathbf{p}_k, \mathbf{e}_j), j \in [1, k-1]. \quad (5)$$

Множитель  $\rho_{kk}$  найдем из условия  $\|\mathbf{e}_k\| = 1$ :

$$\rho_{kk} = \left\| \mathbf{p}_k - \sum_{i=1}^{k-1} \rho_{ik} \mathbf{e}_i \right\|. \quad (6)$$

**Определение 1.** Процесс перехода от векторов  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$  к векторам  $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$  согласно формулам (3) – (6) называется ортогонализацией Грамма-Шмидта •

Схема метода Розенброка.

Каждая итерация метода Розенброка состоит из двух этапов. В зависимости от модификации метода первый этап может выполняться с использованием различных методов. Рассмотрим применение на первом этапе итерационной формулы метода Гаусса-Зейделя. Приведем формулировку этой формулы, несколько отличную от формулировки, рассмотренной в параграфе 6.1.

Положим  $\mathbf{X}_0^{r+1} = \mathbf{X}^r$ ,  $\mathbf{X}_n^{r+1} = \mathbf{X}^{r+1}$  и пусть  $\mathbf{e}_1^r, \mathbf{e}_2^r, \dots, \mathbf{e}_n^r$  – орты системы координат, используемой на  $r$ -ой итерации. Тогда итерационную формулу метода Гаусса-Зейделя можно записать в виде

$$\mathbf{X}_i^{r+1} = \mathbf{X}_{i-1}^{r+1} + \lambda_i^r \mathbf{e}_i^r = \mathbf{X}_{i-1}^{r+1} + \mathbf{q}_i^r, \quad i = 1, 2, \dots, n, \quad (7)$$

где коэффициенты  $\lambda_i^r$  находятся из условий

$$\min_{\lambda \in (-\infty, \infty)} \Phi(\mathbf{X}_{i-1}^{r+1} + \lambda \mathbf{e}_i^r) = \Phi(\mathbf{X}_{i-1}^{r+1} + \lambda_i^r \mathbf{e}_i^r) = \Phi(\mathbf{X}_i^{r+1}), \quad i = 1, 2, \dots, n. \quad (8)$$

На втором этапе каждой из итераций система векторов  $\mathbf{e}_1^r, \mathbf{e}_2^r, \dots, \mathbf{e}_n^r$  с использованием ортогонализации Грамма-Шмидта заменяется новой системой линейно независимых векторов  $\mathbf{e}_1^{r+1}, \mathbf{e}_2^{r+1}, \dots, \mathbf{e}_n^{r+1}$ .

Схема метода Розенброка:

Задаем начальную точку  $\mathbf{X}^0$ , полагаем  $r=0$ ,  $i=1$ , и орты исходной системы координат обозначаем  $\mathbf{e}_1^0, \mathbf{e}_2^0, \dots, \mathbf{e}_n^0$ .

Исходя из точки  $\mathbf{X}^r$  по формулам (7), (8) выполняем одну итерацию по методу Гаусса-Зейделя – получаем точку  $\mathbf{X}^{r+1}$  и совокупность векторов  $\mathbf{q}_1^r, \mathbf{q}_2^r, \dots, \mathbf{q}_n^r$ .  
Если одно из стандартных условий окончания итераций  
 $\|\mathbf{X}^{r+1} - \mathbf{X}^r\| \leq \varepsilon_X$ , (9)

$$|\Phi(\mathbf{X}^{r+1}) - \Phi(\mathbf{X}^r)| \leq \varepsilon_\Phi$$

выполнено, то полагаем  $\mathbf{X}^* \approx \mathbf{X}^{r+1}$ , и заканчиваем вычисления. Иначе переходим к п.4).

На основе векторов  $\mathbf{q}_1^r, \mathbf{q}_2^r, \dots, \mathbf{q}_n^r$  находим векторы  $\mathbf{p}_1^r, \mathbf{p}_2^r, \dots, \mathbf{p}_n^r$ :

$$\mathbf{p}_i^r = \sum_{j=i}^n \mathbf{q}_j^r, \quad i=1,2,\dots,n. \quad (10)$$

С помощью процедуры ортогонализации Грамма-Шмидта (3) –(6) выполняем переход от системы векторов  $\mathbf{p}_1^r, \mathbf{p}_2^r, \dots, \mathbf{p}_n^r$  к системе векторов  $\mathbf{e}_1^{r+1}, \mathbf{e}_2^{r+1}, \dots, \mathbf{e}_n^{r+1}$ , полагаем  $r=r+1$  и переходим к п. 2 ●

Заметим, что из формулы (10) следует равенство  $\mathbf{p}_n^r = \mathbf{q}_n^r$ .

По сравнению с методом Гаусса-Зейделя и методом Хука-Дживса метод Розенброка имеет, как правило, более высокую эффективность на овражных функциях с не прямолинейным оврагом.

Метод Розенброка иллюстрирует рис. 1, на котором показаны линии уровня функции Химмельблау ( $n=2$ ),

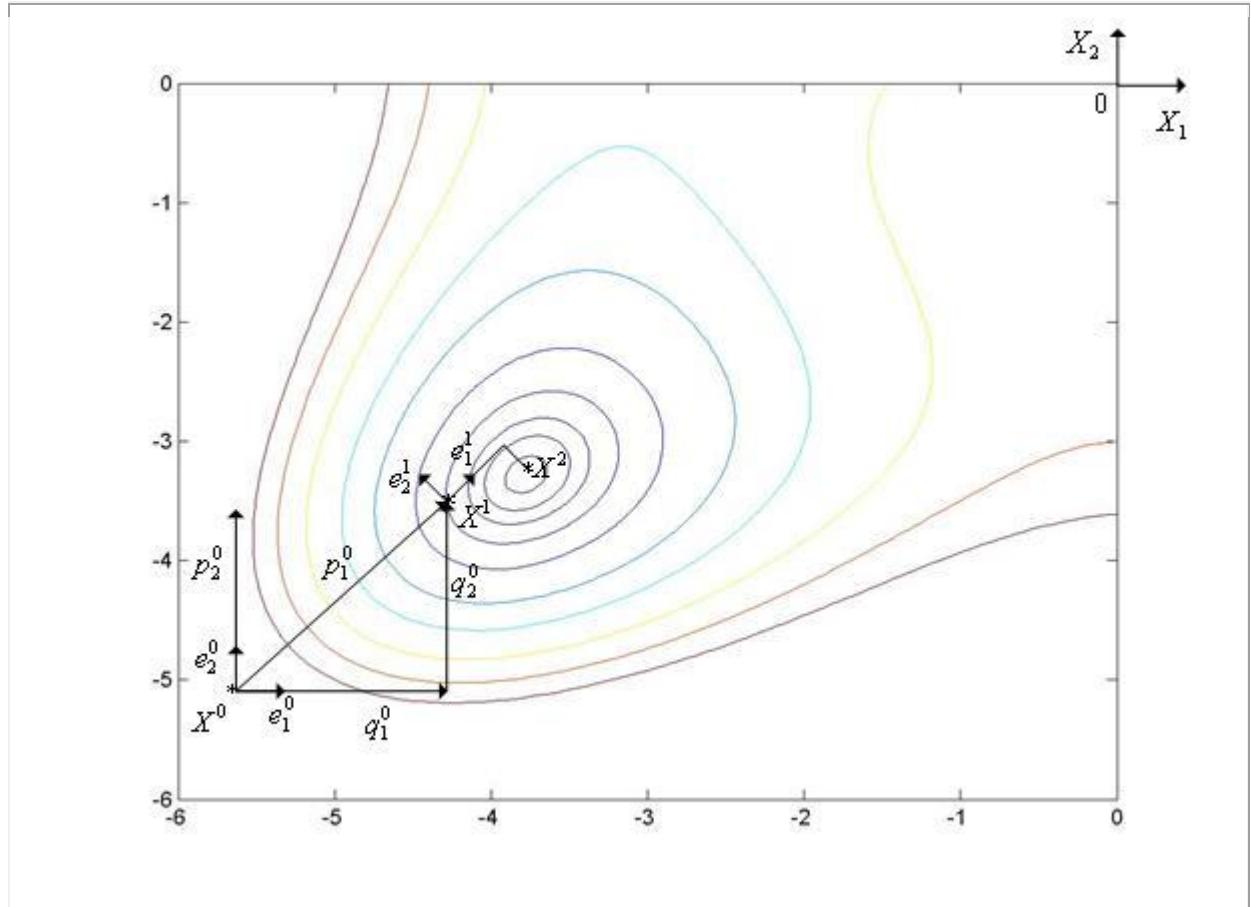


Рис. 1. Траектория поиска минимума функции Химмельблау методом Розенброка.

#### 6.4. Метод сопряженных направлений

Рассмотрим следующую задачу локальной безусловной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного в  $n$ -мерном евклидовом пространстве  $R^n$ ,

$$\min_{\mathbf{X} \in R^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*. \quad (1)$$

Введем прежде следующие понятия: векторы  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k$ , принадлежащие пространству  $R^n$ , называются векторами сопряженными относительно матрицы A ( $n \times n$ ) (A - ортогональными векторами), если  $\langle A \mathbf{p}_i, \mathbf{p}_j \rangle = 0$  для всех  $i \neq j, i, j \in [1, k]$ .

В методе сопряженных направлений применяется итерационная формула метода Гаусса-Зейделя в виде, близком к использованному в параграфе 6.3.

Положим  $\mathbf{X}_0^{r+1} = \mathbf{X}^r$  и пусть  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$  - орты используемой системы координат. Тогда итерационную формулу метода Гаусса-Зейделя можно записать в виде

$$\mathbf{X}_i^{r+1} = \mathbf{X}_{i-1}^{r+1} + \lambda_i^r \mathbf{e}_i = \mathbf{X}_{i-1}^{r+1} + \mathbf{p}_i^r, \quad i=1, 2, \dots, n, \quad (2)$$

где коэффициенты  $\lambda_i^r$  находятся из условий

$$\min_{\lambda \in (-\infty, \infty)} \Phi(\mathbf{X}_{i-1}^{r+1} + \lambda \mathbf{e}_i) = \Phi(\mathbf{X}_{i-1}^{r+1} + \lambda_i^r \mathbf{e}_i) = \Phi(\mathbf{X}_i^{r+1}), \quad i=1, 2, \dots, n. \quad (3)$$

Схема метода сопряженных направлений:

Задаем начальную точку  $\mathbf{X}^0$  и полагаем  $r=0, i=1$ .

Последовательно для  $i=1, 2, \dots, n$  по формулам (2), (3) находим точки  $\mathbf{X}_1^{r+1}, \mathbf{X}_2^{r+1}, \mathbf{X}_n^{r+1}$ .

Исходя из точки  $\mathbf{X}_n^{r+1}$ , еще раз находим минимум функции  $\Phi(\mathbf{X})$  вдоль первого координатного направления - вычисляем координаты точки

$$\mathbf{X}_{n+1}^{r+1} = \mathbf{X}_n^{r+1} + \tilde{\lambda}_1^r \mathbf{e}_1, \quad (4)$$

где коэффициент  $\tilde{\lambda}_1^r$  находится из условия

$$\min_{\lambda \in (-\infty, \infty)} \Phi(\mathbf{X}_{n+1}^{r+1} + \lambda \mathbf{e}_1) = \Phi(\mathbf{X}_n^{r+1} + \tilde{\lambda}_1^r \mathbf{e}_1) = \Phi(\mathbf{X}_{n+1}^{r+1}). \quad (5)$$

Исходя из точки  $\mathbf{X}_1^{r+1}$ , находим минимум функции  $\Phi(\mathbf{X})$  вдоль вектора  $\mathbf{p}_{n+1}^r = \mathbf{X}_{n+1}^{r+1} - \mathbf{X}_1^{r+1}$  - вычисляем

$$\mathbf{X}^{r+1} = \mathbf{X}_1^{r+1} + \lambda_{n+1}^{r+1} \mathbf{p}_{n+1}^r, \quad (6)$$

где коэффициент  $\lambda_{n+1}^r$  находится из условия

$$\min_{\lambda \in (-\infty, \infty)} \Phi(\mathbf{X}_1^{r+1} + \lambda \mathbf{p}_{n+1}^r) = \Phi(\mathbf{X}^r + \lambda_{n+1}^r \mathbf{p}_{n+1}^r) = \Phi(\mathbf{X}^{r+1}). \quad (7)$$

Если одно из стандартных условий окончания итераций

$$\|\mathbf{X}^{r+1} - \mathbf{X}^r\| \leq \varepsilon_X, \quad (8)$$

$$|\Phi(\mathbf{X}^{r+1}) - \Phi(\mathbf{X}^r)| \leq \varepsilon_\Phi$$

выполнено, то полагаем  $\mathbf{X}^* \approx \mathbf{X}^{r+1}$ , и заканчиваем вычисления. Иначе - полагаем  $r = r + 1$  и переходим к п.2) ●

Метод сопряженных направлений иллюстрирует рис. 1, на котором показаны линии уровня функции Химмельблау ( $n=2$ ).

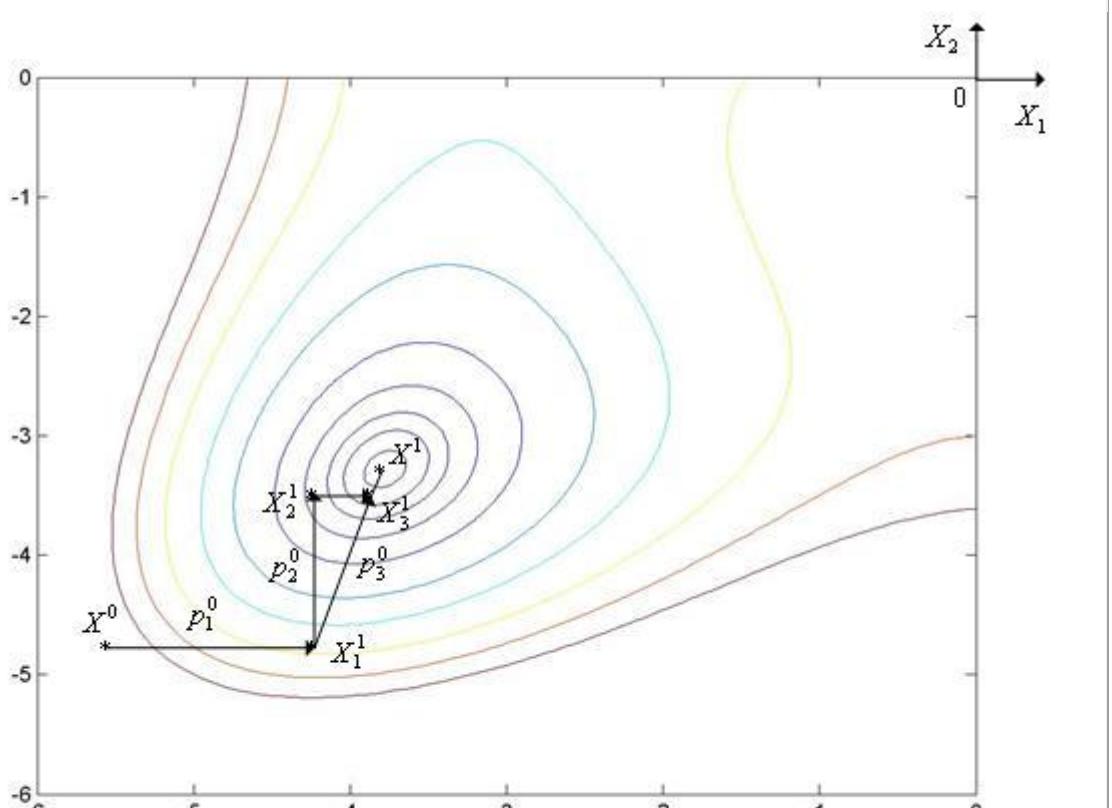


Рис. 1. Траектория поиска минимума функции Химмельблау методом сопряженных направлений.

Рассмотрим еще один пример – см. рис. 2, на котором показаны линии уровня двумерной квадратичной функции

$$\Phi(x_1, x_2) = x_1^2 + x_2^2 + 3(x_1 + x_2)^2 = 4x_1^2 + 4x_2^2 + 6x_1 x_2. \quad (9)$$

Линии уровня получены с помощью MATLAB-программы, приведенной в параграфе 6.1. Произвольную  $n$ -мерную квадратичную функцию можно записать в виде

$$\Phi(\mathbf{X}) = \frac{1}{2}(\mathbf{AX}, \mathbf{X}^T) + (\mathbf{b}, \mathbf{X}) + c, \quad (10)$$

где  $\mathbf{A}$  – квадратная  $n \times n$  матрица,  $\mathbf{b}$  –  $n \times 1$  столбец,  $c$  – скалярная константа. Например, если положить

$\mathbf{A} = \begin{pmatrix} 8 & 6 \\ 6 & 8 \end{pmatrix}$ ,  $\mathbf{b} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ ,  $c = 0$ , то имеем функцию (9):

$$\begin{aligned} \Phi(\mathbf{X}) &= \frac{1}{2}(\mathbf{AX}, \mathbf{X}^T) + (\mathbf{b}, \mathbf{X}) + c = \frac{1}{2}((8x_1 + 6x_2, 6x_1 + 8x_2), (x_1, x_2)) = \\ &= \frac{1}{2}(8x_1^2 + 6x_1x_2 + 6x_1x_2 + 8x_2^2) = 4x_1^2 + 4x_2^2 + 6x_1x_2. \end{aligned}$$

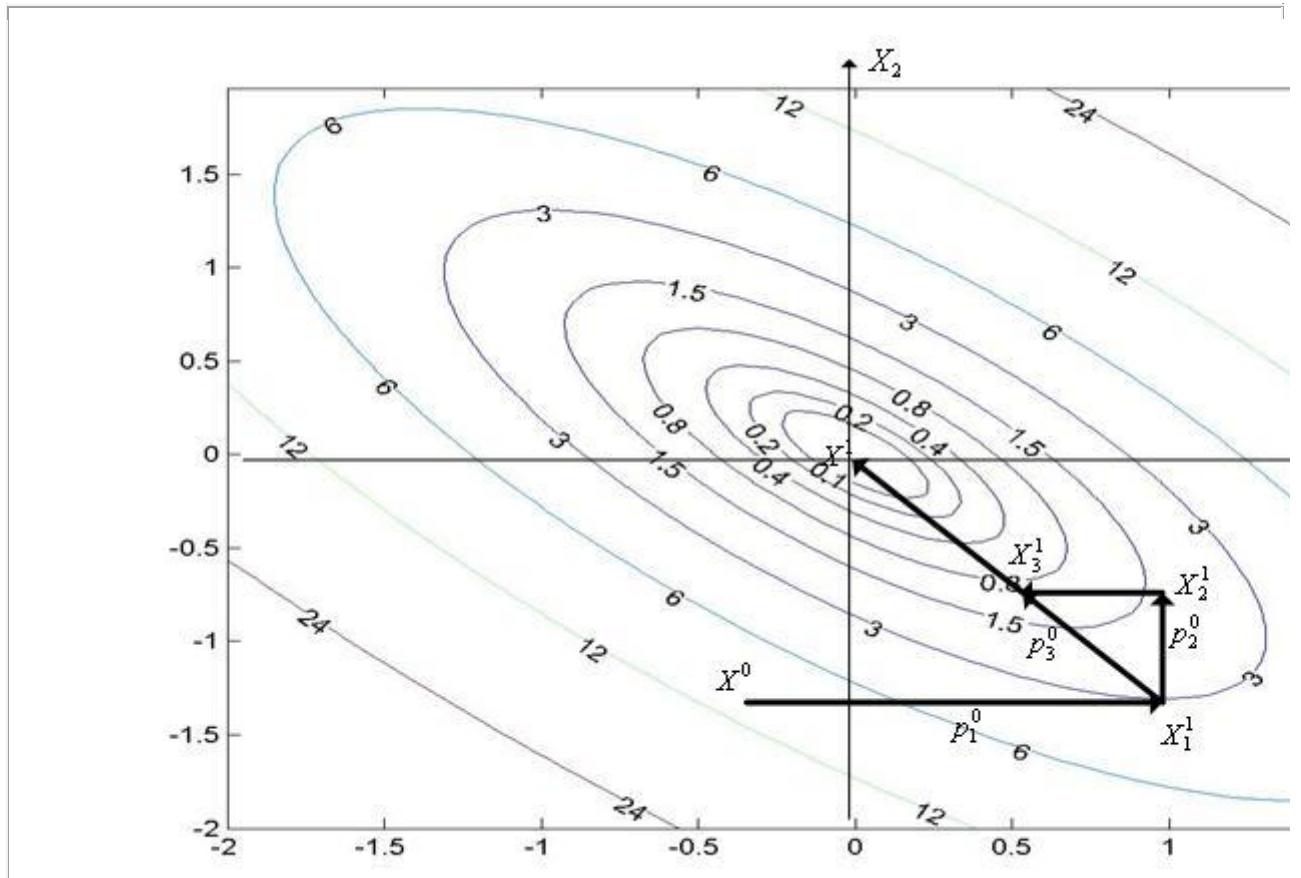


Рис. 2. Траектория поиска минимума квадратичной функции (9) методом сопряженных направлений.

Утверждение 1. В случае минимизации двумерной квадратичной функции (10) методом сопряженных направлений, направления  $\mathbf{p}_1^r, \mathbf{p}_3^r$  являются  $\mathbf{A}$ -ортогональными.

Доказательство (см. рис. 2). По определению  $\mathbf{A}$ -ортогональности для доказательства утверждения достаточно показать, что скалярное произведение

$$(\mathbf{A}\mathbf{p}_3^r, \mathbf{p}_1^r) = (\mathbf{A}(\mathbf{X}_3^r - \mathbf{X}_1^r), \mathbf{p}_1^r) = 0. \quad (11)$$

Легко видеть, что производная функции (10) равна  $\Phi'(\mathbf{X}) = \mathbf{A}\mathbf{X} + \mathbf{b}$ . Поэтому  $\mathbf{A}(\mathbf{X}_3^r - \mathbf{X}_1^r) = \mathbf{A}\mathbf{X}_3^r - \mathbf{A}\mathbf{X}_1^r = \Phi'(\mathbf{X}_3^r) - \Phi'(\mathbf{X}_1^r)$ . Подставляя этот результат в выражение (11), получим  $(\mathbf{A}\mathbf{p}_3^r, \mathbf{p}_1^r) = ((\Phi'(\mathbf{X}_3^r) - \Phi'(\mathbf{X}_1^r)), \mathbf{p}_1^r) = (\Phi'(\mathbf{X}_3^r), \mathbf{p}_1^r) - (\Phi'(\mathbf{X}_1^r), \mathbf{p}_1^r) = 0$ .

Последнее равенство следует из ортогональности пар векторов  $(\Phi'(\mathbf{X}_3^r), \mathbf{p}_1^r)$ ,  $(\Phi'(\mathbf{X}_1^r), \mathbf{p}_1^r)$ .

Утверждение 1 объясняет название рассмотренного метода.

Заметим, что при минимизации квадратичной функции методом сопряженных направлений минимум достигается за одну итерацию.

## 6.5. Симплекс-метод

Рассмотрим следующую задачу локальной безусловной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного в  $n$ -мерном евклидовом пространстве  $R^n$ ,

$$\min_{\mathbf{X} \in R^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*. \quad (1)$$

Регулярный симплекс и некоторые операции над ним.

Регулярным симплексом в пространстве  $R^n$  называется правильный многогранник, образованный  $(n+1)$ -ой равноотстоящими друг от друга вершинами. Для случая  $n=2$  – это равносторонний треугольник, для случая  $n=3$  – тетраэдр.

Если в пространстве  $R^n$  необходимо построить регулярный симплекс, одна из вершин которого находится в точке  $\mathbf{X}^0 = (x_1^0, x_2^0, \dots, x_n^0)$ , то координаты вершин такого симплекса удобно задавать с помощью  $n^{*(n+1)}$  матрицы

$$\mathbf{R}_{n \times (n+1)} = \begin{bmatrix} x_1^0 & x_1^0 + r_1 & x_1^0 + r_2 & \dots & x_1^0 + r_2 \\ x_2^0 & x_2^0 + r_2 & x_2^0 + r_1 & \dots & x_2^0 + r_2 \\ x_3^0 & x_3^0 + r_2 & x_3^0 + r_2 & \dots & x_3^0 + r_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n^0 & x_n^0 + r_2 & x_n^0 + r_2 & \dots & x_n^0 + r_1 \end{bmatrix}. \quad (2)$$

Здесь  $i$ -й столбец представляет собой координаты  $i$ -й вершины симплекса,  $i \in [1, n+1]$ ;

$$r_1 = l \frac{\sqrt{n+1+n-1}}{n\sqrt{2}}, \quad r_2 = l \frac{\sqrt{n+1}-1}{n\sqrt{2}}, \quad (3)$$

$l$  – длина ребра симплекса.

Например, регулярный симплекс в двумерном пространстве  $R^2$  с одной из вершин в начале координат (когда  $x_1^0=0, x_2^0=0$ ) определяется  $(2*3)$  матрицей

$$\mathbf{R}_{2 \times 3} = \begin{pmatrix} 0 & r_1 & r_2 \\ 0 & r_2 & r_1 \end{pmatrix}$$

и имеет вид, представленный на рис. 1.

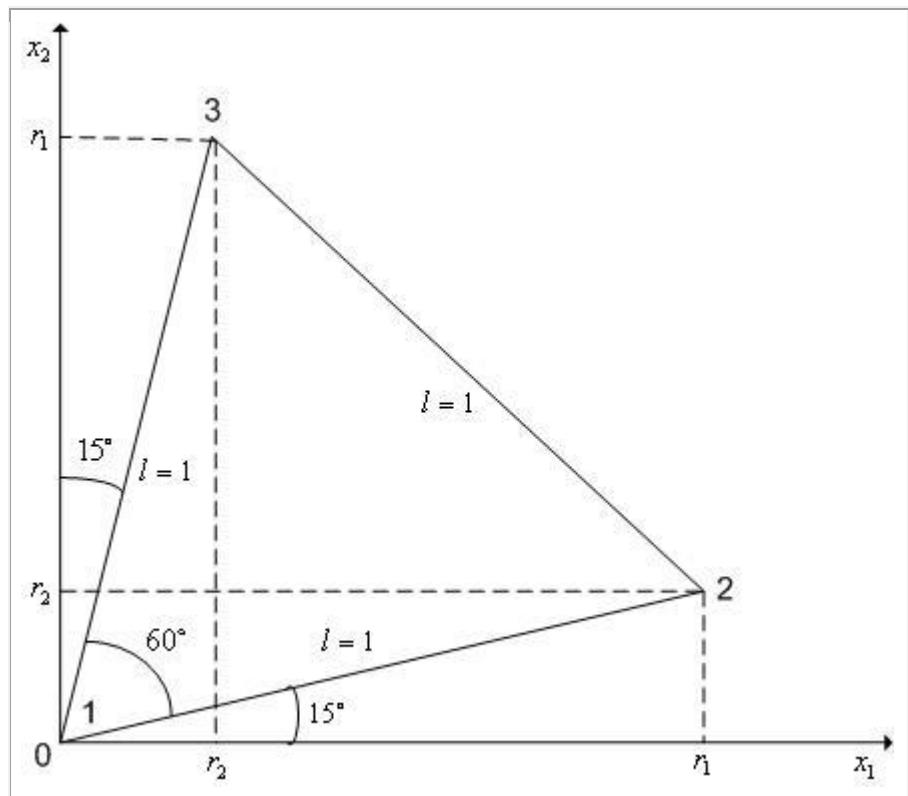


Рис. 1. Регулярный симплекс в пространстве  $R^2$  с одной из вершин в начале координат. В алгоритме симплекс-метода используется следующее важное свойство регулярного симплекса: если одну из вершин регулярного симплекса перенести на надлежащее расстояние вдоль прямой, соединяющей данную вершину и центр тяжести оставшихся вершин, то вновь получится регулярный симплекс (см. рис. 2). Будем называть эту процедуру отражением вершины симплекса относительно центра тяжести остальных вершин.

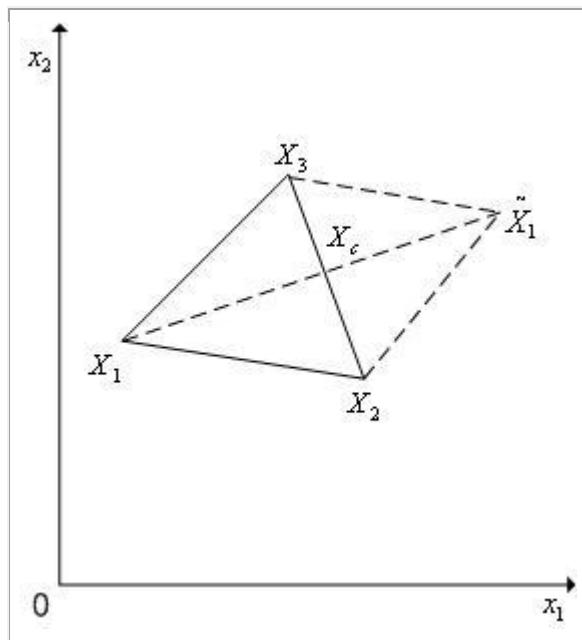


Рис. 2. Отражение вершины  $X_1$  регулярного симплекса в пространстве  $R^2$  относительно центра тяжести  $X_c$  остальных вершин.

Пусть  $\mathbf{X}_i^r, i \in [1, n+1]$  - векторы координат вершин регулярного симплекса. Тогда при выполнении операции отражения  $k$ -й вершины симплекса имеет место следующая связь координат этой вершины и новой вершины:

$$\frac{\mathbf{X}_k^{r+1} + \mathbf{X}_k^r}{2} = \mathbf{X}_c^r. \quad (4)$$

Здесь

$$\mathbf{X}_c^r = \frac{1}{n} \sum_{i=1, i \neq k}^{n+1} \mathbf{X}_i^r - \quad (5)$$

вектор координат центра тяжести остальных вершин симплекса (за исключением отраженной вершины  $k$ )

Таким образом, после отражения  $k$ -й вершины симплекса с координатами вершин  $\mathbf{X}_i^r, i \in [1, n+1]$ , получаем новый симплекс с координатами вершин

$$\mathbf{X}_i^{r+1} = \mathbf{X}_i^r, i \in [1, n+1], i \neq k, \mathbf{X}_k^{r+1} = 2\mathbf{X}_c^r - \mathbf{X}_k^r \quad (6)$$

Кроме операции отражения вершины симплекса симплекс-метод может использовать операцию редукции симплекса (см. рис. 3) - уменьшение длин всех ребер симплекса на одну и ту же величину.

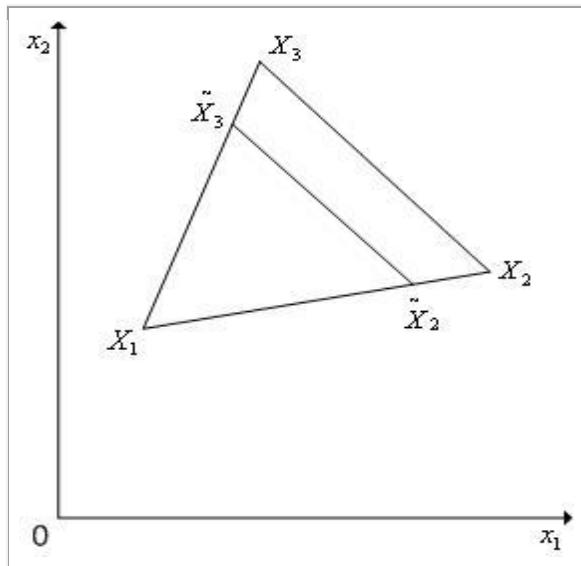


Рис. 3. Редукция вершин регулярного симплекса в пространстве R2 к вершине X1.

Пусть  $\mathbf{X}_i^r, i \in [1, n+1]$  - векторы координат вершин регулярного симплекса. Тогда при выполнении операции редукции вершин этого симплекса к вершине  $\mathbf{X}_k$  новые координаты остальных вершин симплекса определяются по формуле

$$\mathbf{X}_i^{r+1} = \mathbf{X}_k^r + \gamma (\mathbf{X}_i^r - \mathbf{X}_k^r), i \in [1, n+1], i \neq k,$$

где  $\gamma \in (0, 1)$  - коэффициент редукции. Рекомендуется использовать  $\gamma = 0.5$ .

Таким образом, после редукции вершин симплекса  $\mathbf{X}_i^r, i \in [1, n+1]$  к вершине  $\mathbf{X}_k$  получаем новый симплекс с координатами вершин

$$\mathbf{X}_i^{r+1} = \mathbf{X}_k^r + \gamma (\mathbf{X}_i^r - \mathbf{X}_k^r), i \in [1, n+1], i \neq k, \mathbf{X}_k^{r+1} = \mathbf{X}_k^r \quad (7)$$

Схема простейшего варианта симплекс-метода.

Суть симплекс-метода раскрывает его простейший вариант.

Задаем начальную точку  $\mathbf{X}^0$ , длину ребра симплекса  $l$  и полагаем  $r=0$ .

По формулам (2), (3) находим координаты  $\mathbf{X}_i^r, i \in [1, n+1]$  всех вершин симплекса.

Вычисляем значения  $\Phi(\mathbf{X}_i^r), i \in [1, n+1]$  минимизируемой функции во всех вершинах симплекса.

Находим максимальное из значений функции  $\Phi(\mathbf{X})$  в вершинах симплекса  $\mathbf{X}_i^r, i \in [1, n+1]$ :

$$\max(\Phi(\mathbf{X}_1^r), \Phi(\mathbf{X}_2^r), \dots, \Phi(\mathbf{X}_{n+1}^r)) = \Phi(\mathbf{X}_k^r).$$

По формулам (5), (6) отражаем вершину  $\mathbf{X}_k^r$  относительно центра тяжести остальных вершин симплекса – получаем новый симплекс с координатами вершин  $\mathbf{X}_i^{r+1}, i \in [1, n+1]$ .

Вычисляем значение  $\Phi(\mathbf{X}_k^{r+1})$  минимизируемой функции в новой вершине симплекса.

Если условие окончания итераций (см. ниже) выполнено, то в качестве приближенного значения точки минимума функции  $\Phi(\mathbf{X})$  принимаем ту вершину симплекса  $\mathbf{X}_i^{r+1} = \mathbf{X}_i^r, i \in [1, n+1]$ , в которой  $\Phi(\mathbf{X})$  имеет минимальное значение, и заканчиваем вычисления. Иначе полагаем  $r=r+1$  и переходим к п. 4 •

Поскольку размер симплекса в простейшем варианте симплекс-метода фиксирован, в качестве условия окончания итераций в данном случае можно использовать условие

$$\max_{i \in [1, n+1], i \neq k} |\Phi(\mathbf{X}_i^{r+1}) - \Phi(\mathbf{X}_k^{r+1})| \leq \varepsilon_\Phi, \quad (8)$$

где  $\varepsilon_\Phi$  – требуемая точность решения по  $\Phi$ ,  $k \in [1, n+1]$  – номер произвольной вершины симплекса. Отметим, что выражение в левой части неравенства (8) есть максимальная

разность значений функции  $\Phi(\mathbf{X})$  в двух вершинах симплекса  $\mathbf{X}_i^{r+1} = \mathbf{X}_i^r, i \in [1, n+1]$ .

Простейший вариант симплекс-метода иллюстрирует рис. 4, на котором показаны линии уровня функции Химмельблау ( $n=2$ ).

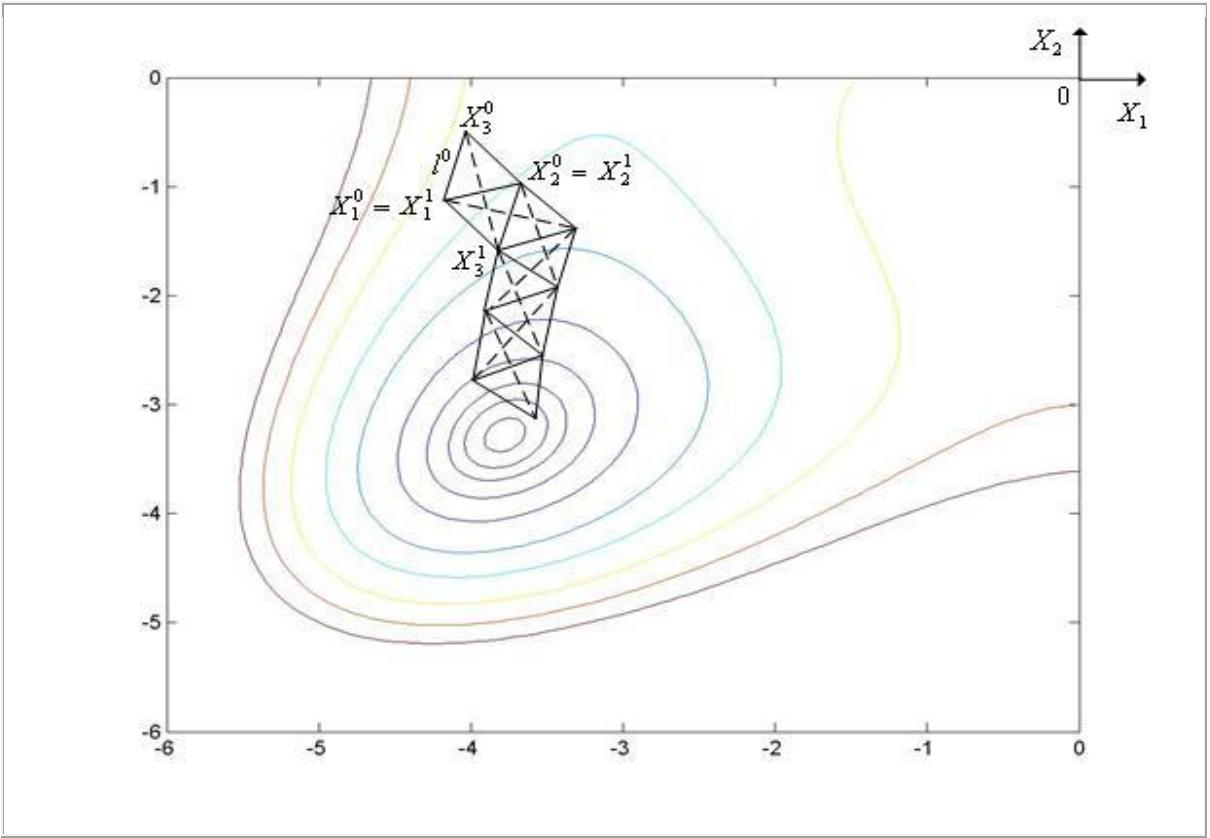


Рис. 4. Траектория поиска минимума функции Химмельблау простейшим симплекс-методом.

#### Модифицированный симплекс-метод.

Рассмотренный простейший симплекс-метод склонен к зацикливанию и медленно сходится, если длина ребра симплекса  $\ell$  выбрана малой (выбор же большой длины ребра симплекса обеспечивает высокую скорость сходимости, но дает малую точность решения). Поэтому в вычислительной практике используются различные модификации простейшего метода, направленные на преодоление его указанных недостатков.

Основной идеей модифицированного симплекс-метода является изменение по некоторому правилу размера симплекса в процессе поиска. При этом наряду с условием (8) в качестве условия окончания итераций можно использовать условие

$$\ell^r \leq \varepsilon_x, \quad (9)$$

где  $\ell^r$  - текущая длина ребра симплекса,  $\varepsilon_x$  - требуемая точность решения по  $X$ .

Обычно размер симплекса изменяется при выполнении следующих условий:

при «накрытии» симплексом дна оврага или точки минимума;

при циклическом движении.

«Накрытие» симплексом дна оврага или точки минимума. Пусть  $\mathbf{X}_k^{r+1}$  - вершина, которая получилась на  $r$ -ой итерации в результате отражения вершины  $\mathbf{X}_k^r$ . Так что координаты вершин нового симплекса равны  $\mathbf{X}_i^{r+1} = \mathbf{X}_i^r, i \in [1, n+1], i \neq k, \mathbf{X}_k^{r+1}$ . Ситуация  $\Phi(\mathbf{X}_k^{r+1}) > \Phi(\mathbf{X}_k^r)$  интерпретируется как «накрытие» этим симплексом дна оврага или точки минимума и простейший симплекс-метод модифицируется следующим образом (см. рис. 5):

Полагаем  $k = k + 1$  (если  $k = n + 2$ , то полагаем  $k = 1$ );

Выполняем отражение  $k$ -ой вершины симплекса  $\mathbf{X}_i^r, i \in [1, n+1]$ ;

Если  $\Phi(\mathbf{X}_k^{r+1}) > \Phi(\mathbf{X}_k^r)$  и не все вершины перебраны, то переходим к п.1.  
Иначе - продолжаем итерации по схеме простейшего симплекс-метода •

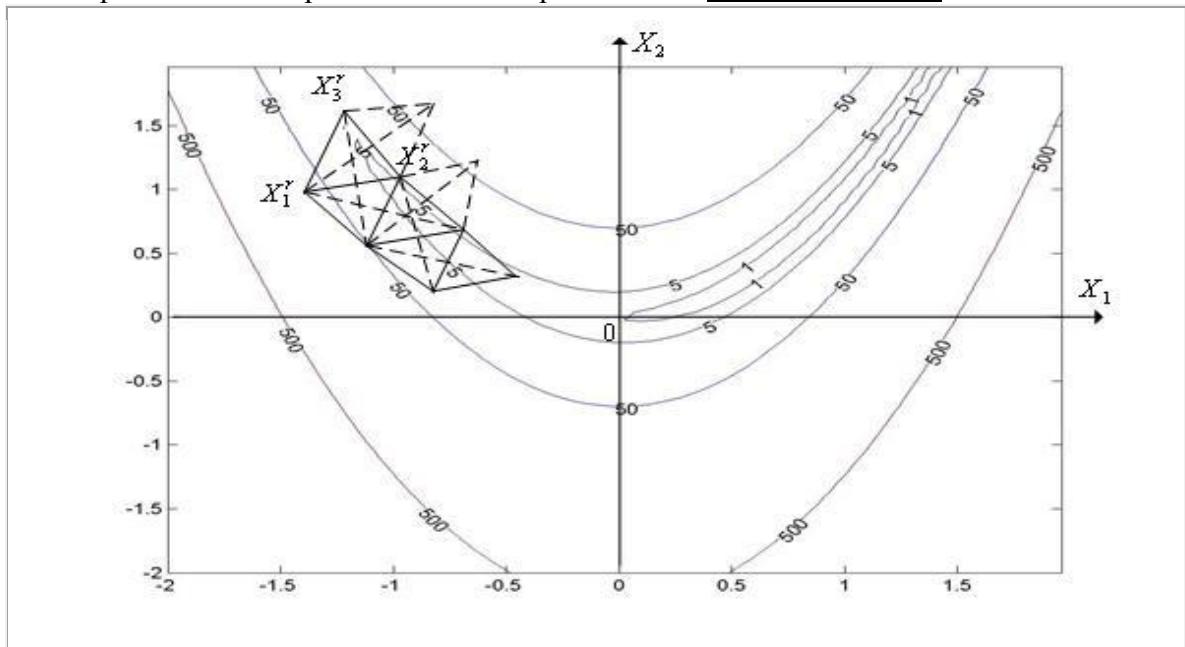


Рис. 5. Траектория поиска минимума функции Розенброка модифицированным симплекс-методом при «накрытии» дна оврага. Пунктиром показаны отвергнутые симплексы.

Циклическое движение. Ситуация, когда некоторая вершина симплекса не исключается на протяжении  $m$  итераций, интерпретируется как «зацикливание» алгоритма. Простейший симплекс-метод модифицируется в этом случае следующим образом:

Находим вершину текущего симплекса  $\mathbf{X}_i^r, i \in [1, n+1]$ , в которой функция  $\Phi(\mathbf{X})$  принимает наименьшее значение  
 $\min(\Phi(\mathbf{X}_1^0), \Phi(\mathbf{X}_2^r), \dots, \Phi(\mathbf{X}_{n+1}^r)) = \Phi(\mathbf{X}_k^r)$ .

По формуле (7) выполняем редукцию симплекса  $\mathbf{X}_i^r, i \in [1, n+1]$  к вершине  $\mathbf{X}_k^r$ .  
Продолжаем итерации по схеме простейшего симплекс-метода •

Здесь количество итераций  $m$  рекомендуется находить из условия  $m = \lceil 1.65n + 0.05n^2 \rceil$ ,  
где  $\lceil *\rceil$  - символ ближайшего целого большего.

## 6.6. Метод деформируемого многогранника (Нелдера-Мида)

Рассмотрим следующую задачу локальной безусловной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного в  $n$ -мерном евклидовом пространстве  $R^n$ ,

$$\min_{\mathbf{X} \in R^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*. \quad (1)$$

В случае если функция  $\Phi(\mathbf{X})$  является овражной функцией, эффективность симплекс-метода при решении задачи (1) значительно снижается в силу того, что регулярный симплекс нельзя «вытянуть» вдоль оврага. Метод Нелдера-Мида (метод деформируемого многогранника) является развитием симплекс-метода и использует в процессе поиска деформацию (изменение размеров и формы) текущего симплекса (не обязательно регулярного).

Метод использует следующие операции над симплексами:

отражение;

редукция;

сжатие;

растяжение.

Отражение вершины симплекса относительно центра тяжести остальных вершин (см. параграф 5, рис. 1). В результате отражения  $k$ -й вершины симплекса с координатами вершин  $\mathbf{X}_i^r, i \in [1, n+1]$ , получаем новый симплекс с координатами вершин

$$\mathbf{X}_i^{r+1} = \mathbf{X}_i^r, i \in [1, n+1], i \neq k, \mathbf{X}_k^{r+1} = 2\mathbf{X}_c^r - \mathbf{X}_k^r, \quad (2)$$

где

$$\mathbf{X}_c^r = \frac{1}{n} \sum_{i=1, i \neq k}^{n+1} \mathbf{X}_i^r. \quad (3)$$

вектор координат центра тяжести остальных вершин симплекса (за исключением отраженной вершины  $k$ ).

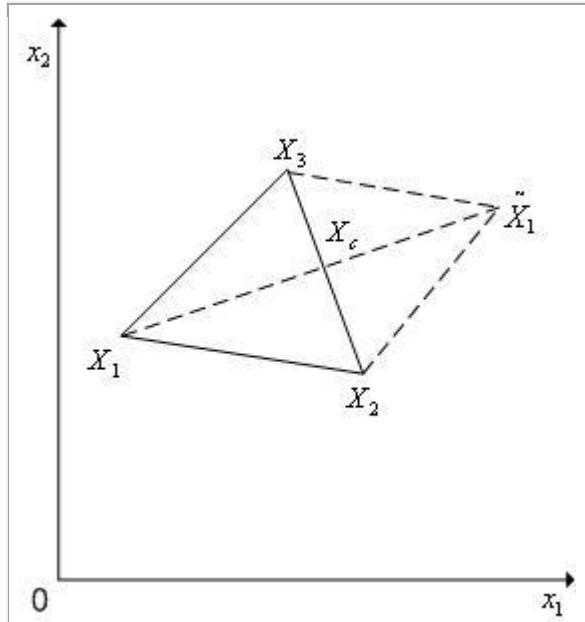


Рис. 1. Отражение вершины  $X_1$  симплекса в пространстве  $R^2$  относительно центра тяжести  $X_c$  остальных вершин.

Редукции симплекса (см. параграф 5, рис. 2) - уменьшение длины всех ребер симплекса в одно и то же количество раз. В результате выполнения редукции вершине симплекса  $\mathbf{X}_i^r, i \in [1, n+1]$  к вершине  $\mathbf{X}_k$  получаем новый симплекс с координатами вершин

$$\mathbf{X}_i^{r+1} = \mathbf{X}_k^r + \gamma (\mathbf{X}_i^r - \mathbf{X}_k^r), \quad i \in [1, n+1], \quad i \neq k, \quad \mathbf{X}_k^{r+1} = \mathbf{X}_k^r \quad (4)$$

где  $\gamma \in (0, 1)$ ,  $\gamma \approx 0.5$  - коэффициент редукции.

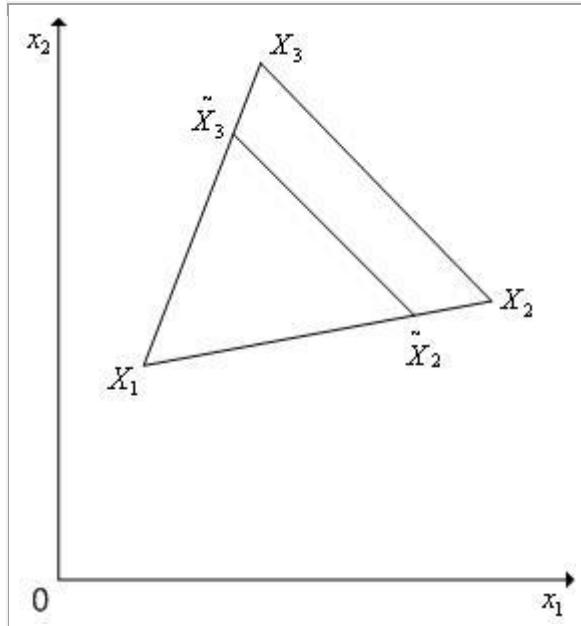


Рис. 2. Редукция вершин симплекса в пространстве R2 к вершине X1.

Сжатие симплекса (см. рис. 3). В результате выполнения сжатия симплекса  $\mathbf{X}_i^r, i \in [1, n+1]$  в направлении  $(\mathbf{X}_k^r - \mathbf{X}_c^r)$  получаем новый симплекс с координатами вершин

$$\mathbf{X}_i^{r+1} = \mathbf{X}_i^r, \quad i \in [1, n+1], \quad i \neq k, \quad \mathbf{X}_k^{r+1} = \mathbf{X}_c^r + \beta (\mathbf{X}_k^r - \mathbf{X}_c^r), \quad (5)$$

где  $\beta \in (0, 1)$ ,  $\beta \approx 0.4-0.6$  - коэффициент сжатия,  $\mathbf{X}_c^r$  - вектор координат центра тяжести остальных вершин симплекса (см. (3)).

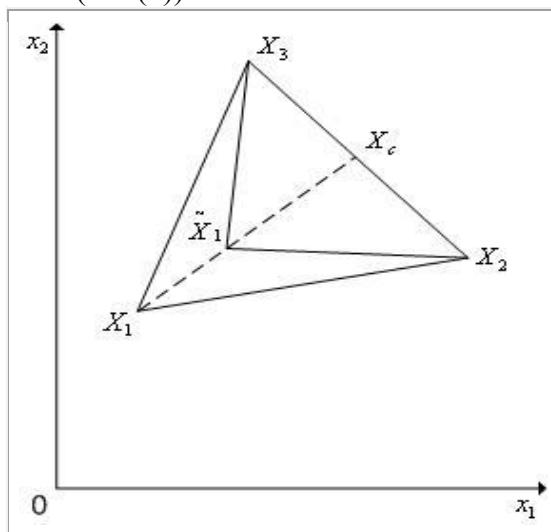


Рис. 3. Сжатие симплекса в пространстве R2 в направлении \$(X\_1-X\_c)\$.

Растяжение симплекса (см. рис. 4). В результате выполнения растяжения симплекса  $\mathbf{X}_i^r, i \in [1, n+1]$  в направлении  $(\mathbf{X}_k^r - \mathbf{X}_c^r)$  получаем новый симплекс с координатами вершин

$$\mathbf{X}_i^{r+1} = \mathbf{X}_i^r, i \in [1, n+1], i \neq k, \mathbf{X}_k^{r+1} = \mathbf{X}_c^r + \alpha(\mathbf{X}_k^r - \mathbf{X}_c^r), \quad (6)$$

где  $\alpha \approx 2.8-3.0$  - коэффициент растяжения,  $\mathbf{X}_c^r$  - вектор координат центра тяжести остальных вершин симплекса (см. (3)).

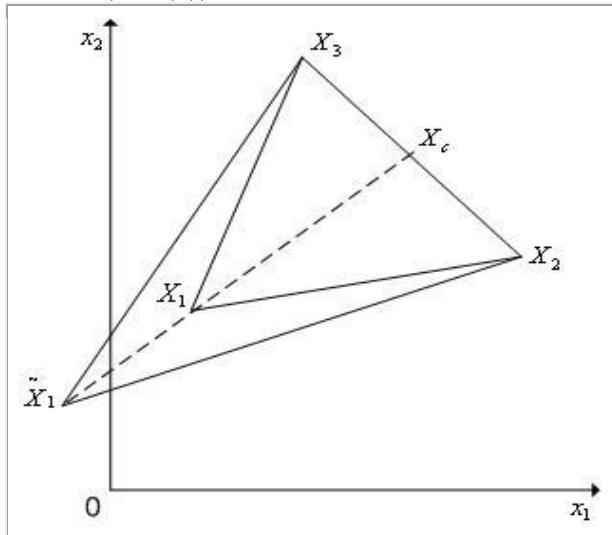


Рис. 4. Растижение симплекса в пространстве R2 в направлении (X1-Xc). Схема метода Нелдера-Мида.

Симплекс с вершинами  $\mathbf{X}_i^r, i \in [1, n+1]$  обозначим  $S^r$ .

Задаем начальную точку  $\mathbf{X}^0$ , длину ребра симплекса  $l$  и полагаем  $r=0$ .

Находим координаты  $\mathbf{X}_i^0, i \in [1, n+1]$  всех вершин регулярного симплекса  $S^0$  с длиной ребра  $l$ . Вычисляем значения  $\Phi(\mathbf{X}_i^r)$  минимизируемой функции во всех вершинах симплекса.

Среди вершин симплекса  $S^r$  находим вершины  $\mathbf{X}_{l=k_l}^r, \mathbf{X}_{h=k_h}^r, \mathbf{X}_{s=k_s}^r$ , в которых функция  $\Phi(\mathbf{X})$  принимает, соответственно, наименьшее, наибольшее и следующее за максимальным значения, а также находим значения функции  $\Phi(\mathbf{X})$  в этих точках:

$$\Phi(\mathbf{X}_l^r) = \min_{i \in [1, n+1]} \Phi(\mathbf{X}_i^r),$$

$$\Phi(\mathbf{X}_h^r) = \max_{i \in [1, n+1]} \Phi(\mathbf{X}_i^r),$$

$$\Phi(\mathbf{X}_s^r) = \max_{i \in [1, n+1], i \neq h} \Phi(\mathbf{X}_i^r).$$

По формулам (2), (3) выполняем отражение вершину симплекса  $\mathbf{X}_h^r$  относительно центра тяжести остальных вершин симплекса – получаем новый симплекс  $S^{r+1}$ . Вычисляем значение  $\Phi(\mathbf{X}_{k_h}^{r+1})$  минимизируемой функции в новой вершине симплекса.

Если условие окончания итераций (см. ниже) выполнено, то в качестве приближенного значения точки минимума функции  $\Phi(\mathbf{X})$  принимаем ту вершину симплекса  $S^{r+1}$ , в которой  $\Phi(\mathbf{X})$  имеет минимальное значение, и заканчиваем вычисления.

Если  $\Phi(\mathbf{X}_{k_h}^{r+1}) \leq \Phi(\mathbf{X}_h^r)$  и  $\Phi(\mathbf{X}_{k_h}^{r+1}) \leq \Phi(\mathbf{X}_l^r)$ , то переходим к п.7 (растяжению симплекса) - см. рис. 5. Если  $\Phi(\mathbf{X}_{k_h}^{r+1}) \leq \Phi(\mathbf{X}_h^r)$ , но  $\Phi(\mathbf{X}_{k_h}^{r+1}) > \Phi(\mathbf{X}_l^r)$ , то переходим к п.3 (отражению)

– см. рис. 6. Если  $\Phi(\mathbf{X}_{kh}^{r+1}) \geq \Phi(\mathbf{X}_h^r)$ , то переходим к п.8 (сжатию симплекса) – см. рис. 7, рис. 8.

Ситуация  $\Phi(\mathbf{X}_{kh}^{r+1}) \leq \Phi(\mathbf{X}_h^r)$  и  $\Phi(\mathbf{X}_{kh}^{r+1}) \leq \Phi(\mathbf{X}_l^r)$ . По формуле (6) выполняем растяжение симплекса  $S^{r+1}$  в направлении  $(\mathbf{X}_{kh}^{r+1} - \mathbf{X}_c^{r+1})$ . получаем новый симплекс  $S^{r+2}$ .

Вычисляем значение минимизируемой функции в новой вершине симплекса  $\mathbf{X}_{kh}^{r+2}$ . Если  $\Phi(\mathbf{X}_{kh}^{r+2}) \leq \Phi(\mathbf{X}_l^r)$ , то полагаем  $r=r+2$  и переходим к п.3 (отражению вершины симплекса). Иначе полагаем  $r=r+1$  и переходим к п.3 (отражению вершины симплекса) с симплексом  $S^r$  (т.е. не используем результаты растяжения).

Ситуация  $\Phi(\mathbf{X}_{kh}^{r+1}) \geq \Phi(\mathbf{X}_h^r)$ . По формуле (5) выполняем сжатие симплекса  $S^{r+1}$  в направлении  $(\mathbf{X}_{kh}^{r+1} - \mathbf{X}_c^{r+1})$ . получаем новый симплекс  $S^{r+2}$ . Вычисляем значение минимизируемой функции в новой вершине симплекса  $\mathbf{X}_{kh}^{r+2}$ . Если  $\Phi(\mathbf{X}_{kh}^{r+2}) \leq \Phi(\mathbf{X}_{kh}^r)$ , то полагаем  $r=r+2$  и переходим к п.3 (отражению вершины симплекса). Иначе по формуле (4) выполняем редукцию симплекса  $S^r$  к вершине  $\mathbf{X}_{l=k_l}^r$  – получаем новый симплекс  $S^{r+1}$ . Вычисляем значение минимизируемой функции во всех новых вершинах симплекса  $S^{r+1}$ . Полагаем  $r=r+1$  и переходим к п.3 (отражению симплекса) •

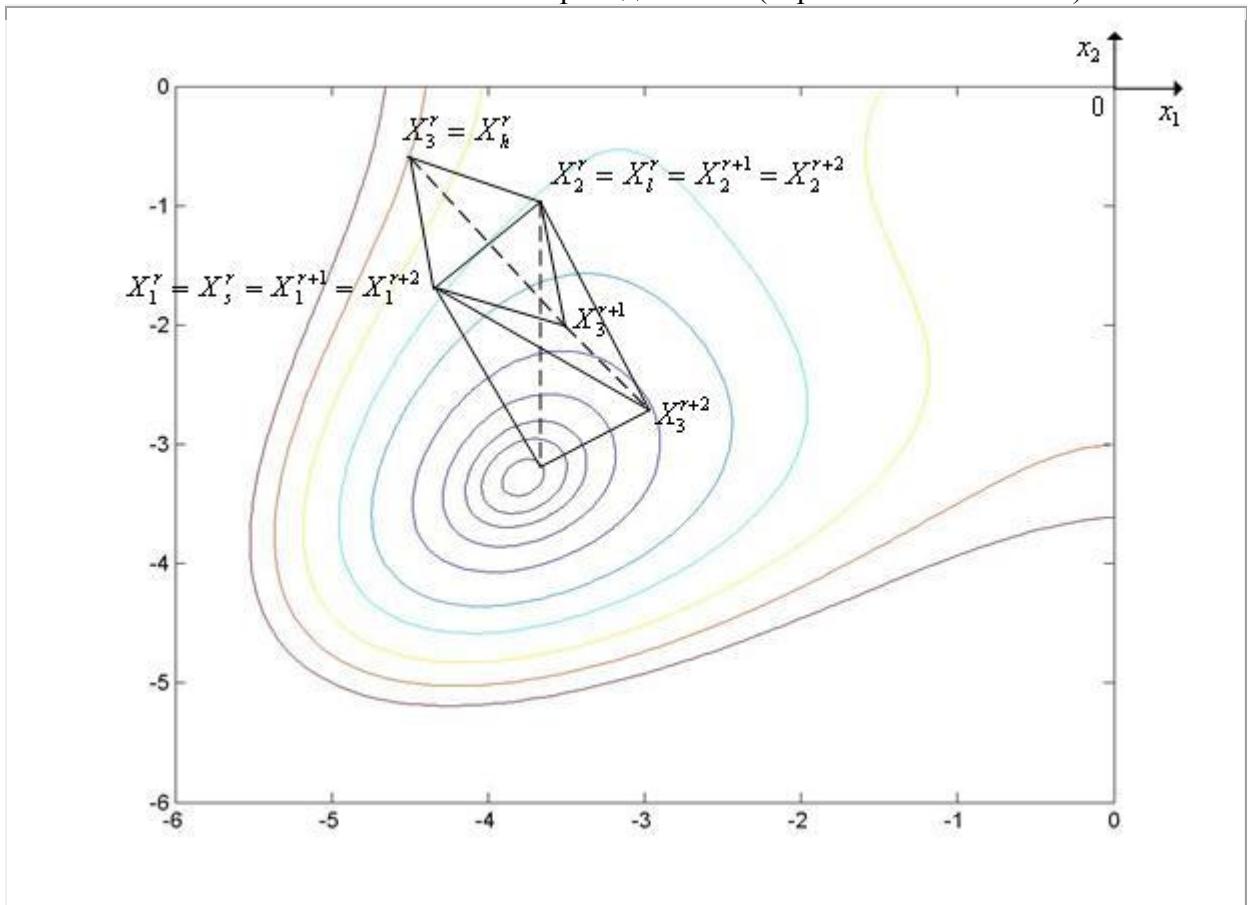


Рис. 5. Ситуация, когда метод Нелдера-Мида использует успешное растяжение симплекса.

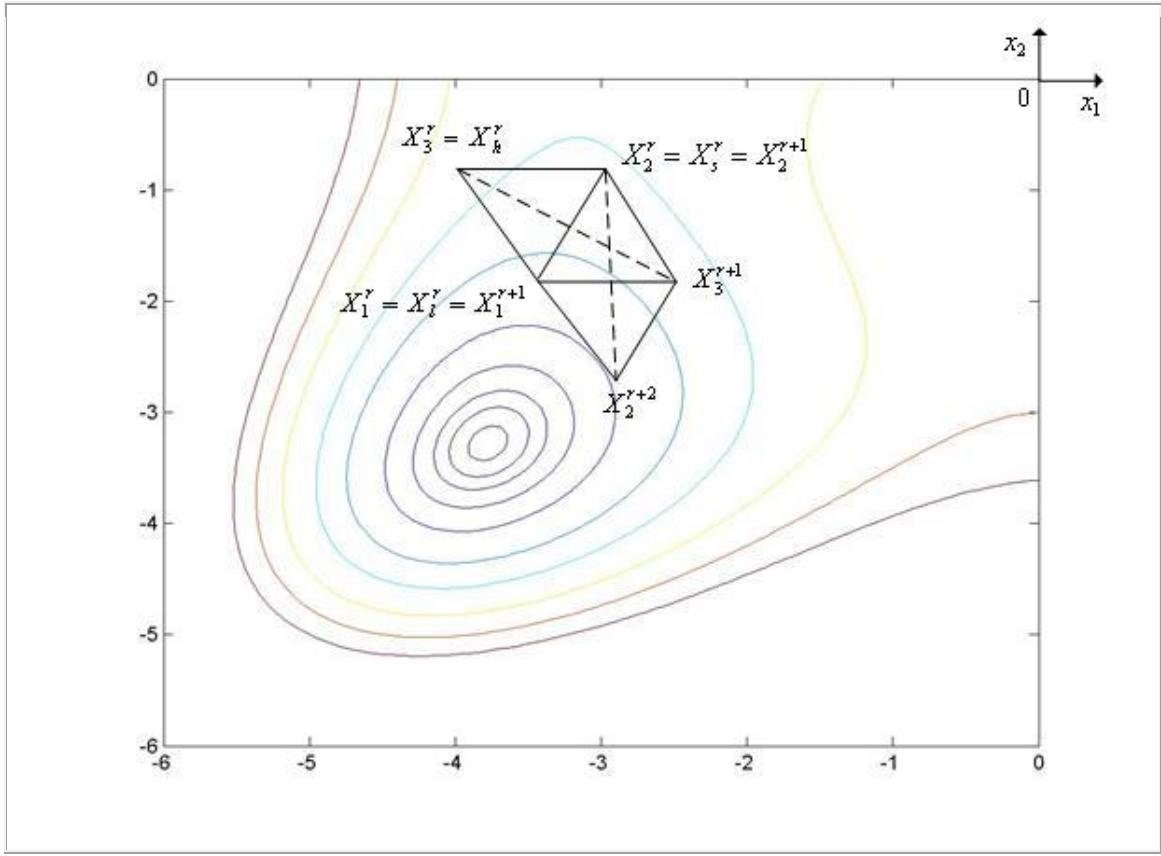


Рис. 6. Ситуация, когда метод Нелдера-Мида использует успешное отражение симплекс.

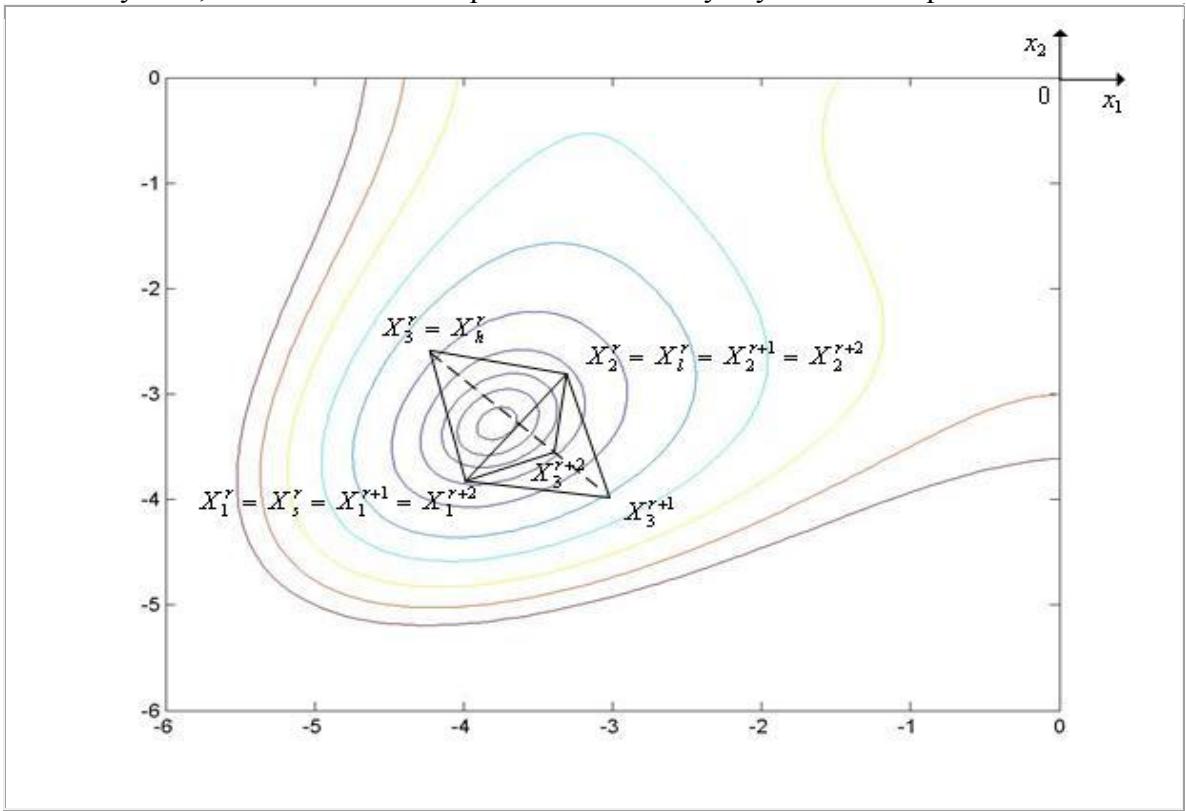


Рис. 7. Ситуация, когда метод Нелдера-Мида использует успешное сжатие симплекс.

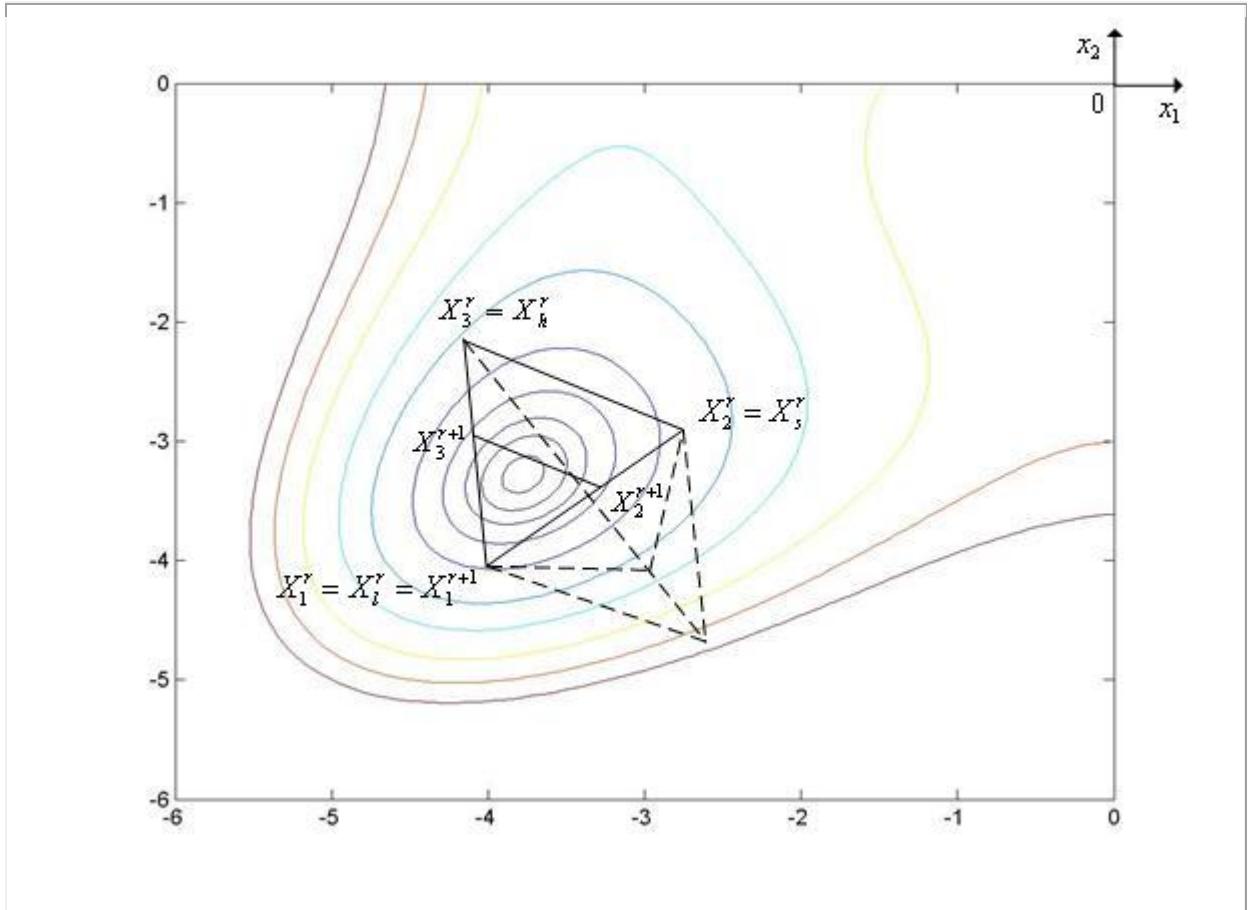


Рис. 8. Ситуация, когда метод Нелдера-Мида использует редукцию после неудачного сжатия симплекса. Пунктиром показаны отвергнутые (неудачные) итерации.

На рис. 5 - рис. 8 минимизируемой функцией  $\Phi(\mathbf{X})$  является функция Химмельблау. В качестве условия окончания итераций в методе Нелдера-Мида можно использовать условие

$$\max_{i \in [1, n+1], i \neq k} |\Phi(\mathbf{X}_i^{r+1}) - \Phi(\mathbf{X}_k^{r+1})| \leq \varepsilon_\Phi,$$

где  $\varepsilon_\Phi$  - требуемая точность решения по  $\Phi$ ,  $k \in [1, n+1]$  - номер произвольной вершины симплекса. Можно также завершать итерации, когда длина максимального из ребер

текущего симплекса станет меньше или равна  $\varepsilon_X$  - требуемой точности решения по  $\mathbf{X}$ . Изложенная схема метода Нелдера-Мида имеет тот недостаток, что для сильно овражных функций может происходить вырождение («сплющивание») симплекса. Поэтому к рассмотренной схеме метода Нелдера-Мида добавляется этап периодического (через  $N$  итераций) восстановления симплекса, который заключается в следующем:

в текущем симплексе выбираются две «лучшие» вершины и определяется расстояние

между ними  $\tilde{l}$ ;

исходя из «лучшей» вершины текущего симплекса строится новый симплекс, длина ребра которого принимается равной  $\tilde{l}$  (см. параграф 5).

## 7.1. Метод наискорейшего спуска. Метод дробления шага

Рассматривается следующая многомерная задача локальной безусловной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного в  $n$ -мерном евклидовом пространстве  $R^n$ ,

$$\min_{\mathbf{X} \in R^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*. \quad (1)$$

Положим, что функция  $\Phi(\mathbf{X})$  всюду дифференцируема в  $n$ -мерном евклидовом пространстве  $R^n$ .

Направление спуска в градиентных методах оптимизации совпадает с направлением антиградиента минимизируемой функции  $\Phi(\mathbf{X})$ . Итерационная формула градиентных методов оптимизации имеет вид

$$\mathbf{X}^{r+1} = \mathbf{X}^r + \lambda^r \mathbf{S}^r. \quad (2)$$

Здесь  $\lambda^r$  - длина шага на  $r$ -ой итерации в направлении  $\mathbf{S}^r$ , где

$$\mathbf{S}^r = -\frac{\nabla \Phi^r}{\|\nabla \Phi^r\|}, \quad (3)$$

единичный вектор направления антиградиента функции  $\Phi(\mathbf{X})$  в точке  $\mathbf{X}^r$ ,  $\|\cdot\|$  - некоторая векторная норма, например, евклидова. Напомним, что градиент функции  $\Phi(\mathbf{X})$  в точке  $\mathbf{X}^r$  есть значение вектора частных производных этой функции в точке  $\mathbf{X}^r$ :

$$\nabla \Phi^r = \nabla \Phi(\mathbf{X}^r) = \left( \begin{array}{c} \frac{\partial \Phi(\mathbf{X})}{\partial x_1} \\ \frac{\partial \Phi(\mathbf{X})}{\partial x_2} \\ \vdots \\ \frac{\partial \Phi(\mathbf{X})}{\partial x_n} \end{array} \right) \Bigg|_{\mathbf{X}=\mathbf{X}^r}$$

Различные градиентные методы оптимизации отличаются между собой правилами выбора длины шага  $\lambda^r$ .

Градиентный метод наискорейшего спуска.

Градиентный метод наискорейшего спуска в качестве длины шага  $\lambda^r$  использует величину, при которой достигается минимум функции  $\Phi(\mathbf{X})$  в направлении  $\mathbf{S}^r$ :

$$\Phi(\mathbf{X}^{r+1}) = \Phi(\mathbf{X}^r + \lambda^r \mathbf{S}^r) = \min_{\lambda \in R^1} \Phi(\mathbf{X}^r + \lambda \mathbf{S}^r). \quad (4)$$

Задача (2) есть одномерная задача локальной безусловной оптимизации, которая может быть решена рассмотренными в главе 4 методами, например, методом Паулла (см. параграф 4.7).

Схема метода:

Задаем начальную точку  $\mathbf{X}^0$  и полагаем счетчик числа итераций  $r=0$ .

По формуле (3) вычисляем компоненты вектора  $\mathbf{S}^r$ .

Каким-либо методом решаем одномерную задачу безусловной оптимизации (4) – определяем точку  $\mathbf{X}^{r+1}$ .

Вычисляем величину  $\Phi(\mathbf{X}^{r+1})$  - значение функции  $\Phi(\mathbf{X})$  в точке  $\mathbf{X}^{r+1}$ .

Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем  $\mathbf{X}^* \approx \mathbf{X}^{r+1}$  и завершаем итерации. Иначе – полагаем  $r=r+1$ , переходим к п. 2 ●

В качестве критерия окончания поиска можно использовать одно из стандартных условий окончания итераций

$$\|\mathbf{X}^{r+1} - \mathbf{X}^r\| = \lambda^r \leq \varepsilon_{\mathbf{X}}, \quad (5)$$

$$|\Phi(\mathbf{X}^{r+1}) - \Phi(\mathbf{X}^r)| \leq \varepsilon_{\Phi}. \quad (6)$$

В качестве критерия окончания поиска можно использовать также условие

$$\|\nabla \Phi^r\| \leq \varepsilon_{\nabla}, \quad (7)$$

где  $\varepsilon_{\nabla}$  - константа, определяющая требуемую точность решения по градиенту функции  $\Phi(\mathbf{X})$ .

Градиентный метод наискорейшего спуска иллюстрирует рис. 1, на котором показан фрагмент линий уровня функции Химмельблау.

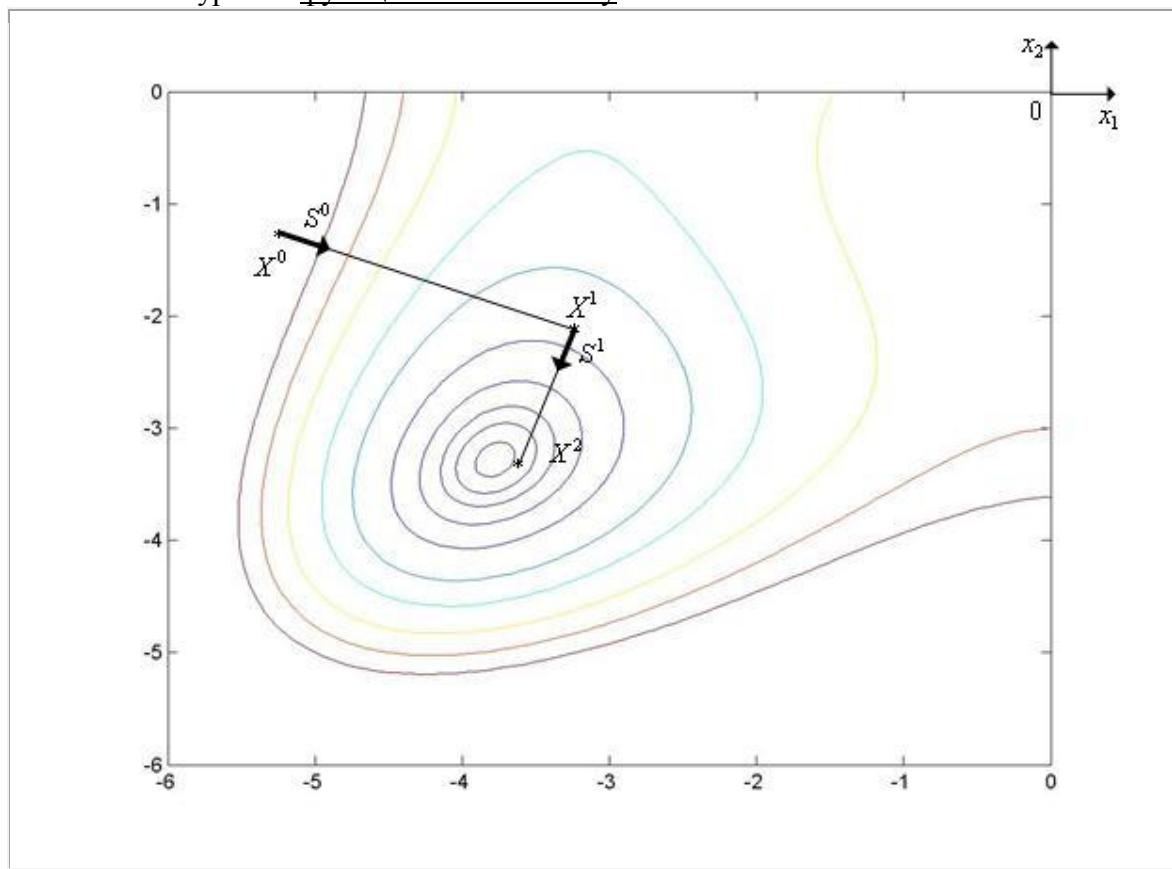


Рис. 1. Траектория поиска минимума функции Химмельблау градиентным методом наискорейшего спуска.

Градиентный метод с дроблением шага

В градиентном методе с дроблением шага точка  $\mathbf{X}^{r+1}$  определяется по формуле

$$\mathbf{X}^{r+1} = \mathbf{X}^r + \lambda^r \mathbf{S}^r, \quad (8)$$

где величина шага  $\lambda^r$  находится из условия

$$\Phi(\mathbf{X}^r) - \Phi(\mathbf{X}^{r+1}) \geq 0.5 \lambda^r \|\nabla \Phi^r\|. \quad (9)$$

Схема метода:

Задаем начальную точку  $\mathbf{X}^0$ , начальную величину шага  $\lambda^0$  и коэффициент дробления шага  $v \in (0, 1]$ . Полагаем счетчик числа итераций  $r=0$ .

По формуле (8) вычисляем компоненты вектора  $\mathbf{X}^{r+1}$ .

Вычисляем величину  $\Phi(\mathbf{X}^{r+1})$  – значение функции  $\Phi(\mathbf{X})$  в точке  $\mathbf{X}^{r+1}$ .

Если условие (9) выполнено, то переходим к следующему пункту. Иначе – переходим к пункту 6.

Полагаем  $\lambda^r = v\lambda^r$  и переходим к пункту 2.

Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем  $\mathbf{X}^* \approx \mathbf{X}^{r+1}$ , и завершаем итерации. Иначе – полагаем  $r=r+1$  и переходим к п. 2.

В качестве критерия окончания поиска можно использовать условия (5) – (7).

Градиентный метод дробления шага иллюстрирует рис. 2, на котором показан фрагмент линий уровня функции Химмельблау.

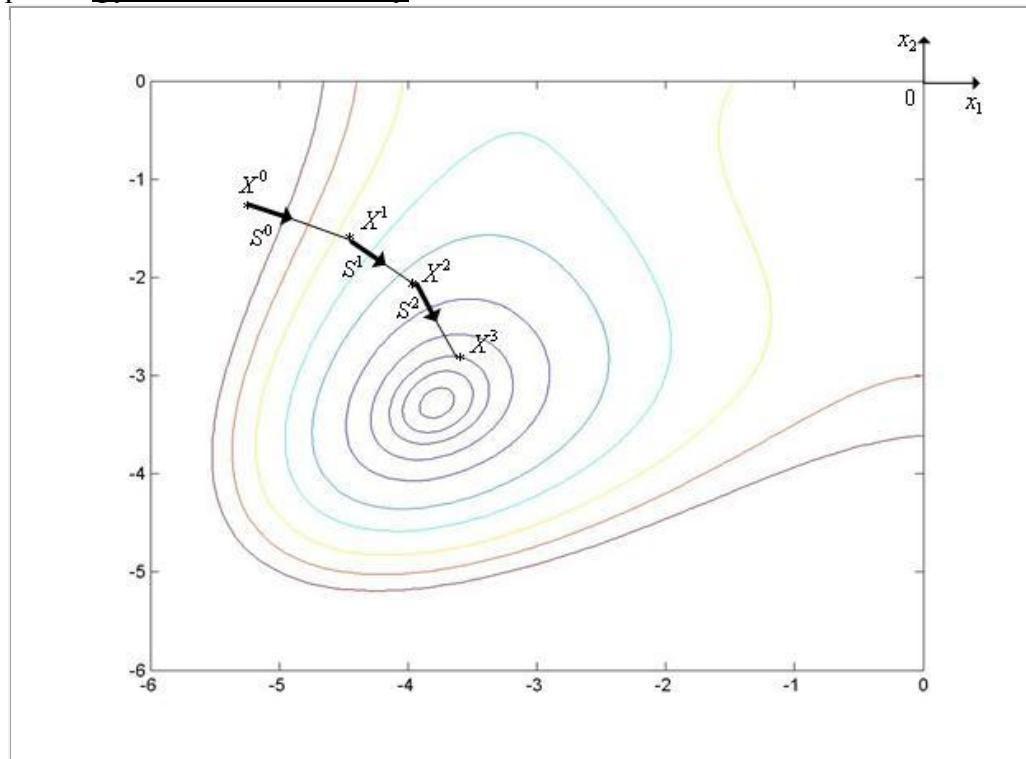


Рис. 2. Траектория поиска минимума функции Химмельблау градиентным методом дробления шага.

### Пример 1

1. Постановка задачи.

Выполнить несколько итераций (не менее двух) решения двумерной задачи локальной безусловной оптимизации

$$\min_{\mathbf{X} \in \mathbb{R}^2} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (10)$$

$$\text{где } \Phi(\mathbf{X}) = \Phi(x, y) = x^2 + y^2 + 3(x+y)^2 = 4x^2 + 4y^2 + 6xy, \quad (11)$$

градиентным методом с дроблением шага, исходя из точки  $\mathbf{X}^0 = (x^0, y^0) = (-2.0, 1.0)$ .

Принять  $\lambda^0 = 1.0$ ,  $v = 0.7$ , в качестве нормы вектора градиента использовать евклидову норму.

Траекторию поиска изобразить на рис. 3, на котором приведены линии уровня квадратичной функции (11), полученные с помощью MATLAB-программы, приведенной в параграфе 6.1.

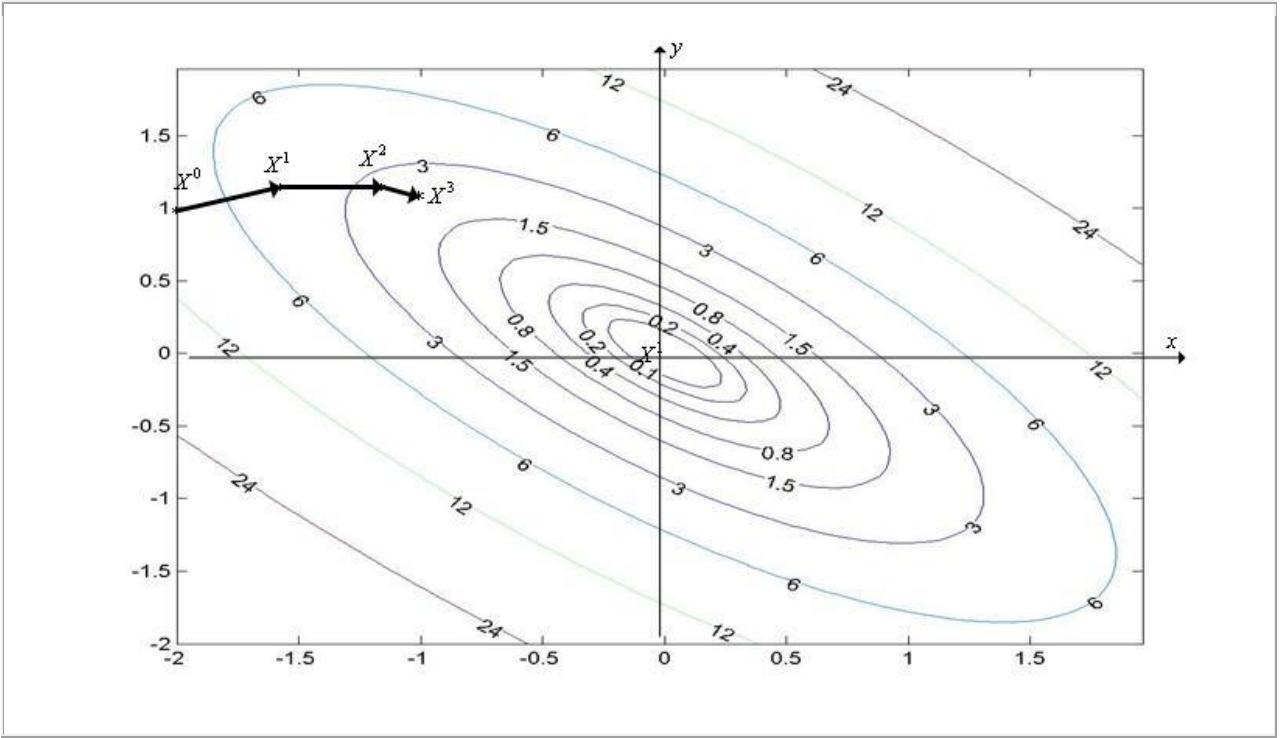


Рис. 3. К примеру 1. Фрагмент (три итерации) траектории поиска минимума функции (11) градиентным методом с дроблением шага, исходя из точки  $\mathbf{X}^0 = (x_0, y_0) = (-2.0, 1.0)$ .

2. Итерационная формула.

Итерация градиентного метода с дроблением шага для задачи (10), (11) имеет вид

$$\mathbf{X}^{r+1} = \begin{pmatrix} x^{r+1} \\ y^{r+1} \end{pmatrix} = \begin{pmatrix} x^r \\ y^r \end{pmatrix} + \lambda^r \begin{pmatrix} s_x^r \\ s_y^r \end{pmatrix} \quad (12)$$

$$\text{где } \begin{pmatrix} s_x^r \\ s_y^r \end{pmatrix} = \mathbf{S}^r = -\frac{\nabla \Phi^r}{\|\nabla \Phi^r\|}, \quad \nabla \Phi^r = \begin{pmatrix} \frac{\partial \Phi(x^r, y^r)}{\partial x} \\ \frac{\partial \Phi(x^r, y^r)}{\partial y} \end{pmatrix}, \quad (13)$$

а величина шага  $\lambda^r$  находится из условия

$$\Phi(\mathbf{X}^r) - \Phi(\mathbf{X}^{r+1}) \geq 0.5 \lambda^r \|\nabla \Phi^r\|. \quad (14)$$

Найдем явные выражения для частных производных функции (11):

$$\frac{\partial \Phi(x, y)}{\partial x} = 8x + 6y; \quad \frac{\partial \Phi(x, y)}{\partial y} = 8y + 6x. \quad (15)$$

Таким образом, из (12), (13), (15) имеем искомую итерационную формулу градиентного метода с дроблением шага для задачи (10), (11).

$$\mathbf{X}^{r+1} = \begin{pmatrix} x^{r+1} \\ y^{r+1} \end{pmatrix} = \begin{pmatrix} x^r \\ y^r \end{pmatrix} + \lambda^r \begin{pmatrix} s_x^r \\ s_y^r \end{pmatrix}, \quad s_x^r = -\frac{8x^r + 6y^r}{\|\nabla \Phi^r\|}, \quad s_y^r = -\frac{8y^r + 6x^r}{\|\nabla \Phi^r\|},$$

$$\|\nabla \Phi(\mathbf{X})\| = \sqrt{(8x + 6y)^2 + (8y + 6x)^2}. \quad (16)$$

3. Первая итерация ( $r=0$ ).

Из формул (15), (16) последовательно имеем

$$\mathbf{X}^1 = \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} = \begin{pmatrix} x^0 \\ y^0 \end{pmatrix} + \lambda^0 \begin{pmatrix} s_x^0 \\ s_y^0 \end{pmatrix}$$

$$\frac{\partial \Phi(x^0, y^0)}{\partial x} = 8x^0 + 6y^0 = 8(-2) + 6 = -10, \quad \frac{\partial \Phi(x^0, y^0)}{\partial y} = 8y^0 + 6x^0 = 8 + 6(-2) = -4,$$

$$\|\nabla \Phi^0\| = \sqrt{(-10)^2 + (-4)^2} \approx 10.77,$$

$$s_x^0 \approx \frac{10}{10.77} \approx 0.93, \quad s_y^0 \approx \frac{-4}{10.77} \approx 0.37,$$

$$\mathbf{X}^1 = \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} = \begin{pmatrix} x^0 \\ y^0 \end{pmatrix} + \lambda^0 \begin{pmatrix} s_x^0 \\ s_y^0 \end{pmatrix} = \begin{pmatrix} -2 \\ 1 \end{pmatrix} + 0.5 \begin{pmatrix} 0.93 \\ 0.37 \end{pmatrix} = \begin{pmatrix} -1.54 \\ 1.18 \end{pmatrix}$$

Таким образом,  $\mathbf{X}^1 = \begin{pmatrix} -1.54 \\ 1.18 \end{pmatrix}$  (см. рис. 3).

Условие (14) на первой итерации имеет вид

$$\Phi(\mathbf{X}^0) - \Phi(\mathbf{X}^1) \geq 0.5 \lambda^0 \|\nabla \Phi^0\|.$$

Поскольку

$$\Phi(\mathbf{X}^0) = \Phi(x^0, y^0) = 4(x^0)^2 + 4(y^0)^2 + 6x^0y^0 = 4(-2)^2 + 4 + 6(-2) = 8,$$

$$\Phi(\mathbf{X}^1) = \Phi(x^1, y^1) = 4(x^1)^2 + 4(y^1)^2 + 6x^1y^1 \approx 4(-1.54)^2 + 4(1.18)^2 + 6(-1.54)(1.18) \approx 4.16,$$

левая часть этого неравенства равна  $8 \cdot 0.4 \cdot 1.6 = 3.84$ . Его правая часть, легко видеть, равна  $0.5^2 \cdot 10.77 = 2.69$ .

Таким образом, на первой итерации условие (14) выполняется и величина шага  $\lambda$  должна быть изменена:  $\lambda^1 = 0.7 \cdot 0.5 = 0.35$ .

4. Вторая итерация ( $r=1$ ).

Аналогично первой итерации последовательно имеем

$$\mathbf{X}^2 = \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} = \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} + \lambda^1 \begin{pmatrix} s_x^1 \\ s_y^1 \end{pmatrix}$$

$$\frac{\partial \Phi(x^1, y^1)}{\partial x} = 8x^1 + 6y^1 = 8(-1.54) + 6 \cdot 1.18 = -5.24, \quad \frac{\partial \Phi(x^1, y^1)}{\partial y} = 8y^1 + 6x^1 = 8 \cdot 1.18 + 6(-1.54) = 0.20,$$

$$\|\nabla \Phi^1\| = \sqrt{(-5.24)^2 + (0.20)^2} \approx 5.24,$$

$$s_x^1 \approx \frac{5.24}{5.24} \approx 1.0, \quad s_y^1 \approx -\frac{0.2}{5.24} \approx -0.04,$$

$$\mathbf{X}^2 = \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} = \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} + \lambda^1 \begin{pmatrix} s_x^1 \\ s_y^1 \end{pmatrix} = \begin{pmatrix} -1.54 \\ 1.18 \end{pmatrix} + 0.35 \begin{pmatrix} 1.0 \\ -0.04 \end{pmatrix} = \begin{pmatrix} -1.19 \\ 1.22 \end{pmatrix}$$

$$\mathbf{X}^2 = \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} = \begin{pmatrix} -1.19 \\ 1.22 \end{pmatrix}$$

Таким образом, (см. рис. 3).

Условие (14) на второй итерации имеет вид

$$\Phi(\mathbf{X}^1) - \Phi(\mathbf{X}^2) \geq 0.5 \lambda^1 \|\nabla \Phi^1\|.$$

Поскольку

$$\Phi(\mathbf{X}^1) \approx 4.16$$

$$\Phi(\mathbf{X}^2) = \Phi(x^2, y^2) = 4(x^2)^2 + 4(y^2)^2 + 6x^2y^2 \approx 4(-1.19)^2 + 4(1.22)^2 + 6(-1.19)(1.22) \approx 2.91,$$

левая часть этого неравенства равна  $4.16 - 2.91 = 1.25$ . Его правая часть, легко видеть, равна  $0.5 \cdot 0.35 \cdot 5.24 = 0.92$ .

Таким образом, на второй итерации условие (14) выполняется и величина шага  $\lambda$  должна быть изменена:  $\lambda^2 = 0.7 \cdot 0.35 \approx 0.24$ .

5. Третья итерация ( $r=2$ ).

Аналогично первой итерации последовательно имеем

$$\mathbf{X}^3 = \begin{pmatrix} x^3 \\ y^3 \end{pmatrix} = \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} + \lambda^2 \begin{pmatrix} s_x^2 \\ s_y^2 \end{pmatrix}$$

$$\frac{\partial \Phi(x^2, y^2)}{\partial x} = 8x^2 + 6y^2 = 8(-1.19) + 6 \cdot 1.22 = -2.20, \quad \frac{\partial \Phi(x^2, y^2)}{\partial y} = 8y^2 + 6x^2 = 8 \cdot 1.22 + 6(-1.19) = 2.62,$$

$$\|\nabla \Phi^2\| = \sqrt{(-2.20)^2 + (2.62)^2} \approx 3.42,$$

$$s_x^2 \approx \frac{-2.20}{3.42} \approx -0.64, \quad s_y^2 \approx \frac{2.62}{3.42} \approx 0.77,$$

$$\mathbf{X}^3 = \begin{pmatrix} x^3 \\ y^3 \end{pmatrix} = \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} + \lambda^2 \begin{pmatrix} s_x^2 \\ s_y^2 \end{pmatrix} = \begin{pmatrix} -1.19 \\ 1.22 \end{pmatrix} + 0.24 \begin{pmatrix} 0.64 \\ -0.77 \end{pmatrix} \approx \begin{pmatrix} -1.04 \\ 1.03 \end{pmatrix}$$

$$\mathbf{X}^3 = \begin{pmatrix} x^3 \\ y^3 \end{pmatrix} = \begin{pmatrix} -1.04 \\ 1.03 \end{pmatrix}$$

Таким образом, (см. рис. 3).

Условие (14) на третьей итерации имеет вид

$$\Phi(\mathbf{X}^2) - \Phi(\mathbf{X}^3) \geq 0.5 \lambda^2 \|\nabla \Phi^2\|.$$

Поскольку

$$\Phi(\mathbf{X}^2) \approx 2.91,$$

$$\Phi(\mathbf{X}^3) = \Phi(x^3, y^3) = 4(x^3)^2 + 4(y^3)^2 + 6x^3y^3 \approx 4(-1.04)^2 + 4(1.03)^2 + 6(-1.04)(1.03) \approx 2.14,$$

левая часть этого неравенства равна  $2.91 - 2.14 = 0.77$ . Его правая часть, легко видеть, равна  $0.5 \cdot 0.24 \cdot 3.42 = 0.41$ .

Таким образом, на третьей итерации условие (14) выполняется и величина шага  $\lambda$  должна быть изменена:  $\lambda^3 = 0.7 \cdot 0.24 \approx 0.17$ .

## 7.2. Метод оптимизации Ньютона

Положим, что функция  $\Phi(\mathbf{X})$  всюду дважды дифференцируема в  $n$ -мерном евклидовом пространстве  $\mathbb{R}^n$ .

Рассмотрим следующую многомерную задачу локальной безусловной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного в  $n$ -мерном евклидовом пространстве  $\mathbb{R}^n$ ,

$$\min_{\mathbf{X} \in \mathbb{R}^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*. \quad (1)$$

Обоснование метода оптимизации Ньютона.

Рассмотрим первые три члена разложения функции  $\Phi(\mathbf{X})$  в ряд Тейлора в окрестности точки  $\mathbf{X}^r$ :

$$\Phi(\mathbf{X}) \approx \tilde{\Phi}^r(\mathbf{X}) = \Phi(\mathbf{X}^r) + (\nabla \Phi(\mathbf{X}^r), (\mathbf{X} - \mathbf{X}^r)) + \frac{1}{2} (\mathbf{H}(\mathbf{X}^r)(\mathbf{X} - \mathbf{X}^r), (\mathbf{X} - \mathbf{X}^r)) \quad (2)$$

Здесь  $\mathbf{H}(\mathbf{X})$  - матрица Гессе функции  $\Phi(\mathbf{X})$ . Из (2) следует, что градиент функции  $\tilde{\Phi}^r(\mathbf{X})$  равен

$$\nabla \tilde{\Phi}^r(\mathbf{X}) = \nabla \Phi(\mathbf{X}^r) + \mathbf{H}(\mathbf{X}^r)(\mathbf{X} - \mathbf{X}^r). \quad (3)$$

Если матрица Гессе  $\mathbf{H}(\mathbf{X}^r)$  положительно определена, то функция  $\tilde{\Phi}^r(\mathbf{X})$  достигает минимума в точке, в которой градиент этой функции равен нулевому вектору.

Таким образом, в точке  $\mathbf{X}^{r+1}$  минимума функции  $\tilde{\Phi}^r(\mathbf{X})$  справедливо равенство

$$\nabla \Phi(\mathbf{X}^r) + \mathbf{H}(\mathbf{X}^r)(\mathbf{X}^{r+1} - \mathbf{X}^r) = 0, \quad (4)$$

где  $0$  -  $n$ -мерный вектор нулей. Отсюда получаем итерационную формулу

$$\mathbf{X}^{r+1} = \mathbf{X}^r - \mathbf{H}^{-1}(\mathbf{X}^r) \nabla \Phi(\mathbf{X}^r) = \mathbf{X}^r + \Delta^r \quad (5)$$

для отыскания очередного приближения к точке минимума функции  $\Phi(\mathbf{X})$ . Здесь

$$\Delta^r = -\mathbf{H}^{-1}(\mathbf{X}^r) \nabla \Phi(\mathbf{X}^r). \quad (6)$$

Выражение (5) представляет собой итерационную формулу решения системы уравнений (4) широко известным методом касательных (методом Ньютона) – см. параграф 4.8. Этим фактом объясняется название рассматриваемого метода оптимизации.

Найдем скалярное произведение градиента функции  $\Phi(\mathbf{X})$  в точке  $\mathbf{X}^r$  и вектора  $\Delta^r$ :

$$(\nabla \Phi(\mathbf{X}^r), \Delta^r) = (\nabla \Phi(\mathbf{X}^r), \mathbf{H}^{-1}(\mathbf{X}^r) \nabla \Phi(\mathbf{X}^r)) < 0. \quad (7)$$

Последнее неравенство справедливо в силу постулируемой положительной определенности матрицы Гессе в точке  $\mathbf{X}^r$ . Геометрически неравенство (7) означает, что вектор  $\Delta^r$  образует тупой угол с градиентом целевой функции  $\Phi(\mathbf{X})$  в точке  $\mathbf{X}^r$  (см. рис. 1). Таким образом, при минимизации овражных функций вектор  $\Delta^r$  может составлять с осью оврага меньший угол, чем вектор антиградиента. Эта особенность делает метод оптимизации Ньютона, вообще говоря, более эффективным, чем градиентный метод наискорейшего спуска.

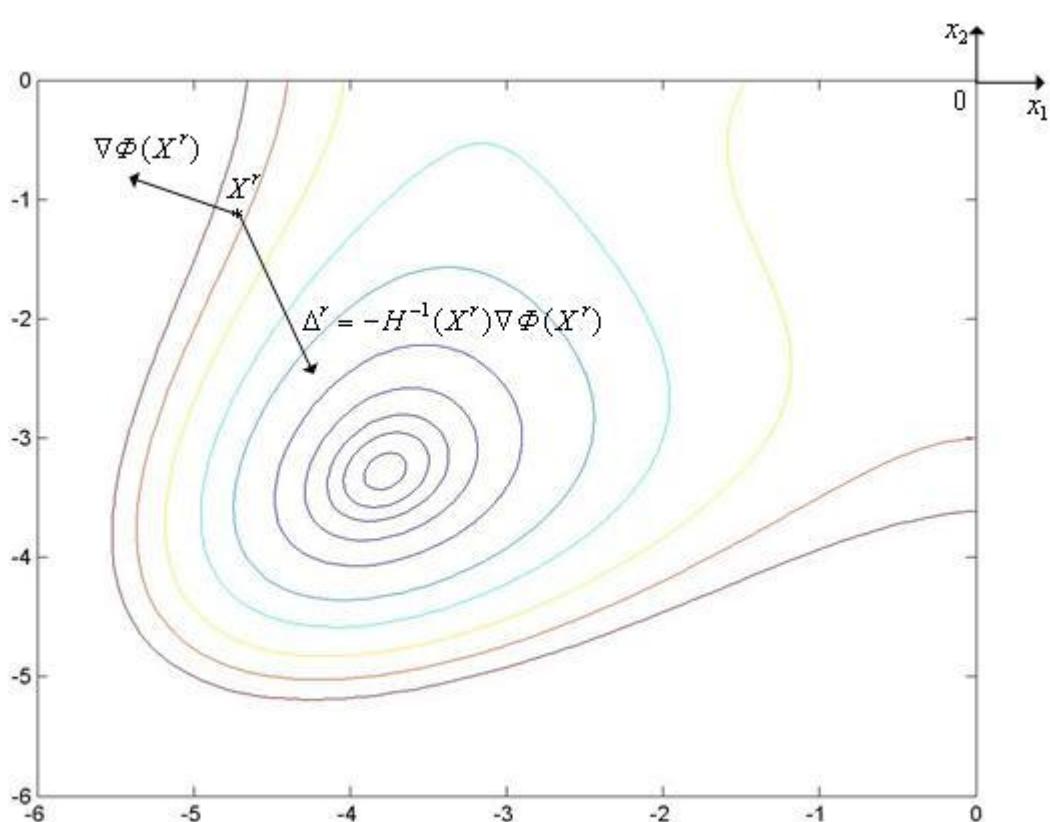


Рис. 1. К обоснованию метода многомерной оптимизации Ньютона.

Отметим трудности, которые могут возникать при использовании итерационной формулы (5):

Если размерность пространства  $R^n$  велика, то обращение на каждой итерации матрицы Гессе  $\mathbf{H}(\mathbf{X}^r)$  может потребовать значительных вычислительных ресурсов;

Значение минимизируемой функции  $\Phi(\mathbf{X})$  в точке  $\mathbf{X}^{r+1} = \mathbf{X}^r + \Delta^r$  может превышать значение функции в предыдущей точке  $\mathbf{X}^r$  вследствие того, что направление  $\Delta^r$  ведет к уменьшению  $\Phi(\mathbf{X})$ , но величина шага слишком велика;

Направление спуска, определяемое вектором  $\Delta^r = -\mathbf{H}^{-1}(\mathbf{X}^r) \nabla \Phi(\mathbf{X}^r)$ , ведет к убыванию целевой функции только при положительной определенности матрицы Гессе  $\mathbf{H}(\mathbf{X}^r)$ . Это приводит к тому, что на каждой итерации необходимы вычислительные затраты на

проверку обусловленности этой матрицы. Указанная матрица может быть плохо обусловленной. Более того, указанная матрица может быть вырожденной и, поэтому, не иметь обратной матрицы.

Вследствие этих трудностей итерационная формула (5) в «чистом» виде не используется в вычислительной практике.

Для того чтобы избежать обращения матрицы Гессе, на практике вектор  $\Delta^r$  находят обычно из следующей системы линейных алгебраических уравнений (СЛАУ), вытекающей из равенства (6):

$$\mathbf{H}(\mathbf{X}^r)\Delta^r = -\nabla\Phi(\mathbf{X}^r). \quad (8)$$

СЛАУ (8) может быть решена различными численными методами (например, прямыми методами, итерационными методами).

Величина шага в направлении  $\Delta^r$ , которая приводит к убыванию функции  $\Phi(\mathbf{X})$ , может быть обеспечена путем добавления в итерационную формулу (5) коэффициента  $\lambda^r \in (0, 1]$ , т.е. путем использования вместо формулы (5) итерационной формулы

$$\mathbf{X}^{r+1} = \mathbf{X}^r + \lambda^r \mathbf{H}^{-1}(\mathbf{X}^r) \nabla\Phi(\mathbf{X}^r) = \mathbf{X}^r + \lambda^r \Delta^r, \quad (9)$$

где коэффициент  $\lambda^r$  выбирают тем или иным способом так, чтобы обеспечить условие  $\Phi(\mathbf{X}^{r+1}) < \Phi(\mathbf{X}^r)$ .

Для того, чтобы направление спуска независимо от определенности матрицы Гессе  $\mathbf{H}(\mathbf{X}^r)$  вело у убыванию функции  $\Phi(\mathbf{X})$ , в качестве вектора  $\Delta^r$  можно использовать вектор

$$\Delta^r = (\mu^r \mathbf{I} + \mathbf{H}(\mathbf{X}^r))^{-1} \nabla\Phi(\mathbf{X}^r), \quad (10)$$

где  $\mathbf{I}$  -  $(n \times n)$ -единичная матрица, а  $\mu^r > 0$  - параметр, выбираемый так, чтобы матрица  $\mu^r \mathbf{I} + \mathbf{H}(\mathbf{X}^r)$  являлась положительно определенной.

Схема метода оптимизации Ньютона.

Рассмотрим схему одной из модификаций метода оптимизации Ньютона, в которой используется итерационная формула (9) и вектор  $\Delta^r$  находят путем решения на каждой итерации СЛАУ (8).

Задаем начальную точку  $\mathbf{X}^0$ , начальную величину шага  $\lambda^0 = 1$  и коэффициент дробления шага  $\nu \in (0, 1]$ . Полагаем счетчик числа итераций  $r = 0$ .

Вычисляем в точке  $\mathbf{X}^r$  вектор градиента  $\nabla\Phi(\mathbf{X}^r)$  и матрицу Гессе  $\mathbf{H}(\mathbf{X}^r)$ .

Решаем СЛАУ (8) и находим вектор  $\Delta^r$ .

По формуле (9) вычисляем компоненты вектора  $\mathbf{X}^{r+1}$ .

Вычисляем величину  $\Phi(\mathbf{X}^{r+1})$  - значение функции  $\Phi(\mathbf{X})$  в точке  $\mathbf{X}^{r+1}$ .

Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем  $\mathbf{X}^* \approx \mathbf{X}^{r+1}$ , и завершаем итерации. Иначе – переходим к следующему пункту.

Если  $\Phi(\mathbf{X}^{r+1}) < \Phi(\mathbf{X}^r)$ , то полагаем  $r = r + 1$  и переходим к п.2. Иначе – фиксированное число раз полагаем  $\lambda^r = v\lambda^r$  и переходим к пункту 4 •

В качестве условия окончания поиска можно использовать одно из стандартных условий окончания итераций:

$$\|\mathbf{X}^{r+1} - \mathbf{X}^r\| = \lambda^r \leq \varepsilon_X, \quad |\Phi(\mathbf{X}^{r+1}) - \Phi(\mathbf{X}^r)| \leq \varepsilon_\Phi$$

или условие  $\|\nabla \Phi^r\| \leq \varepsilon_V$ , где  $\varepsilon_V$  – константа, определяющая требуемую точность решения по градиенту функции  $\Phi(\mathbf{X})$ .

### 8.1. Метод с возвратом при неудачном шаге. Метод наилучшей пробы

Рассматривается следующая многомерная задача локальной безусловной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного в  $n$ -мерном евклидовом пространстве  $R^n$ ,

$$\min_{\mathbf{X} \in R^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*. \quad (1)$$

При решении задачи (1) методом с возвратом при неудачном шаге (одношаговый метод оптимизации) используется итерационная формула

$$\mathbf{X}^{r+1} = \mathbf{X}^r + \lambda^r \frac{\Psi^r}{\|\Psi^r\|}, \quad (2)$$

где  $\lambda^r$  - величина шага на  $r$ -ой итерации,  $\Psi^r = (\psi_1^r, \psi_2^r, \dots, \psi_n^r)$  - реализация  $n$ -мерного случайного вектора,  $\|\cdot\|$  - некоторая векторная норма. Обычно в качестве координат вектора  $\Psi^r$  используют независимые случайные величины, равномерно распределенные в интервале  $[-1, 1]$ .

Схема метода с возвратом при неудачном шаге.

Задаем начальную точку  $\mathbf{X}^0$ , начальную длину шага  $\lambda^0$  и полагаем счетчик числа итераций  $r=0$ .

Задаем начальное значение счетчика числа неудачных попыток  $k=1$ .

Получаем реализацию случайных чисел  $\psi_1^r, \psi_2^r, \dots, \psi_n^r$  - компонент вектора  $\Psi^r$  и по формуле (2) находим пробную точку  $\mathbf{X}^{r+1}$ .

Вычисляем значение  $\Phi(\mathbf{X}^{r+1})$  функции  $\Phi(\mathbf{X})$  в точке  $\mathbf{X}^{r+1}$ .

Если  $\Phi(\mathbf{X}^{r+1}) < \Phi(\mathbf{X}^r)$ , то полагаем  $r=r+1$  и переходим к п.3. Иначе – переходим к п.6. Полагаем  $k=k+1$ . Если  $k < K$ , то переходим к п.3. Иначе – переходим к п.7. Здесь  $K$  – предельное количество неудачных попыток (свободный параметр метода). Рекомендуется  $K=3n$ .

Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска

$\mathbf{X}^* \approx \mathbf{X}^{r+1}$  выполнено, то полагаем  $\mathbf{X}^* = \mathbf{X}^{r+1}$  и завершаем итерации. Иначе – полагаем  $r=r+1$ ,  $\lambda^r = \alpha \lambda^r$  и переходим к п. 2. Здесь  $\alpha \in (0, 1)$  - коэффициент уменьшения шага (свободный параметр метода) •

В качестве условия окончания поиска можно использовать одно из стандартных условий окончания итераций:

$$\|\mathbf{X}^{r+1} - \mathbf{X}^r\| \leq \lambda^r \leq \varepsilon_X, \quad (3)$$

где  $\varepsilon_X$  - константа, определяющая требуемую точность решения по  $\mathbf{X}$ ;

$$|\Phi(\mathbf{X}^{r+1}) - \Phi(\mathbf{X}^r)| \leq \varepsilon_\Phi, \quad (4)$$

где  $\varepsilon_\Phi$  - константа, определяющая требуемую точность решения по  $\Phi$ .

Метод с возвратом при неудачном шаге иллюстрирует рис. 1, на котором показан фрагмент линий уровня функции Химмельблау.

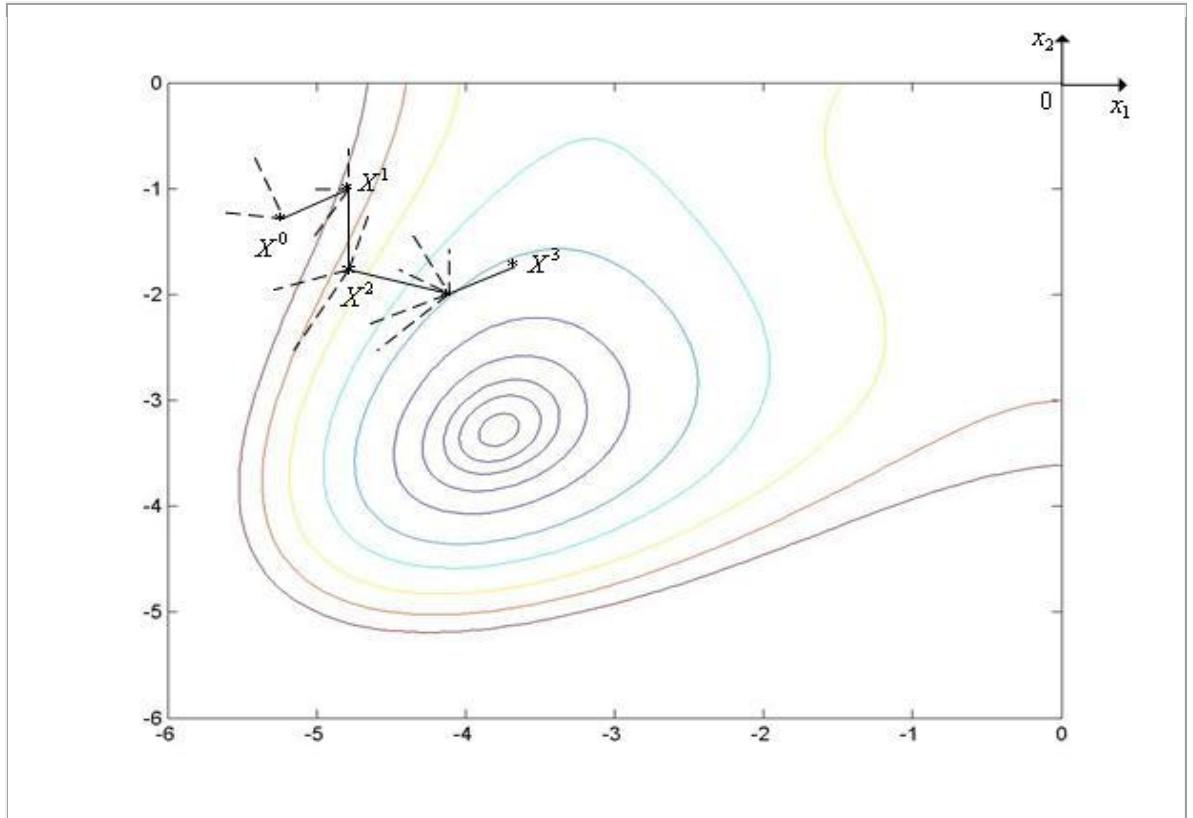


Рис. 1. Траектория поиска минимума функции Химмельблау методом с возвратом при неудачном шаге. Пунктиром на рисунке показаны неудачные шаги.

Модификацией метода с возвратом при неудачном шаге является метод наилучшей пробы (также одношаговый метод оптимизации).

Схема метода наилучшей пробы.

Задаем начальную точку  $\mathbf{X}^0$ , начальную длину шага  $\lambda^0$  и полагаем счетчик числа итераций  $r=0$ .

Генерируем  $M$  случайных векторов  $\Psi_i^r, i \in [1, M]$  и по формуле (2) находим пробные точки  $\mathbf{X}_i^{r+1}, i \in [1, M]$

Вычисляем значения  $\Phi(\mathbf{X}_i^{r+1})$  функции  $\Phi(\mathbf{X})$  в пробных точках  $\mathbf{X}_i^{r+1}, i \in [1, M]$  и находим минимальное из этих значений

$$\Phi(\mathbf{X}^{r+1}) = \Phi(\mathbf{X}_k^{r+1}) = \min_{i \in [1, M]} \Phi(\mathbf{X}_i^{r+1}).$$

Если  $\Phi(\mathbf{X}^{r+1}) < \Phi(\mathbf{X}^r)$ , то полагаем  $r=r+1$  и переходим к п.2. Иначе – переходим к п.5.

Проверяем условие окончания поиска (см. (3), (4)). Если условие окончания поиска

выполнено, то полагаем  $\mathbf{X}^* \approx \mathbf{X}^{r+1}$  и завершаем итерации. Иначе – полагаем  $r=r+1$ ,  $\lambda^r = \alpha \lambda^r$  и переходим к п.2. Здесь  $\alpha \in (0, 1)$  – коэффициент уменьшения шага (свободный параметр метода) •

Метод наилучшей пробы иллюстрирует рис. 2, на котором показан фрагмент линий уровня функции Химмельблау.

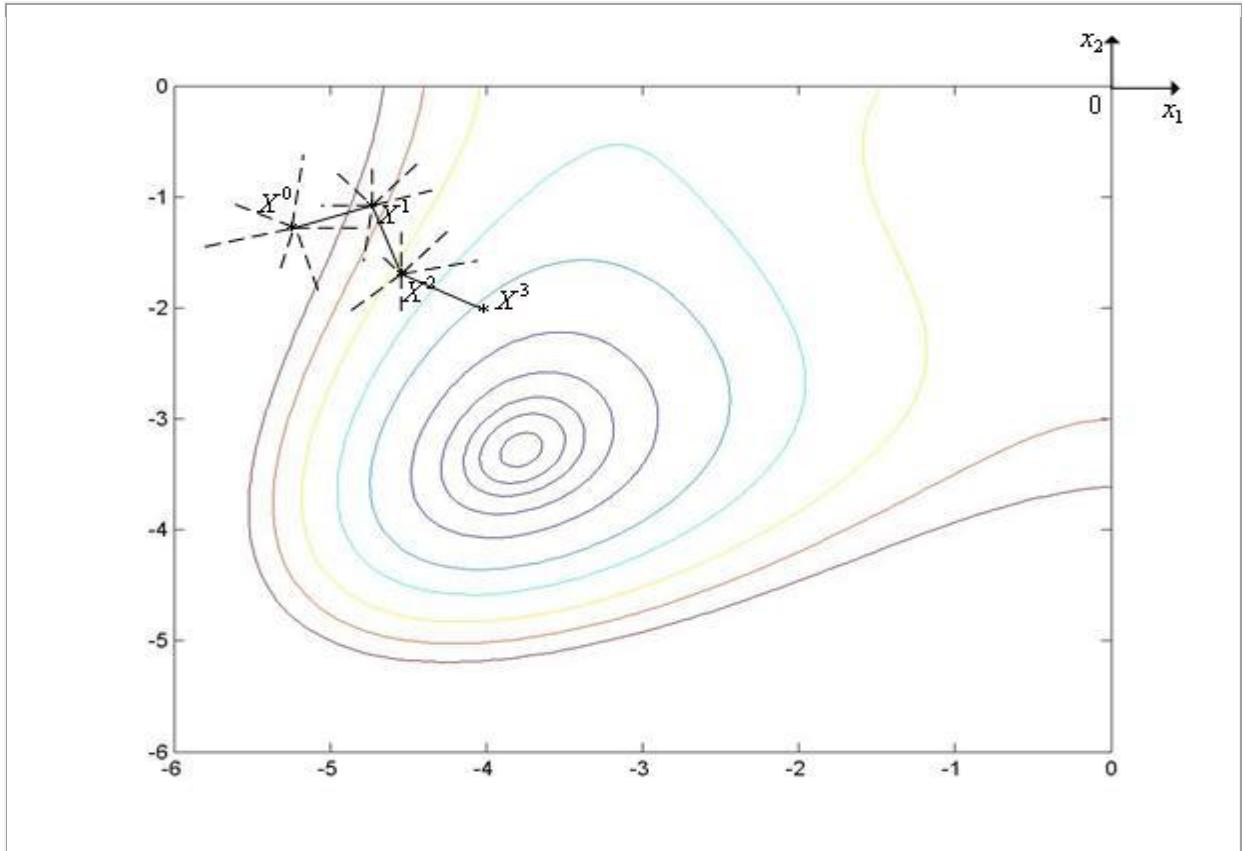


Рис. 2. Траектория поиска минимума функции Химмельблау методом наилучшей пробы. Пунктиром на рисунке показаны неудачные пробы.

## 8.2. Метод комплексов

Рассматривается следующая многомерная задача локальной безусловной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного в  $n$ -мерном евклидовом пространстве  $R^n$ ,

$$\min_{\mathbf{X} \in R^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*. \quad (1)$$

Комплексом называется многогранник с  $N > n + 1$  вершинами (не обязательно выпуклый). Рекомендуется  $N = 2n$ . Вообще говоря, комплексом в комбинаторной топологии называется геометрическая фигура, которая может быть разбита на более элементарные фигуры. В нашем случае такими элементарными фигурами являются симплексы. Поэтому, говоря более строго, в данном параграфе рассматриваются симплексиальные комплексы.

При решении задачи (1) методом комплексов используются следующие операции: генерация случайного комплекса;

отражение вершины комплекса с растяжением;

сжатие комплекса.

Генерация случайного комплекса. В пространстве  $R^n$  координаты вершин случайного комплекса с  $N$  вершинами могут быть найдены по формуле

$$\mathbf{X}_i = \mathbf{X}_0 + l \frac{\Psi_i}{\|\Psi_i\|}, \quad i \in [0, N], \quad (2)$$

где  $\mathbf{X}_0$  - произвольная начальная точка,  $i$  - номер вершины комплекса,  $l$  - скаляр, определяющий размеры комплекса,  $\Psi_i$  - реализация  $n$ -мерного случайного вектора,  $\|\cdot\|$  - некоторая векторная норма. Обычно в качестве координат вектора  $\Psi_i$  используют независимые случайные величины, равномерно распределенные в интервале  $[-1, 1]$ .

Отражение вершины комплекса с растяжением. Положим, что в пространстве  $R^n$  тем или иным способом задан комплекс  $C^r$  с  $N$  вершинами  $\mathbf{X}_i^r, i \in [1, N]$ , и его вершину  $\mathbf{X}_k^r$  необходимо отразить через центр тяжести комплекса с растяжением. В новом комплексе  $C^{r+1}$  все вершины, кроме  $k$ -ой, совпадают с соответствующими вершинами исходного комплекса  $C^r$ , а  $k$ -я вершина находится на прямой, проходящей через центр тяжести этого комплекса и его вершину  $\mathbf{X}_k^r$  (см. рис. 1). Обозначим координаты вершин нового комплекса  $\mathbf{X}_i^{r+1}, i \in [1, N]$ . Тогда имеем

$$\mathbf{X}_i^{r+1} = \mathbf{X}_i^r, \quad i \in [1, N], \quad i \neq k; \quad \mathbf{X}_k^{r+1} = \mathbf{X}_c^r + \alpha (\mathbf{X}_c^r - \mathbf{X}_k^r), \quad (3)$$

где  $\alpha$  - коэффициент растяжения комплекса (рекомендуемое значение - 1.3),  $\mathbf{X}_c^r$  - вектор координат центра тяжести комплекса  $C^r$ :

$$\mathbf{X}_c^r = \frac{1}{N} \sum_{i=1}^N \mathbf{X}_i^r. \quad (4)$$

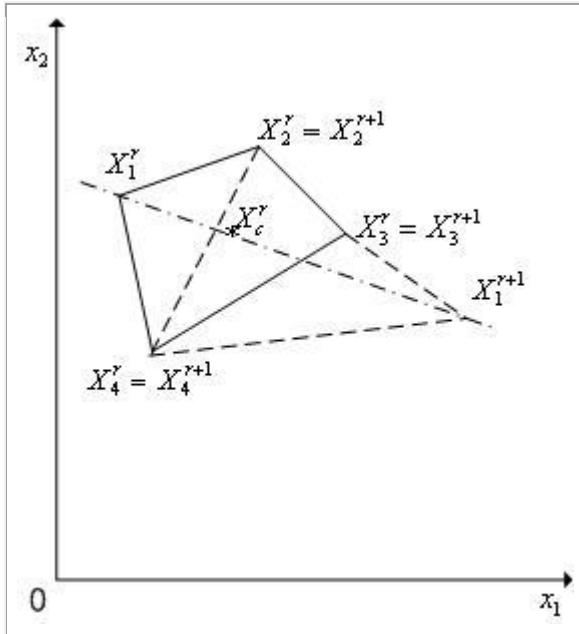


Рис. 1. Отражение вершины  $X_1^r$  комплекса  $C^r$  через центр его тяжести с растяжением. Пунктиром показан новый комплекс  $C^{r+1}$ .

Сжатие комплекса. Положим, что в пространстве  $\mathbb{R}^n$  тем или иным способом задан комплекс  $C^r$  с  $N$  вершинами  $\mathbf{X}_i^r$ ,  $i \in [1, N]$ , и его вершину  $\mathbf{X}_k^r$  необходимо переместить ближе к центру тяжести комплекса  $C^r$  - выполнить сжатие комплекса. В новом комплексе  $C^{r+1}$  все вершины, кроме  $k$ -ой, совпадают с соответствующими вершинами исходного комплекса  $C^r$ , а  $k$ -я вершина находится на прямой, проходящей через центр тяжести этого комплекса и его вершину  $\mathbf{X}_k^r$  (см. рис. 2). Обозначим координаты вершин нового комплекса  $\mathbf{X}_i^{r+1}$ ,  $i \in [1, N]$ . Тогда имеем

$$\mathbf{X}_i^{r+1} = \mathbf{X}_i^r, i \in [1, N], i \neq k; \mathbf{X}_k^{r+1} = \mathbf{X}_c^r + \beta (\mathbf{X}_k^r - \mathbf{X}_c^r), \quad (5)$$

где  $\beta$  - коэффициент сжатия комплекса (рекомендуемое значение - 2),  $\mathbf{X}_c^r$  - вектор координат центра тяжести комплекса  $C^r$  (см. (4)).

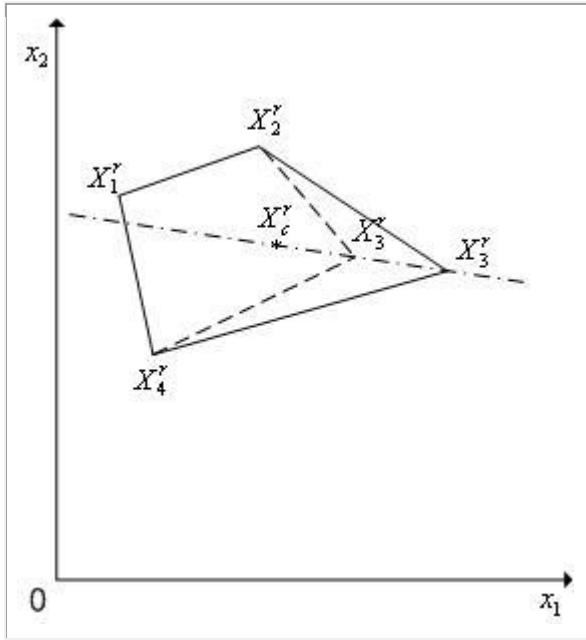


Рис. 2. Сжатие комплекса  $\text{Cr}$  Пунктиром показан новый комплекс  $\text{Cr}$ .  
Схема простейшего варианта метода комплексов.

Задаем начальную точку  $\mathbf{X}_0^0$ , исходя из которой должен быть построен комплекс  $C^0$ , начальное значение величины  $l=l^0$  и полагаем счетчик числа итераций  $r=0$ .

Генерируем  $N$  случайных векторов  $\Psi_i^r, i \in [1, N]$  и по формуле (2) находим координаты  $\mathbf{X}_i^r, i \in [1, N]$  вершин комплекса  $C^r$ .

Вычисляем значения  $\Phi(\mathbf{X}_i^r)$  функции  $\Phi(\mathbf{X})$  во всех вершинах комплекса  $C^r$ .

Находим максимальное из значений функции  $\Phi(\mathbf{X})$  в вершинах комплекса  $C^r$   
 $\Phi(\mathbf{X}_k^r) = \max_{i \in [1, N]} \Phi(\mathbf{X}_i^r)$ .

По формулам (3), (4) для комплекса  $C^r$  выполняем отражение вершины комплекса с растяжением  $\mathbf{X}_k^r$  - получаем вершину  $\mathbf{X}_k^{r+1}$  и новый комплекс  $C^{r+1}$ .

Вычисляем значение  $\Phi(\mathbf{X}_k^{r+1})$  функции  $\Phi(\mathbf{X})$  в вершине  $\mathbf{X}_k^{r+1}$ .

Если  $\Phi(\mathbf{X}_k^{r+1}) < \Phi(\mathbf{X}_k^r)$ , то полагаем  $r=r+1$  и переходим к п.8. Иначе – по формулам (5), (4) выполняем сжатие симплекса  $C^{r+1}$  в направлении  $\mathbf{X}_k^{r+1} - \mathbf{X}_c^{r+1}$ , полагаем  $r=r+2$  и переходим к п.6.

Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то в качестве точки  $\mathbf{X}^*$  полагаем вершину комплекса  $C^r$ , к которой функция  $\Phi(\mathbf{X})$  имеет наименьшее значение и завершаем итерации. Иначе – переходим к п. 4 •

В качестве критерия окончания поиска может использоваться следующее условие: максимальная длина ребра комплекса  $C^r$  не превышает  $\epsilon_X$  - требуемую точность решения по  $\mathbf{X}$ . Может использоваться также следующее аналогичное условие: максимальная разность значений функции  $\Phi(\mathbf{X})$  в двух вершинах комплекса  $C^r$  не превышает  $\epsilon_\Phi$  - требуемую точность решения по  $\Phi$ .

Могут использоваться также более сложные условия окончания поиска

$$\frac{1}{N} \sqrt{\sum_{i=1}^N \| \mathbf{X}_i^r - \mathbf{X}_c^r \|} \leq \varepsilon_X, \quad (6)$$

$$\frac{1}{N} \sqrt{\sum_{i=1}^N (\Phi(\mathbf{X}_i^r) - \Phi_c^r)^2} \leq \varepsilon_\Phi. \quad (7)$$

В формуле (6) векторная норма означает расстояние вершины  $\mathbf{X}_i^r$  до центра тяжести комплекса  $C^r$ , а сама формула (6) определяет среднее расстояние вершин комплекса  $C^r$  до его центра тяжести.

$$\Phi_c^r = \frac{1}{N} \sum_{i=1}^N \Phi(\mathbf{X}_i^r)$$

В формуле (7) есть среднее значение функции  $\Phi(\mathbf{X})$  в вершинах комплекса  $C^r$ , а сама формула (7) определяет среднее отклонение значений функции  $\Phi(\mathbf{X})$  в вершинах комплекса  $C^r$  от этого среднего значения.

Метод комплексов иллюстрирует рис. 3, на котором показан фрагмент линий уровня функции Химмельблау. На рисунке исходный комплекс  $C^r$  имеет вершины  $\mathbf{X}_i^r$ ,  $i \in [1, 4]$ . После отражения с растяжением вершины  $\mathbf{X}_1^r$  этого комплекса, в которой функция имеет максимальное значение, получаем комплекс  $C^{r+1}$  с вершинами  $\mathbf{X}_i^{r+1}$ ,  $i \in [1, 4]$ . После отражения с растяжением вершины  $\mathbf{X}_2^r$  комплекса  $C^{r+1}$ , в которой функция имеет максимальное значение, получаем комплекс  $C^{r+2}$  с вершинами  $\mathbf{X}_i^{r+2}$ ,  $i \in [1, 4]$ .

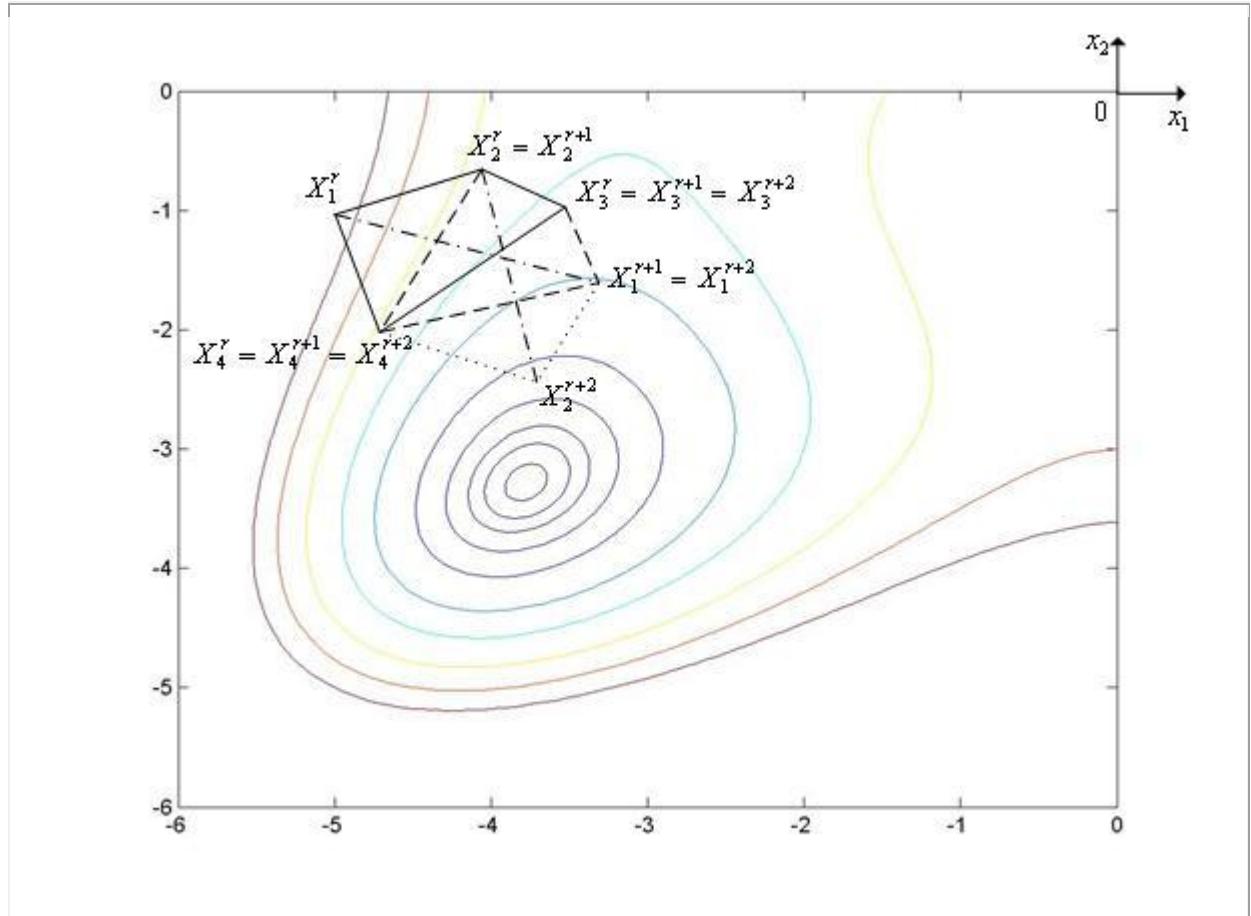


Рис. 3. Траектория поиска минимума функции Химмельблау методом комплексов.  
 Известно множество модификаций рассмотренного метода комплексов, направленных, в частности, на преодоление «уплощения» комплекса в процессе поиска. С этой целью через фиксированное количество итераций находятся максимальная и минимальная диагонали комплекса и, если их отношение превышает заданное, то по рассмотренной схеме производится построение нового комплекса.

### 8.3. Метод повторяющегося случайного поиска

Рассматривается следующая многомерная задача локальной безусловной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного в  $n$ -мерном евклидовом пространстве  $R^n$ ,

$$\min_{\mathbf{X} \in R^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*. \quad (1)$$

В методе повторяющегося случайного поиска (трех-шаговый метод) используется итерационная схема (см. рис. 1)

$$\mathbf{X}^{r+1} = \mathbf{X}^r + \lambda^r \Delta^r, \quad (2)$$

где  $\lambda^r$  - величина шага (скаляр) на  $r$ -ой итерации,  $\Delta^r$  - ( $n * 1$ )-вектор, определяющий направление шага на  $r$ -ой итерации:

$$\Delta^r = \left[ \beta \frac{\mathbf{S}^r}{\|\mathbf{S}^r\|} + (1-\beta) \frac{\mathbf{P}^r}{\|\mathbf{P}^r\|} \right]. \quad (3)$$

Здесь  $\mathbf{S}^r = \gamma \mathbf{S}^{r-1} + (1-\gamma) \mathbf{S}^{r-2}$  - вектор «предыстории», определяющий среднее направление поиска на двух предыдущих шагах;  $\|\cdot\|$  - некоторая векторная норма;  $\mathbf{P}^r$  -  $n$ -мерный вектор псевдослучайных чисел, равномерно распределенных в интервале  $[0, 1]$ ; скаляр  $\beta \in [0, 1]$  - коэффициент, задающий относительные веса детерминированной и случайной компонент в векторе  $\Delta^r$  (свободный параметр метода); скаляр  $\gamma \in [0, 1]$  - коэффициент, задающий относительные веса векторов  $\mathbf{S}^{r-1}, \mathbf{S}^{r-2}$  в векторе  $\mathbf{S}^r$  (свободный параметр метода).

Заметим, что отношение  $\frac{\mathbf{S}^r}{\|\mathbf{S}^r\|}$  представляет собой единичный вектор направления  $\mathbf{S}^r$ , а

отношение  $\frac{\mathbf{P}^r}{\|\mathbf{P}^r\|}$  - единичный вектор направления  $\mathbf{P}^r$ .

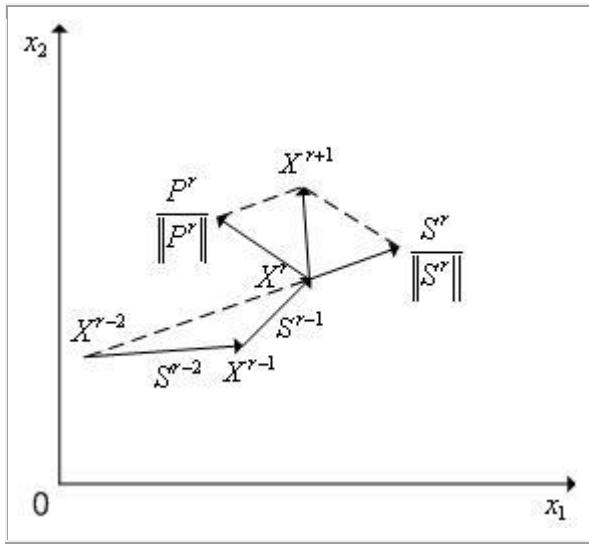


Рис. 1. К итерационной схеме метода повторяющегося случайного поиска.

Принято  $\gamma=0.5$ ,  $\beta=0.5$ ,  $\lambda^r=2$ , так что  $\mathbf{S}^r=0.5(\mathbf{S}^{r-1}+\mathbf{S}^{r-2})$  и  $\mathbf{X}^{r+1}=\mathbf{X}^r+\left[\frac{\mathbf{S}^r}{\|\mathbf{S}^r\|}+\frac{\mathbf{P}^r}{\|\mathbf{P}^r\|}\right]$ . Упрощенная схема метода повторяющегося случайного поиска.

Задаем начальную точку  $\mathbf{X}^0$ , начальный шаг  $\lambda^2$ , значения коэффициентов  $\beta$ ,  $\gamma$  и полагаем счетчик числа итераций  $r=2$ .

Тем или иным способом, например, с помощью одношагового метода наилучшей пробы определяем точки  $\mathbf{X}^1$ ,  $\mathbf{X}^2$  - этап «разгона» метода.

Генерируем  $n$ -мерный случайный вектор  $\mathbf{P}^r$  и по формулам (2), (3) вычисляем координаты точки  $\mathbf{X}^{r+1}$  и значение  $\Phi(\mathbf{X}^{r+1})$  функции  $\Phi(\mathbf{X})$  в этой точке.

Если  $\Phi(\mathbf{X}^{r+1}) < \Phi(\mathbf{X}^r)$ , то проверяем условие окончания итераций (см. ниже). Если

$$\mathbf{X}^* \approx \mathbf{X}^{r+1}$$

условие окончания выполнено, то полагаем  $\mathbf{X}^*$  и завершаем итерации. Если условие окончания итераций не выполнено, то некоторому правилу увеличиваем длину шага  $\lambda^r$ , например, полагая  $\lambda^{r+1}=2\lambda^r$ , принимаем  $r=r+1$  и переходим к п.3. Если  $\Phi(\mathbf{X}^{r+1}) \geq \Phi(\mathbf{X}^r)$ , то переходим к п. 5.

Некоторое фиксированное количество раз делаем попытку, исходя из той же точки  $\mathbf{X}^r$ , не меняя длины шага  $\lambda^r$ , добиться уменьшения значения функции  $\Phi(\mathbf{X})$  путем только изменения вектора  $\mathbf{P}^r$ , т.е., не меняя  $\mathbf{X}^r$  и  $\lambda^r$ , переходим на п. 3. Если это фиксированное количество попыток не привело к успеху, то, исходя из той же точки  $\mathbf{X}^r$

по некоторому правилу уменьшаем длину шага  $\lambda^r$ , например, полагая  $\lambda^r=\frac{\lambda^r}{2}$ , и переходим к п.3 •

В качестве условия окончания поиска можно использовать одно из стандартных условий окончания итераций

$$\|\mathbf{X}^{r+1}-\mathbf{X}^r\| \leq \varepsilon_X,$$

где  $\varepsilon_X$  - константа, определяющая требуемую точность решения по  $\mathbf{X}$ ;

$$|\Phi(\mathbf{X}^{r+1}) - \Phi(\mathbf{X}^r)| \leq \varepsilon_\Phi,$$

где  $\varepsilon_\Phi$  - константа, определяющая требуемую точность решения по  $\Phi$ . Известно множество модификаций рассмотренной простейшей схемы метода повторяющегося случайного поиска. Например, в процессе поиска могут изменяться по некоторым правилам не только длина шага  $\lambda^r$ , но и коэффициенты  $\beta, \gamma$ . Метод повторяющегося случайного поиска иллюстрирует рис. 2, на котором показан фрагмент линий уровня функции Химмельблау.

На рисунке принято  $\bar{\mathbf{S}}^r = \frac{\mathbf{S}^r}{\|\mathbf{S}^r\|}$ ,  $\bar{\mathbf{P}}^r = \frac{\mathbf{P}^r}{\|\mathbf{P}^r\|}$ ,  $\gamma=0.5$ ,  $\beta=0.5$ ,  $\lambda^r=2$ , так что

$$\mathbf{S}^r = 0.5(\mathbf{S}^{r-1} + \mathbf{S}^{r-2}) \text{ и } \mathbf{X}^{r+1} = \mathbf{X}^r + [\bar{\mathbf{S}}^r + \bar{\mathbf{P}}^r].$$

Пунктиром показаны отвергнутые векторы  $\bar{\mathbf{P}}^r$ .

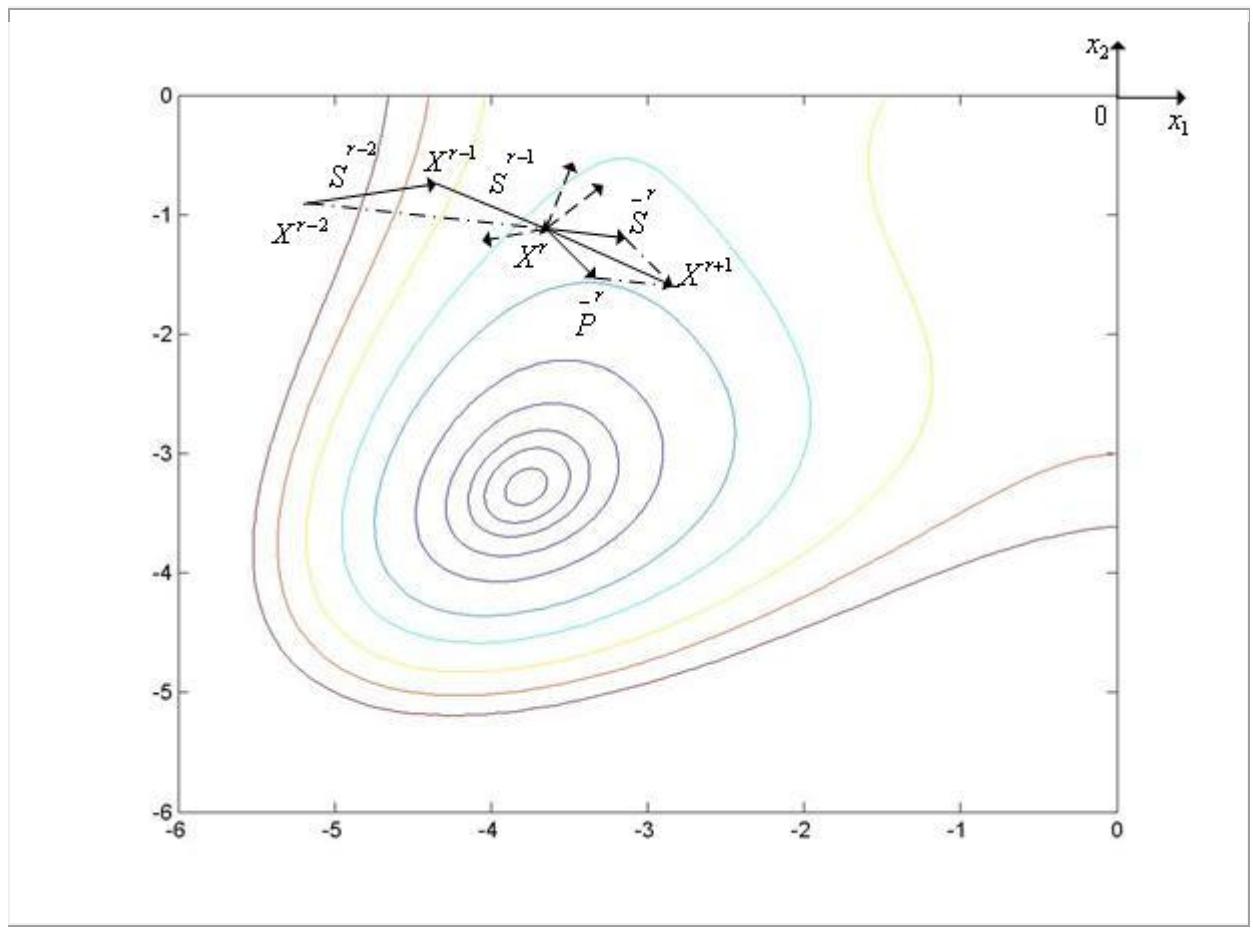


Рис. 2. Траектория поиска минимума функции Химмельблау методом повторяющегося случайного поиска.

#### 8.4. Метод случайного поиска с постоянным радиусом поиска и случайными направлениями

Рассматривается следующая многомерная задача локальной безусловной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного в  $n$ -мерном евклидовом пространстве  $R^n$ ,

$$\min_{\mathbf{X} \in R^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*. \quad (1)$$

Метод случайного поиска с постоянным радиусом поиска и случайными направлениями использует процедуру генерации случайных точек, равномерно распределенных по поверхности гиперсферы в пространстве  $R^n$ . Пусть  $\mathbf{X}^c = (x_1^c, x_2^c, \dots, x_n^c)$  - вектор координат центра гиперсферы,  $\rho$  - радиус гиперсферы,  $\mathbf{R}$  - вектор с началом в точке  $\mathbf{X}^c$  и концом в искомой точке на поверхности гиперсферы,  $\alpha_i, i \in [1, n]$  - углы между вектором  $\mathbf{R}$  и ортами координатных осей  $ox_i, i \in [1, n]$ .

Во введенных обозначениях схема алгоритма генерации случайных точек, равномерно распределенных по поверхности гиперсферы радиуса  $\rho$ , может быть представлена в следующем виде:

генерируем  $n$  случайных чисел, равномерно распределенных в интервале  $[0, 2\pi)$ ;  
вычисляем направляющие косинусы  $\cos \alpha_i, i \in [1, n]$  вектора  $\mathbf{R}$ ;

находим координаты искомой точки  $x_i = x_i^c + \rho \cos \alpha_i, i \in [1, n]$ .

Упрощенная схема метода случайного поиска с постоянным радиусом поиска и случайными направлениями.

Задаем начальную точку  $\mathbf{X}^0$ , начальный радиус гиперсферы  $\rho^0$ , и полагаем счетчик числа итераций  $r=0$ .

Генерируем случайные точки  $\mathbf{X}_i^r, i \in [1, N]$  равномерно распределенные по поверхности гиперсферы радиуса  $\rho^r$  с центром в точке  $\mathbf{X}^r$ . Здесь  $N$  – количество точек – свободный параметр метода.

Вычисляем значения минимизируемой функции  $\Phi(\mathbf{X}_i^r)$  в полученных точках и находим точку, в которой достигается минимальное значение функции  $\Phi(\mathbf{X})$ :  
 $\Phi(\mathbf{X}_k^r) = \min_{i \in [1, N]} \Phi(\mathbf{X}_i^r)$ .

Каким-либо из рассмотренных в главе 4 одномерных методов оптимизации (например, методом Паулла) находим минимум функции  $\Phi(\mathbf{X})$  в направлении  $(\mathbf{X}_k^r - \mathbf{X}^r)$ :  
 $\Phi(\mathbf{X}^{r+1}) = \min_{\lambda \in R^1} \Phi(\mathbf{X}^r + \lambda(\mathbf{X}_k^r - \mathbf{X}^r))$ .

Проверяем условие окончания итераций (см. ниже). Если условие окончания выполнено, то полагаем  $\mathbf{X}^* \approx \mathbf{X}^{r+1}$  и завершаем итерации. Иначе - принимаем  $r=r+1$  и переходим к п.2.

В качестве условия окончания поиска можно использовать одно из стандартных условий окончания итераций:

$$\|\mathbf{X}^{r+1} - \mathbf{X}^r\| \leq \varepsilon_X,$$

где  $\varepsilon_X$  - константа, определяющая требуемую точность решения по  $\mathbf{X}$ ;

$$|\Phi(\mathbf{X}^{r+1}) - \Phi(\mathbf{X}^r)| \leq \varepsilon_\Phi,$$

где  $\varepsilon_\Phi$  - константа, определяющая требуемую точность решения по  $\Phi$ .

Могут быть использованы также другие критерии окончания поиска, например, условие не превышения текущим радиусом гиперсферы величины  $\varepsilon_X$ :

$$r \leq \varepsilon_X.$$

В процессе поиска радиус гиперсферы может меняться, увеличиваясь при удачных шагах (вдали от точки  $\mathbf{X}^*$ ) и уменьшаясь при неудачных шагах (вблизи от точки  $\mathbf{X}^*$ ).

Поиск может быть ускорен, если точки на гиперсфере выбирать (случайным образом) в некотором секторе по отношению к предыдущему направлению. Угол раскрытия этого сектора может меняться в процессе поиска.

Метод случайного поиска с постоянным радиусом поиска и случайными направлениями иллюстрирует рис. 1, на котором показан фрагмент линий уровня функции Химмельблау.

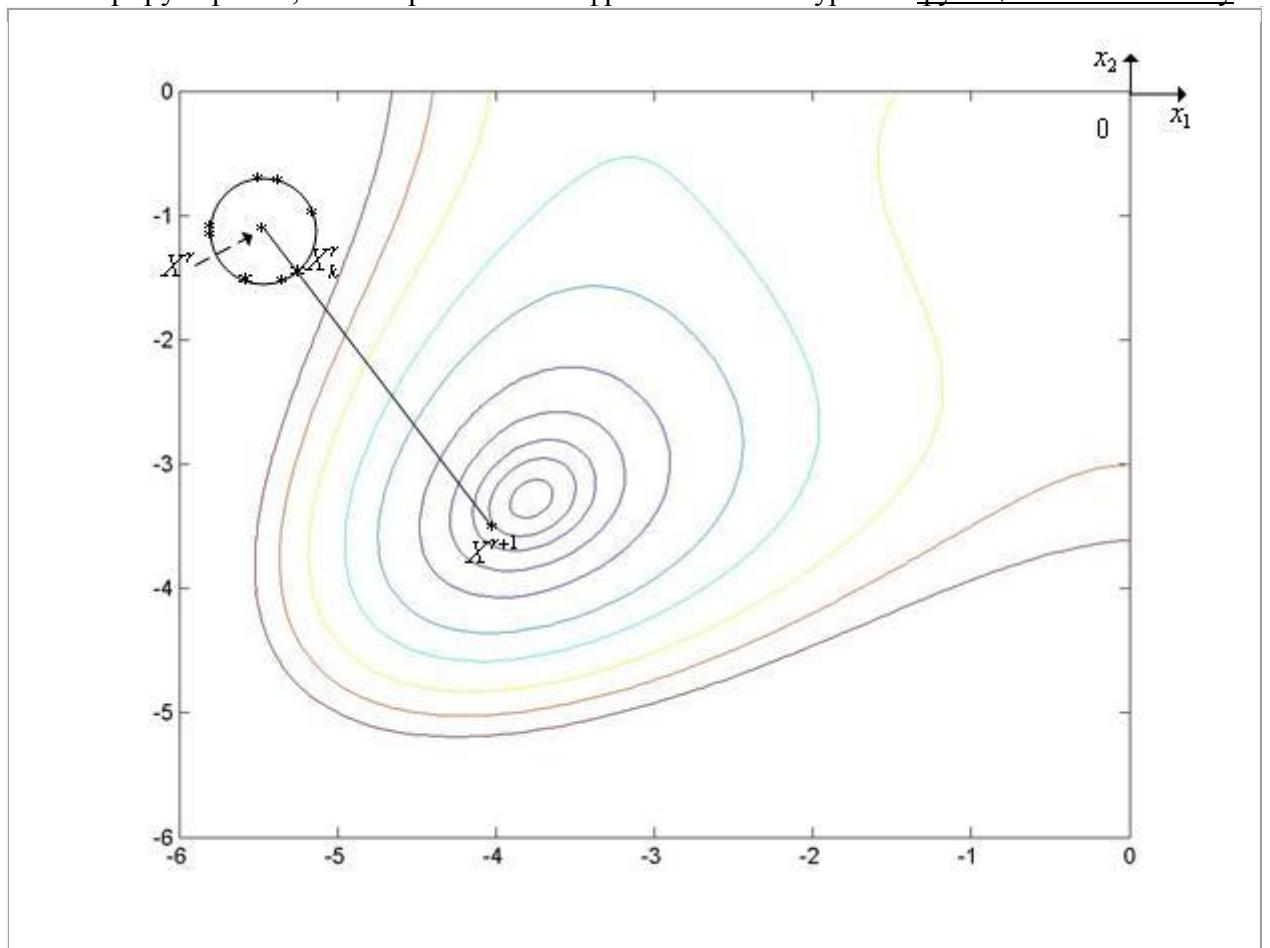


Рис. 1. Траектория поиска минимума функции Химмельблау методом случайного поиска с постоянным радиусом поиска и случайными направлениями ( $n=2$ ).

На рисунке точки, лежащие на окружности с центром в точке  $\mathbf{X}^r$ , соответствуют случайным точкам  $\mathbf{X}_i^r$ ,  $i \in [1, N]$ .

Примечание 1

Одна итерация по методу случайного поиска с постоянным радиусом поиска и случайными направлениями может привести к уменьшению минимизируемой функции в большей степени, чем один шаг поиска в направлении антиградиента этой функции. Данное утверждение иллюстрирует рис. 2, на котором показаны линии уровня двумерной квадратичной функции

$$\Phi(\mathbf{X}) = \Phi(x_1, x_2) = x_1^2 + x_2^2 + 3(x_1 + x_2)^2 = 4x_1^2 + 4x_2^2 + 6x_1x_2$$

(см. параграф 6.4).

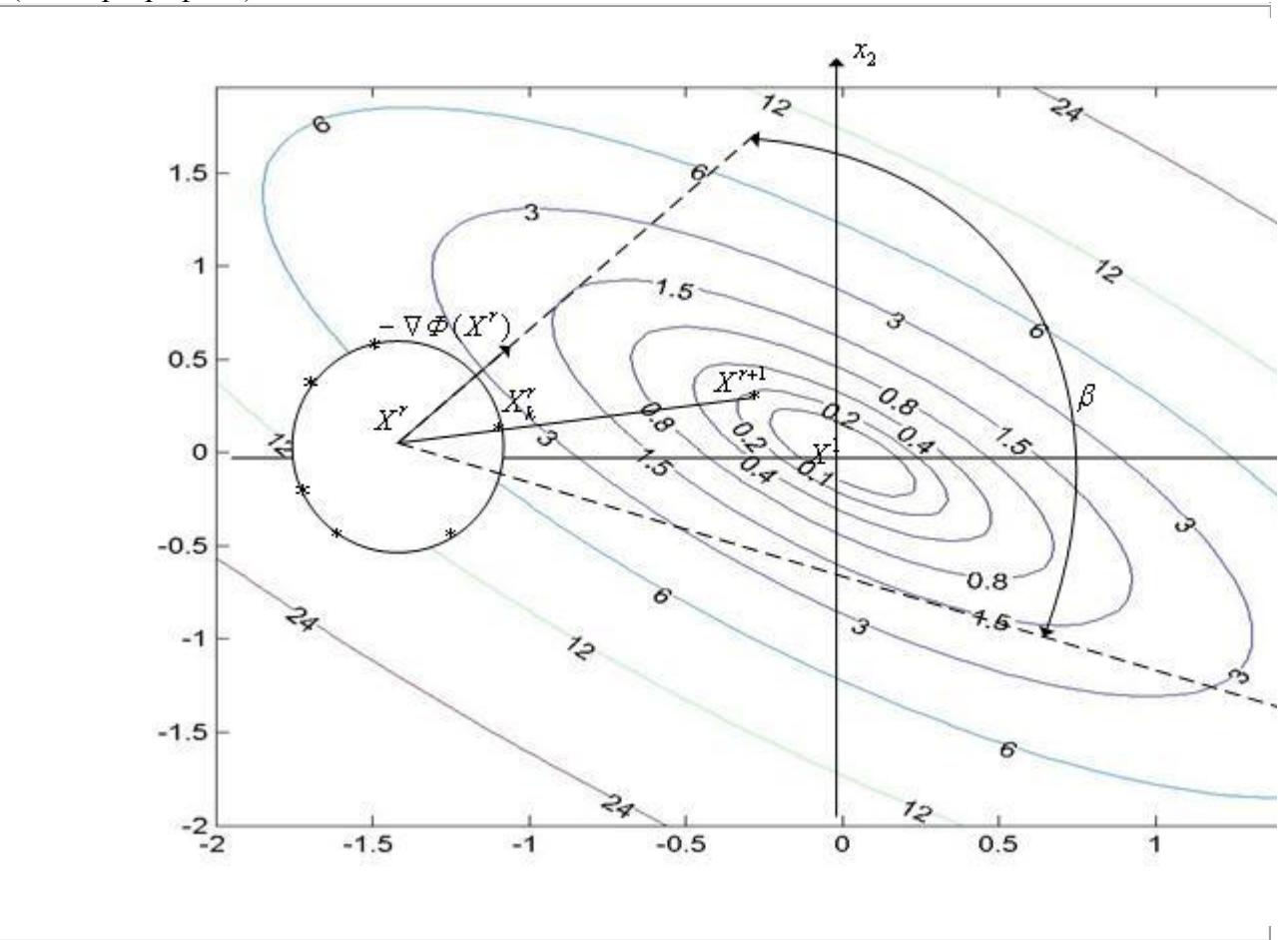


Рис. 2. Один шаг поиска в направлении антиградиента минимизируемой функции ( $\nabla\Phi(X_k)$ ) приводит на линию уровня (1.5). В то же время одна итерация по методу случайного поиска с постоянным радиусом поиска и случайными направлениями – на линию уровня  $\sim 0.2$ . Точки на окружности с центром в точке  $X_k$  соответствуют случайным точкам. Любое направление поиска в секторе  $\beta$  лучше, чем направление антиградиента.

## 9.1. Методы последовательной безусловной оптимизации

Рассмотрим следующую многомерную задачу локальной условной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного во множестве Деклидова пространства  $\mathbb{R}^n$ ,

$$\min_{\mathbf{X} \in D \subset \mathbb{R}^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где множество допустимых значений

$$D = \{\mathbf{X} \mid h_i(\mathbf{X}) = 0, g_j(\mathbf{X}) \geq 0, i \in [1, m], j \in [1, l]\}. \quad (2)$$

Основная идея методов последовательной безусловной оптимизации состоит в преобразовании задачи условной оптимизации (1), (2) к последовательности задач безусловной оптимизации

$$\min_{\mathbf{X} \in \mathbb{R}^n} Q^\alpha(\mathbf{X}) = \min_{\mathbf{X} \in \mathbb{R}^n} (\Phi(\mathbf{X}) + P^\alpha(\mathbf{X})) = Q^\alpha(\tilde{\mathbf{X}}), \quad (3)$$

где  $P^\alpha(\mathbf{X})$  функции, которые возрастают вблизи границ области допустимых значений  $D$  и тем быстрее, чем больше значение параметра  $\alpha$ . В качестве приближенного решения

задачи (1), (2) принимается решение  $\tilde{\mathbf{X}}$  вспомогательной задачи (3) при достаточно большом  $\alpha$ .

Поясним идею методов последовательной безусловной оптимизации примером.

Пример 1

Пусть

$$\Phi(\mathbf{X}) = (x_1 - 3)^2 + (x_2 - 2)^2 \quad (4)$$

и имеется одно ограничение типа равенств с ограничивающей функцией

$$h_1(\mathbf{X}) = x_1 + x_2 - 4. \quad (5)$$

Положим

$$P^\alpha(\mathbf{X}) = \alpha h_1^2(\mathbf{X}), \quad (6)$$

где  $\alpha \geq 0$  — вещественная константа. На рисунках рис. 1, рис. 2, рис. 3 приведены линии уровня функции при  $\alpha = 0, 1, 2$ , соответственно.

Линии уровня на рисунках рис. 1, рис. 2, рис. 3 получены с помощью следующей MATLAB-программы:

```
x = 0 : 0.01 : 5;
y = x;
[X, Y] = meshgrid(x);
alpha = 2;
Z = (X - 3)^2 + (Y - 2)^2 + alpha * (X + Y - 4)^2;
V = [.02, 2, 1, 2, 5, 10, 20, 40, 80];
[C, h] = contour(X, Y, Z, V);
clabel(C, h);
```

Рисунки показывают, что при увеличении параметра  $\alpha$  минимум функции  $Q^\alpha(\mathbf{X})$  приближается к решению задачи (3), (4), (5), (6)

Среди методов последовательной безусловной оптимизации выделяют метод штрафных функций и метод барьерных функций.

В методе штрафных функций функцию  $P^\alpha(\mathbf{X})$ , которая в этом случае называется штрафной функцией, подбирают таким образом, чтобы при больших  $\alpha$  функция  $Q^\alpha(\mathbf{X})$

мало отличалась от функции  $\Phi(\mathbf{X})$  при  $\mathbf{X} \in D$  и быстро возрастала при удалении точки  $\mathbf{X} \notin D$  от границы области допустимых значений  $D$ . В методе штрафных функций точка  $\mathbf{X}$  в процессе поиска может выходить за границы области  $D$  (см. рис. 4). Т.е. метод штрафных функций относится к классу методов внешней точки. Рассмотренный выше прим. 1 также иллюстрирует метод штрафных функций.

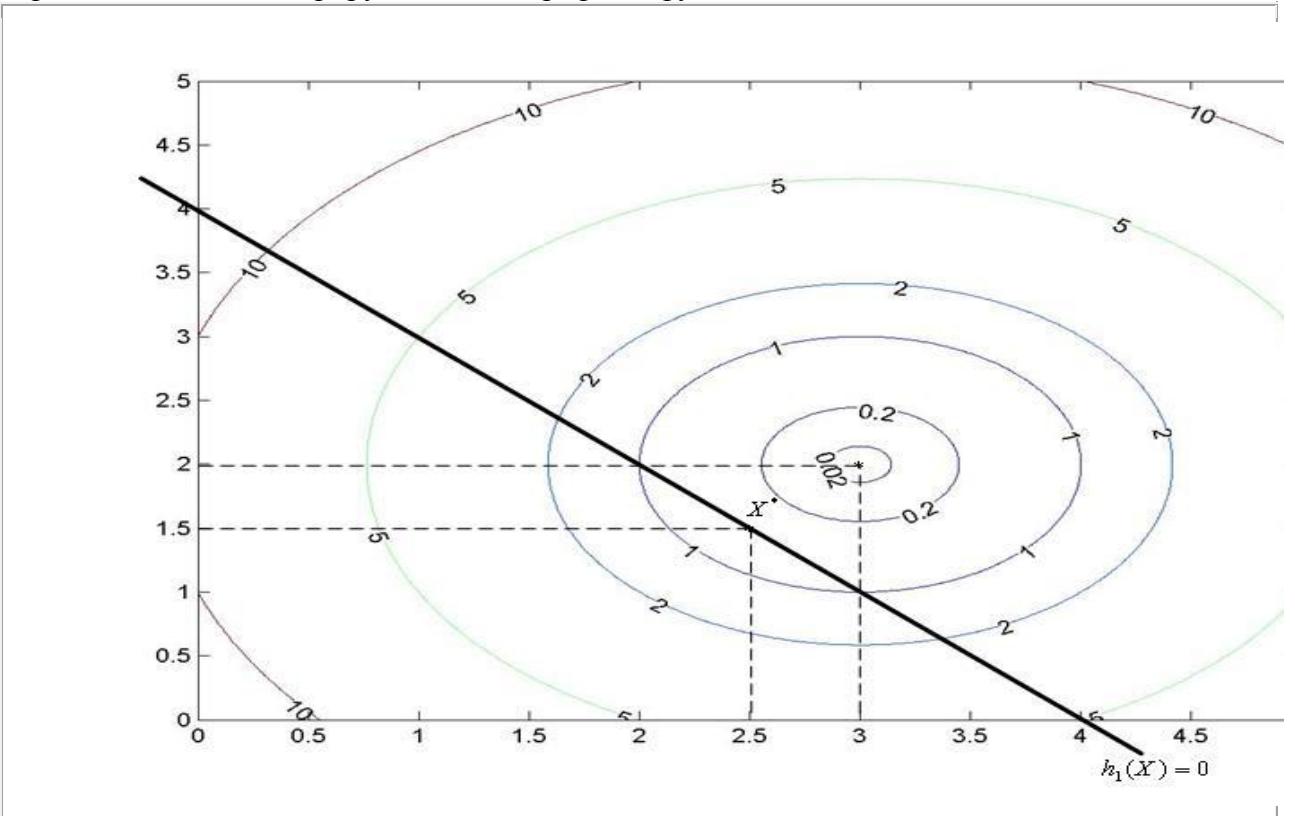


Рис. 1. К прим. 1. Точка минимума функции  $Q\alpha(\mathbf{X})$  при  $\alpha = 0$  имеет координаты  $(3, 2)$ . Решением задачи  $(3), (4), (5), (6)$  является точка  $X^*$  с координатами  $(2.5, 1.5)$ .

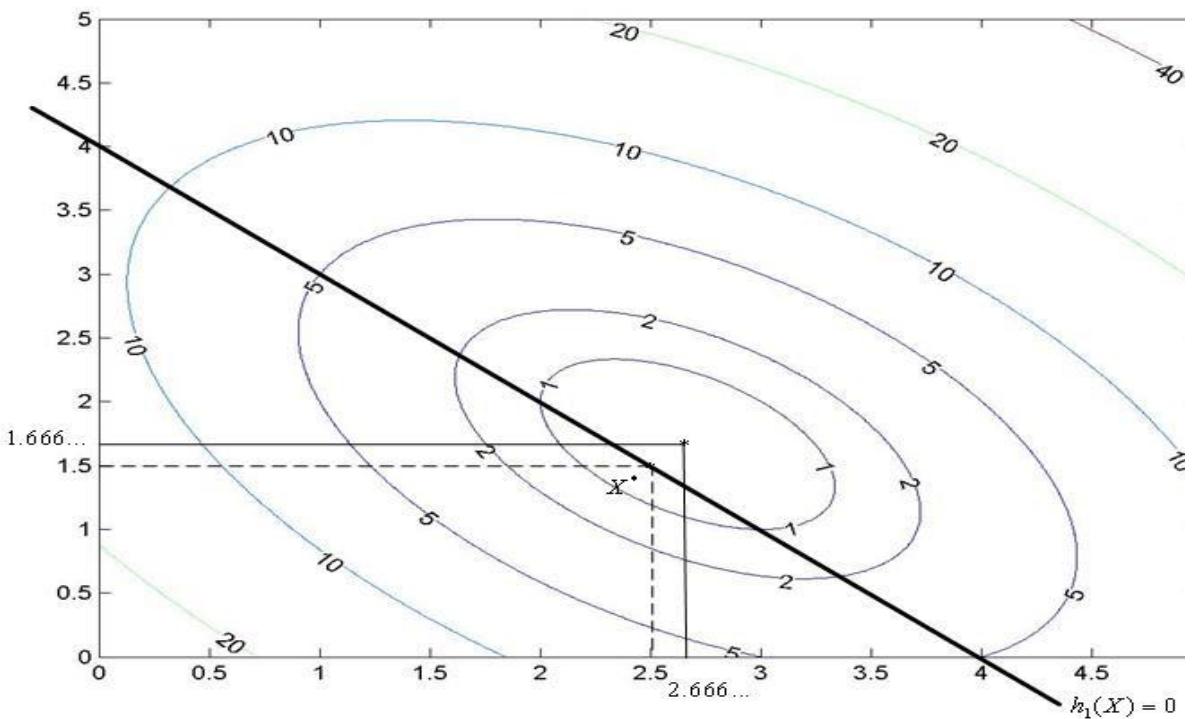


Рис. 2. К прим. 1. Точка минимума функции  $Q_\alpha(X)$  при  $\alpha = 1$  имеет координаты  $(2.666\dots, 1.666\dots)$ . Решением задачи (3), (4), (5), (6) является точка  $X^*$  с координатами  $(2.5, 1.5)$ .

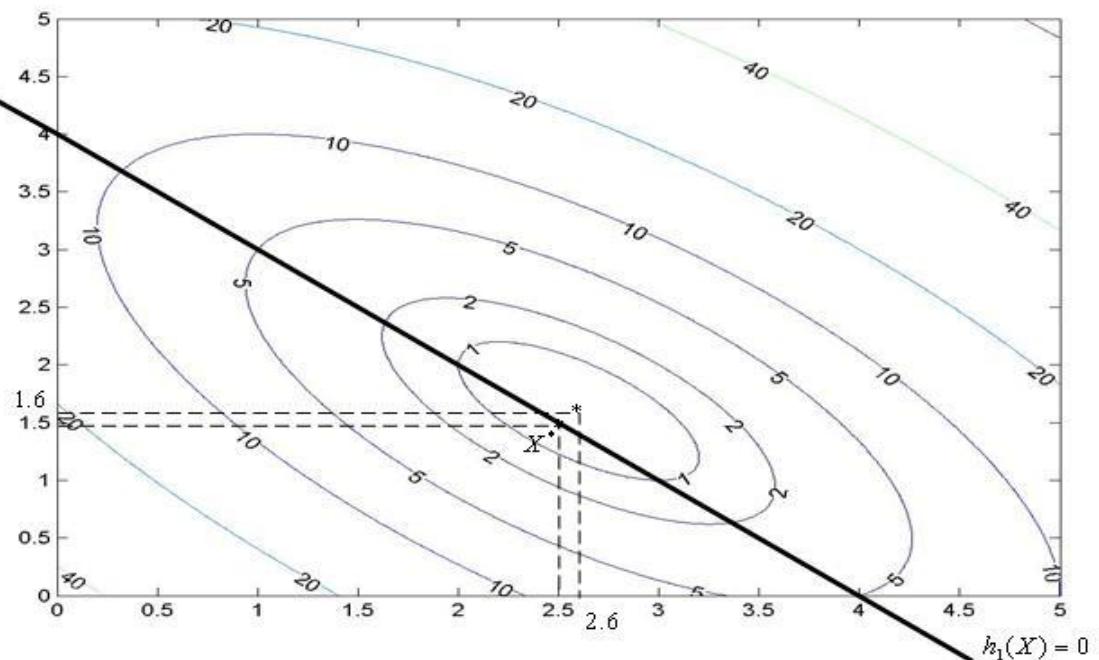


Рис. 3. К прим. 1. Точка минимума функции  $Q_\alpha(X)$  при  $\alpha = 2$  имеет координаты  $(2.6, 1.6)$ . Решением задачи (3), (4), (5), (6) является точка  $X^*$  с координатами  $(2.5, 1.5)$ .

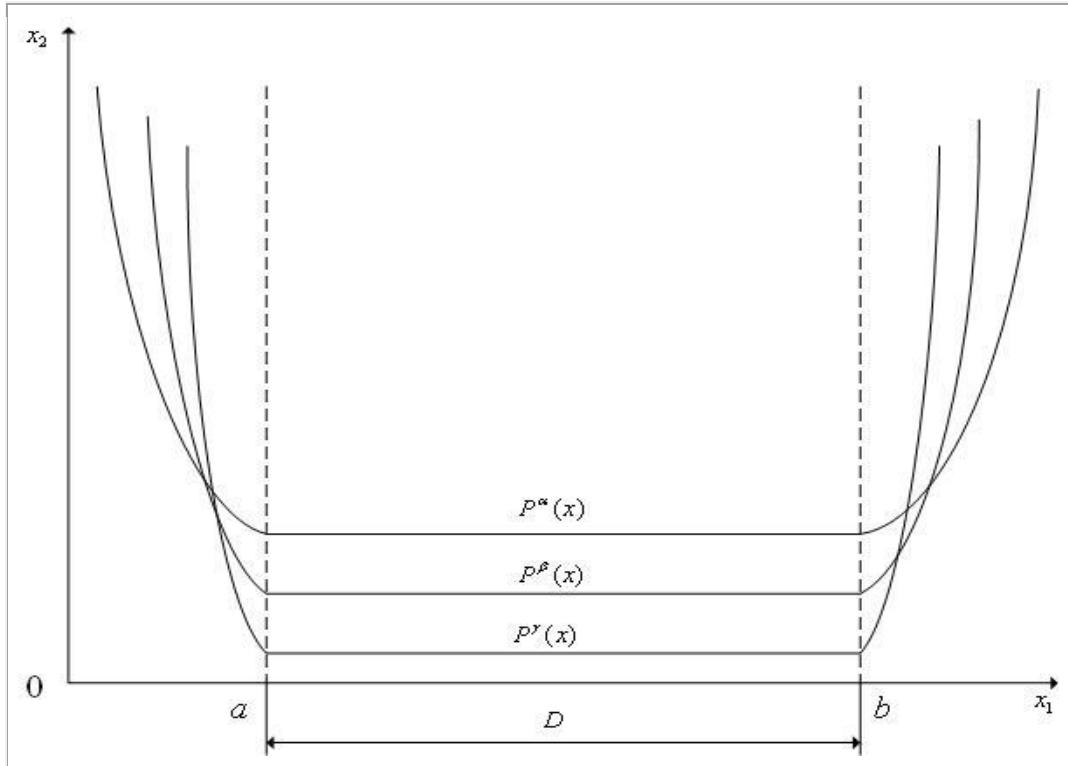


Рис. 4. К методу штрафных функций ( $n = 1$ ) Интервал  $[a, b]$  — область допустимых значений  $D$ ;  $\gamma > \beta > \alpha$ .

В методе барьерных функций функцию  $P^\alpha(\mathbf{X})$ , которая в этом случае называется барьерной функцией, подбирают таким образом, чтобы при больших  $\alpha$  функция  $Q^\alpha(\mathbf{X})$  мало отличалась от функции  $\Phi(\mathbf{X})$  при  $\mathbf{X} \in D$  и быстро возрастала при приближении точки  $\mathbf{X} \in D$  к границе области допустимых значений  $D$ . В методе барьерных функций точка  $\mathbf{X}$  в процессе поиска не может выходить за границы области  $D$  (см. рис. 5). Это означает, что метод барьерных функций относится к классу методов внутренней точки.

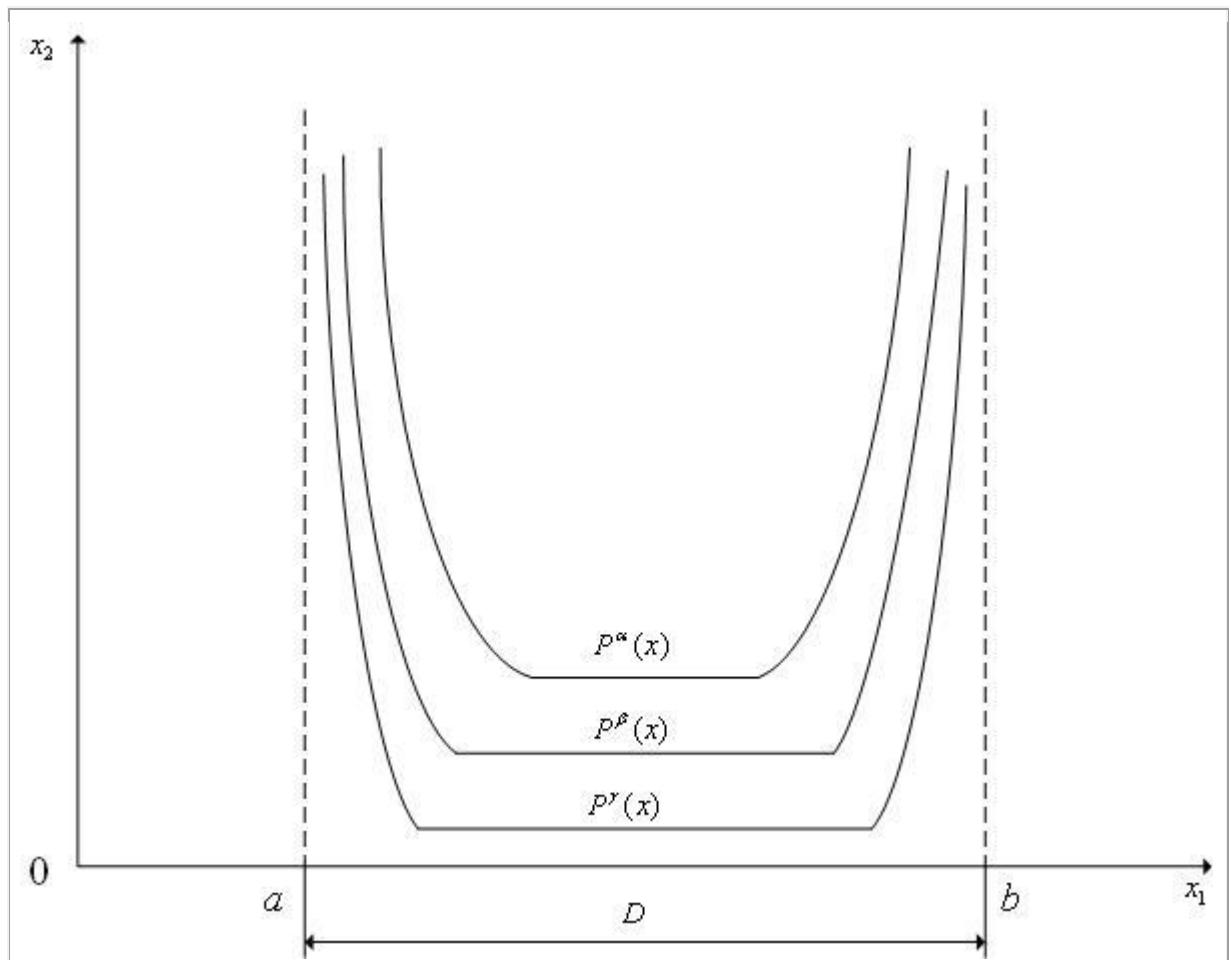


Рис. 5. К методу барьерных функций ( $n = 1$ ) Интервал  $[a, b]$  — область допустимых значений  $D$ ;  $\gamma > \beta > \alpha$ .

В вычислительной практике преимущественно используется метод штрафных функций. Поэтому в дальнейшем ограничимся именно им.

Штрафная функция в общем случае имеет вид

$$P(\mathbf{X}, \alpha_h^r, \alpha_g^r) = \alpha_h^r \sum_{i=1}^m \mu_i H(h_i(\mathbf{X})) + \alpha_g^r \sum_{j=1}^l \lambda_j G(g_j(\mathbf{X})), \quad (7)$$

где  $\alpha^r = (\alpha_h^r, \alpha_g^r)$  — двумерный вектор параметров штрафной функции;  $\mu_i, \lambda_j$  — весовые коэффициенты, могущие изменяться в процессе итераций,  $H(h_i(\mathbf{X}))$ ,  $G(g_j(\mathbf{X}))$  — функционалы над функциями  $h_i(\mathbf{X}) \in [1, m]$ ,  $g_j(\mathbf{X}) \in [1, l]$ , соответственно.

Функционалы  $H(h_i(\mathbf{X}))$ ,  $G(g_j(\mathbf{X}))$  в формуле (7) должны удовлетворять очевидным требованиям:

$$H(h_i(\mathbf{X})) \rightarrow 0 \text{ при } h_i(\mathbf{X}) \rightarrow 0,$$

$$G(g_j(\mathbf{X})) \rightarrow 0 \text{ при } g_j(\mathbf{X}) \rightarrow 0;$$

$$\lim_{r \rightarrow \infty} (\alpha_h^r \sum_{i=1}^m \mu_i H(h_i(\mathbf{X}))) = \begin{cases} 0, & \mathbf{X} \in D, \\ \infty, & \mathbf{X} \notin D. \end{cases}$$

$$\lim_{r \rightarrow \infty} (\alpha_g^r \sum_{j=1}^l \lambda_j G(g_j(\mathbf{X}))) = \begin{cases} 0, & \mathbf{X} \in D, \\ \infty, & \mathbf{X} \notin D. \end{cases}$$

В качестве функционалов  $H(h_i(\mathbf{X}))$ ,  $G(g_j(\mathbf{X}))$  можно взять расстояния в какой-либо метрике от точки  $\mathbf{X}$  до соответствующей границы множества  $D$ . Однако, вычисление этих расстояний, а значит и значений штрафной функции, может быть затруднительным. Поэтому обычно применяют штрафные функции более удобного вида.

Так в качестве функционалов  $H(h_i(\mathbf{X}))$  обычно используют функционалы

$$H(h_i(\mathbf{X})) = h_i^2(\mathbf{X}),$$

в качестве функционалов  $G(g_j(\mathbf{X}))$  — функционалы

$$G(g_j(\mathbf{X})) = (g_j^+(\mathbf{X}))^2,$$

$$g_j^+(\mathbf{X}) = \begin{cases} 0, & g_j(\mathbf{X}) \geq 0, \\ g_j(\mathbf{X}), & g_j(\mathbf{X}) < 0. \end{cases}$$

В качестве критерия окончания итераций в методе последовательной безусловной оптимизации можно использовать неравенство

$$\|\mathbf{X}^r - \mathbf{X}^{r/2}\| \leq \varepsilon_x, \quad (8)$$

где  $r$  — четное число итераций,  $\varepsilon_x$  — требуемая точность решения по  $\mathbf{X}$ .

Недостатком метода последовательной безусловной оптимизации является значительное усложнение структуры минимизируемой функции (см. рис. 1) — плата за исключение ограничений.

Схема метода штрафных функций.

Задаем начальную точку  $\mathbf{X}^0$  и полагаем счетчик числа итераций  $r = 0$ .

Исходя из точки  $\mathbf{X}^r$ , одним из методов локальной безусловной оптимизации решаем задачу — находим точку  $\mathbf{X}^{r+1}$ .

Проверяем условие окончания поиска (8). Если условие окончания поиска выполнено, то полагаем  $\mathbf{X}^* = \mathbf{X}^{r+1}$  и завершаем итерации. Иначе — по некоторому правилу

увеличиваем значения параметров  $a_h^r$ ,  $a_h^r$ , полагаем  $r = r + 1$  и переходим к п.3 •

Примечание 1

В зависимости от метода локальной безусловной оптимизации, который используется для решения задач (3), метод последовательной безусловной оптимизации может быть детерминированным и случайным, нулевого, первого или второго порядка.

## 9.2. Метод скользящего допуска

Рассмотрим следующую многомерную задачу условной локальной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного во множестве Девклидова пространства  $\mathbb{R}^n$ ,

$$\min_{\mathbf{X} \in D \subset \mathbb{R}^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где множество допустимых значений

$$D = \{\mathbf{X} \mid g_j(\mathbf{X}) \geq 0, j \in [1, l]\}. \quad (2)$$

Основы метода скользящего допуска.

Метод скользящего допуска существенно использует множество

$$D^r = \{\mathbf{X} \mid T(\mathbf{X}) \leq \Delta^r\}. \quad (3)$$

где неотрицательный скаляр  $\Delta^r$  — критерий скользящего допуска,  $T(\mathbf{X})$  — неотрицательно определенный функционал над множеством всех ограничивающих функций  $g_j, j \in [1, l]$ .

При этом функционал  $T(\mathbf{X})$  должен быть сконструирован таким образом, чтобы  $T(\mathbf{X}) = 0$  при  $\mathbf{X} \in D$  и значение  $T(\mathbf{X})$  возрастало по мере удаления точки  $\mathbf{X}$  от границы области допустимых значений  $D$ . Критерий скользящего допуска  $\Delta^r$  определяет требуемую точность выполнения ограничений, которые формируют область допустимых значений  $D$ , и конструируется таким образом, чтобы обеспечить его уменьшение с ростом количества итераций  $r$ .

Точка  $\mathbf{X}$  называется допустимой точкой, если  $T(\mathbf{X}) = 0$ , почти допустимой точкой — если  $0 < T(\mathbf{X}) \leq \Delta^r$ , недопустимой точкой — если  $T(\mathbf{X}) > \Delta^r$ . Поскольку величина  $\Delta^r$  с ростом номера итерации уменьшается, отклонение от границы области  $D$ , при котором точка считается допустимой, сужается, так что в пределе рассматриваются только допустимые точки.

Метод скользящего допуска может быть скомбинирован со многими из рассмотренных ранее многомерных методов локальной безусловной оптимизации. Будем называть метод, с которым комбинируется метод скользящего допуска, базовым методом.

Одна итерация метода скользящего допуска состоит из одного или двух этапов:

1) С помощью базового метода, исходя из сточки  $\tilde{\mathbf{X}}^{r+1}$ , выполняем итерацию по решению задачи локальной безусловной оптимизации

$$\min_{\mathbf{X} \in \mathbb{R}^n} \Phi(\mathbf{X}) = \Phi(\tilde{\mathbf{X}}^{r+1}) \quad (4)$$

- находим точку  $\tilde{\mathbf{X}}^{r+1}$ . Если  $T(\tilde{\mathbf{X}}^{r+1}) \leq \Delta^{r+1}$  (точка  $\tilde{\mathbf{X}}^{r+1}$  является допустимой точкой или почти допустимой точкой), то полагаем  $\mathbf{X}^{r+1} = \tilde{\mathbf{X}}^{r+1}$  и заканчиваем данную итерацию.

2) Если  $T(\tilde{\mathbf{X}}^{r+1}) > \Delta^{r+1}$  (точка  $\tilde{\mathbf{X}}^{r+1}$  является недопустимой), то отыскиваем точку  $\mathbf{X}^{r+1}$ , лежащую ближе к границе области  $D$ . Для этого с помощью того же базового метода, исходя из точки  $\tilde{\mathbf{X}}^{r+1}$ , решаем задачу локальной безусловной оптимизации

$$\min_{\mathbf{X} \in \mathbb{R}^n} T(\mathbf{X}) = T(\mathbf{X}^{r+1}) \quad (5)$$

с условием окончания итераций

$$T(\mathbf{X}^{r+1}) \leq \Delta^{r+1} \quad (6)$$

и заканчиваем данную итерацию.

Достоинством метода скользящего допуска является то, что степень нарушения ограничений по мере приближения к минимуму минимизируемой функции постепенно уменьшается. Т.е. на первых итерациях ограничения могут удовлетворяться приближенно, а высокая точность удовлетворения ограничений необходима лишь в окрестности решения. Это обстоятельство позволяет сократить полный объем вычислений по сравнению с другими методами.

Одна из сложностей применения метода скользящего допуска — возможные осцилляции решения относительно границы области  $D$  (см. ниже).

Комбинация метода скользящего допуска с методом Нелдера-Мида.

При комбинации метода скользящего допуска с методом Нелдера-Мида можно предложить разные виды критерия скользящего допуска. Чаще всего в качестве этого критерия используют следующую функцию координат вершин деформируемого многогранника  $(\mathbf{X}_1^r, \mathbf{X}_2^r, \dots, \mathbf{X}_{n+1}^r)$ :

$$\left\{ \begin{array}{l} \Delta^r = \min \left\{ \Delta^{r-1}, \frac{1}{n+1} \sum_{i=1}^{n+1} |\mathbf{X}_i^r - \mathbf{X}_c^r| \right\} \\ \Delta^0 = \frac{1}{n+1} \sum_{i=1}^{n+1} |\mathbf{X}_i^0 - \mathbf{X}_c^0| \end{array} \right. \quad (7)$$

Здесь  $\mathbf{X}_c^r$  — вектор координат центра тяжести многогранника  $(\mathbf{X}_1^r, \mathbf{X}_2^r, \dots, \mathbf{X}_{n+1}^r)$ , так что величина

$$\frac{1}{n+1} \sum_{i=1}^{n+1} |\mathbf{X}_i^r - \mathbf{X}_c^r|$$

есть среднее расстояние вершин многогранника  $(\mathbf{X}_1^r, \mathbf{X}_2^r, \dots, \mathbf{X}_{n+1}^r)$  от его центра тяжести.

Из (7) следует, что критерий скользящего допуска  $\Delta^r$  является положительно определенной функцией координат вершин многогранника  $\Delta^r = \Delta^r(\mathbf{X}_1^r, \mathbf{X}_2^r, \dots, \mathbf{X}_{n+1}^r)$ . С другой стороны, поскольку размер многогранника при приближении к точке минимума  $\mathbf{X}^*$  уменьшается (в пределе до нуля), то справедливо предельное соотношение

$$\lim_{r \rightarrow \infty} \Delta^r = 0$$

Для задачи (1), (2) в качестве функционала  $T(\mathbf{X})$  обычно используют функционал

$$T(\mathbf{X}) = \sum_{j=1}^1 (g_j^+(\mathbf{X}))^2 \quad (8)$$

где

$$g_j^+(\mathbf{X}) = \begin{cases} 0, & g_j(\mathbf{X}) \geq 0, \\ g_j(\mathbf{X}), & g_j(\mathbf{X}) < 0. \end{cases}$$

Из (8) следует, что функционал  $T(\mathbf{X})$  обладает следующим свойством

$$\begin{cases} T(\mathbf{X}) = 0, & \mathbf{X} \in D, \\ T(\mathbf{X}) > 0, & \mathbf{X} \notin D. \end{cases}$$

Из (8) вытекает также, что если значение функционала  $T(\mathbf{X})$  мало, то точка  $\mathbf{X}$  находится недалеко от границы области  $D$ .

Примечание 1

Поскольку метод Нелдера-Мида является детерминированным методом нулевого порядка, комбинация метода скользящего допуска с методом Нелдера-Мида также представляет собой детерминированный метод нулевого порядка.

Упрощенная схема комбинации метода скользящего допуска и метода Нелдера-Мида.

Симплекс с вершинами  $\mathbf{X}_i^r, i \in [1, n+1]$  обозначим  $S^r$ .

Задаем начальный симплекс  $S^0$  и полагаем счетчик числа итераций  $r = 0$ .

С помощью метода Нелдера-Мида, исходя из симплекса  $S^r$ , выполняем одну итерацию по решению задачи локальной безусловной оптимизации (4) — находим симплекс  $\tilde{S}^{r+1}$  с вершинами  $\tilde{\mathbf{X}}_i^r, i \in [1, n+1]$ .

Вычисляем значения функционала  $T(\mathbf{X})$  во всех вершинах симплекса  $\tilde{S}_i^{r+1}$  и значение критерия скользящего допуска  $\Delta^r$ . Находим вершину симплекса  $\tilde{\mathbf{X}}_T^{r+1}$ , в которой значение функционала  $T(\mathbf{X})$  максимально, т.е. вершину, которая расположена дальше всех от границы области  $D$ . Обозначим эту вершину  $\tilde{\mathbf{X}}_T^{r+1}$ .

$$T(\tilde{\mathbf{X}}_T^{r+1}) \leq \Delta^{r+1} \quad \tilde{\mathbf{X}}_T^{r+1}$$

Если  $(\text{точка } \tilde{\mathbf{X}}_T^{r+1} \text{ является допустимой точкой или почти допустимой точкой})$ , то проверяем условие окончания поиска (см. схему метода Нелдера-Мида). Если это условие выполнено, то завершаем итерации. Если условие окончания поиска не

выполнено, то формируем симплекс  $S^{r+1}$  с вершинами  $\tilde{\mathbf{X}}_i^{r+1}, i \in [1, n+1]$ , полагаем  $r = r + 1$  и переходим к п.2.

$$T(\tilde{\mathbf{X}}_T^{r+1}) > \Delta^{r+1} \quad \tilde{\mathbf{X}}_T^{r+1}$$

Если  $(\text{точка } \tilde{\mathbf{X}}_T^{r+1} \text{ является недопустимой точкой})$ , то с помощью метода Нелдера-Мида, исходя из точки  $\tilde{\mathbf{X}}_T^{r+1}$ , решаем задачу локальной безусловной

оптимизации (5) с критерием окончания итераций (6) — находим точку  $\mathbf{X}_T^{r+1}$ .

Формируем новый симплекс  $S^{r+1}$  с вершинами  $\mathbf{X}_i^{r+1} = \tilde{\mathbf{X}}_i^{r+1}, i \in [1, n+1], i \neq T, \mathbf{X}_T^{r+1}$ , полагаем  $r = r + 1$  и переходим к п.2 •

Ослабление осцилляций решения

Как отмечалось выше, одной из сложностей применения метода скользящего допуска являются возможные осцилляции решения относительно границы области  $D$ . Поясним суть этого явления на примере.

Пример 1

Рассмотрим двумерную задачу условной оптимизации (1), когда критерий оптимальности равен

$$\Phi(\mathbf{X}) = -x_1 - x_2$$

и множество допустимых значений  $D$  определяется ограничениями

$$\begin{cases} g_1(\mathbf{X}) = -x_1^2 - x_2^2 + 9 \geq 0, \\ g_2(\mathbf{X}) = -x_1 \geq 0, \\ g_3(\mathbf{X}) = -x_2 \geq 0. \end{cases} \quad (9)$$

Положим, что на  $r$ -ой итерации координаты вершин текущего симплекса  $S^r$  равны  $\mathbf{X}_1^r, \mathbf{X}_2^r, \mathbf{X}_3^r$ . Тогда после одной итерации по решению задачи локальной безусловной оптимизации (4) методом Нелдера-Мида получим симплекс  $\tilde{S}^{r+1}$  с вершинами  $\tilde{\mathbf{X}}_i^{r+1}, i \in [1, n+1]$  (см. рис. 1).

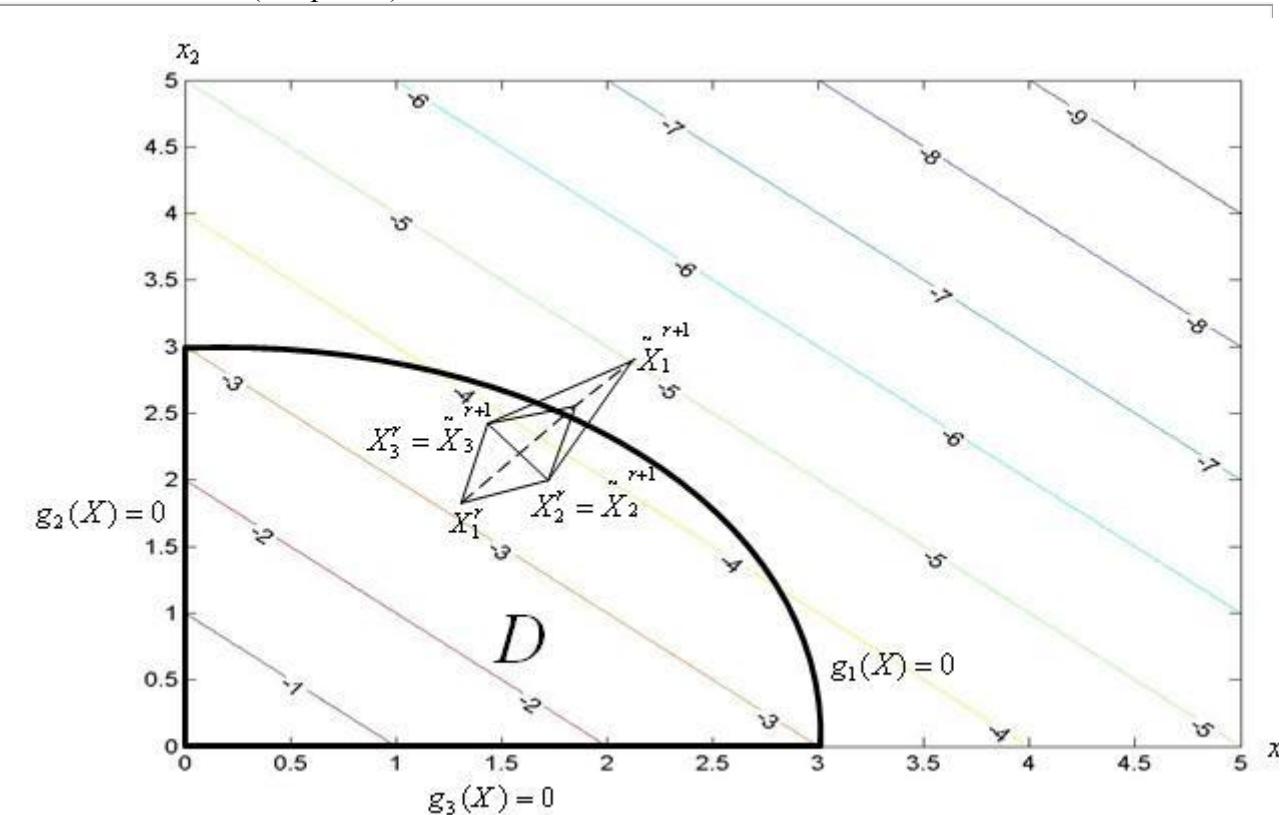


Рис. 1. К прим. 1. После успешного отражения вершины  $X_{r1}$  выполнено успешное растяжение симплекса.

Линии уровня функции  $\Phi(\mathbf{X})$  на рис. 1 получены с помощью следующей MATLAB-программы:

```
x = 0 : 0.15 : 5;
y = x;
[X, Y] = meshgrid(x);
Z = -X - Y;
V = [-1, -2, -3, -4, -5, -6, -7, -8, -9, -10];
[C, h] = contour(X, Y, Z, V);
clabel(C, h);
```

$$\tilde{\mathbf{X}}_1^{r+1}$$

Положим далее, что точка  $\tilde{\mathbf{X}}_1^{r+1}$ , расположенная далее всех от границы области  $D$ , является недопустимой точкой, т.е.  $T(\tilde{\mathbf{X}}_1^{r+1}) > \Delta^{r+1}$ . Тогда при решении помошью метода Нелдера-Мида задачи локальной безусловной оптимизации (5) возможна ситуация, приведенная на рис. 2.

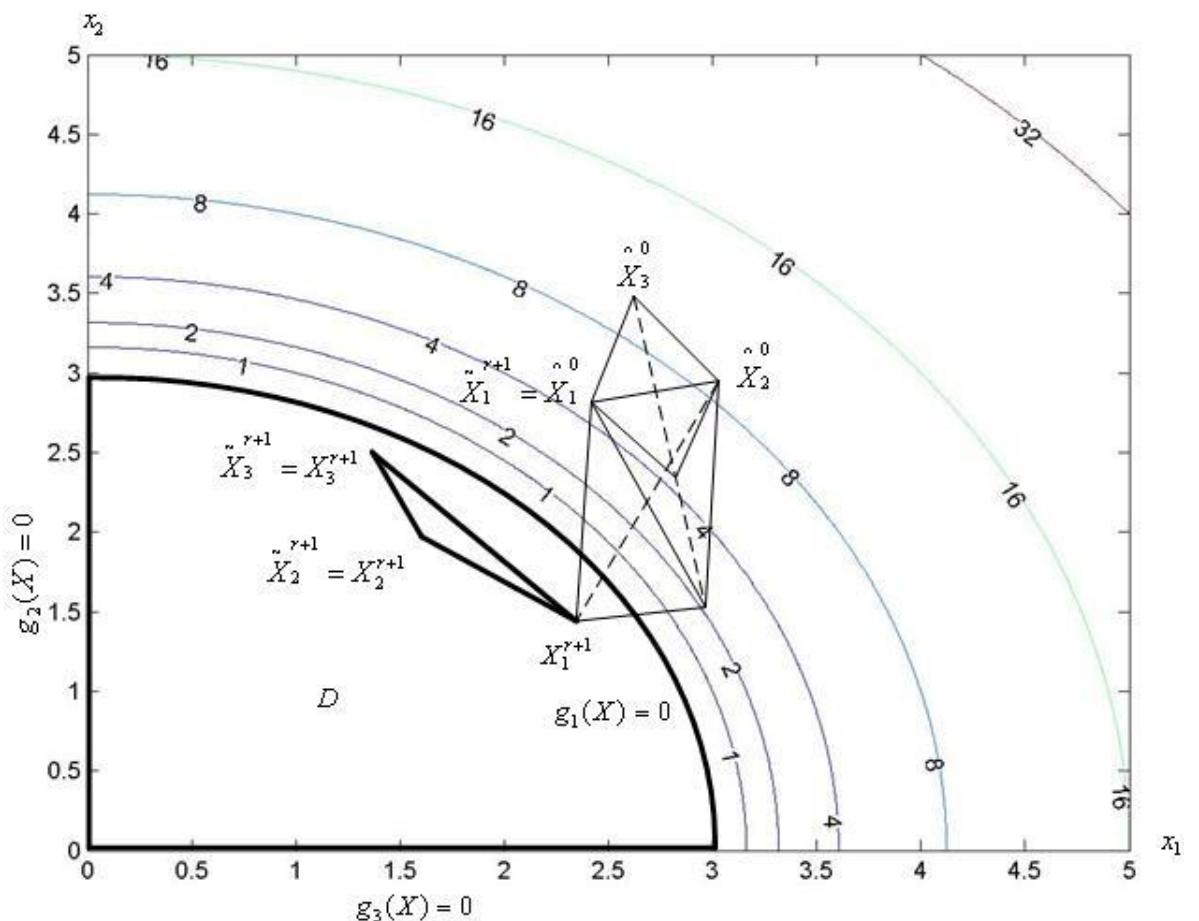


Рис. 2. К прим. 1. После успешного отражения вершины  $X_{03}$  выполнено растяжение симплекса и отражение вершины  $X_{02}$ .

Из (8), (9) следует, что если  $\mathbf{X} \notin D$  и точка  $\mathbf{X}$  лежит в первой четверти системы координат  $0 \leq x_1, x_2$ , то  $T(\mathbf{X}) = x_1^2 + x_2^2 - 9$ . На рис. 2 показаны линии уровня функции  $T(\mathbf{X})$  именно для этого случая. Линии уровня получены с помощью следующей MATLAB-программы:

```
x = 0 : 0.01 : 5;
y = x;
[X, Y] = meshgrid(x);
```

```

Z = X.^2 + Y.^2 - 9.;
V =[1, 2, 4, 8, 16, 32, 64, 128];
[C, h] = contour(X, Y, Z, V);
clabel(C, h);

```

Рассмотренный пример иллюстрирует тот факт, что поскольку вершина  $\tilde{\mathbf{X}}_1^r$  симплекса  $S^r$  расположена далеко от границы области  $D$ , то после операций отражения и растяжения

$$\tilde{\mathbf{X}}_1^{r+1}$$

точка  $\tilde{\mathbf{X}}_1^{r+1}$  может оказаться глубоко в недопустимой области. В результате в процессе минимизации функционала  $T(\mathbf{X})$  может получиться точка  $\tilde{\mathbf{X}}_T^{r+1}$ , которая снова оказывается далеко от границы области  $D$ . И т.д.

Эффект, рассмотренный в прим. 1, и называется осцилляцией решения относительно границы области  $D$ .

$$\tilde{\mathbf{X}}_T^{r+1}$$

Для ослабления влияния осцилляций в простейшем случае можно вместо точки

$$[\mathbf{X}_T^r, \tilde{\mathbf{X}}_T^{r+1}]$$

использовать точку  $A^r$  — середину отрезка

Чаще с этой целью используют квадратичную интерполяцию функции  $T(\mathbf{X})$  на отрезке  $[\mathbf{X}_T^r, \tilde{\mathbf{X}}_T^{r+1}]$  по трем точкам  $(\mathbf{X}_T^r, \Phi(\mathbf{X}_T^r)), (A^r, \Phi(A^r)), (\tilde{\mathbf{X}}_T^{r+1}, \Phi(\tilde{\mathbf{X}}_T^{r+1}))$ , где  $A^r$  —

$$[\mathbf{X}_T^r, \tilde{\mathbf{X}}_T^{r+1}]$$

также середина отрезка — см. рис. 3. Обозначим эту интерполирующую

$$\tilde{\mathbf{X}}_T^{r+1}$$

функцию  $y(x) = \alpha x^2 + \beta x + \gamma$  (см. параграф 4.7). Вместо точки  $\tilde{\mathbf{X}}_T^{r+1}$  в этом случае можно использовать один из нулей функции  $y(x)$  либо его приближенное значение, найденное, например, методом касательных.

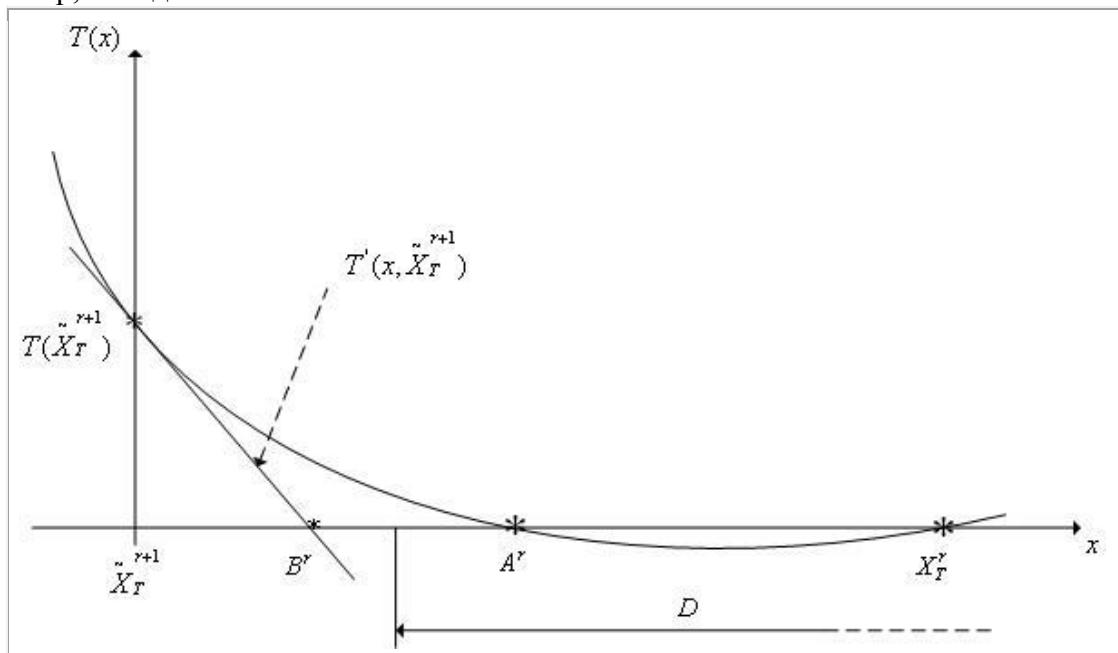


Рис. 3. Использование квадратичной интерполяции функции  $T(\mathbf{X})$  на отрезке по трем точкам для ослабления осцилляций. Случай, когда точка  $A^r$  принадлежит области допустимых значений  $D$ .

### 9.3. Модифицированный метод комплексов

Рассмотрим многомерную задачу локальной условной оптимизации

$$\min_{\mathbf{X} \in D \subset \mathbb{R}^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где множество допустимых значений определяется только ограничениями типа неравенств и представляет собой гиперпараллелепипед, т.е.

$$D = \{\mathbf{X} \mid \mathbf{x}_i^U \geq \mathbf{x}_i \geq \mathbf{x}_i^L, i \in [1, n]\}. \quad (2)$$

Здесь  $\mathbf{x}_i^L, \mathbf{x}_i^U, i \in [1, n]$  — нижняя и верхняя границы области допустимых значений  $D$  по  $i$ -му измерению (см. рис. 1).

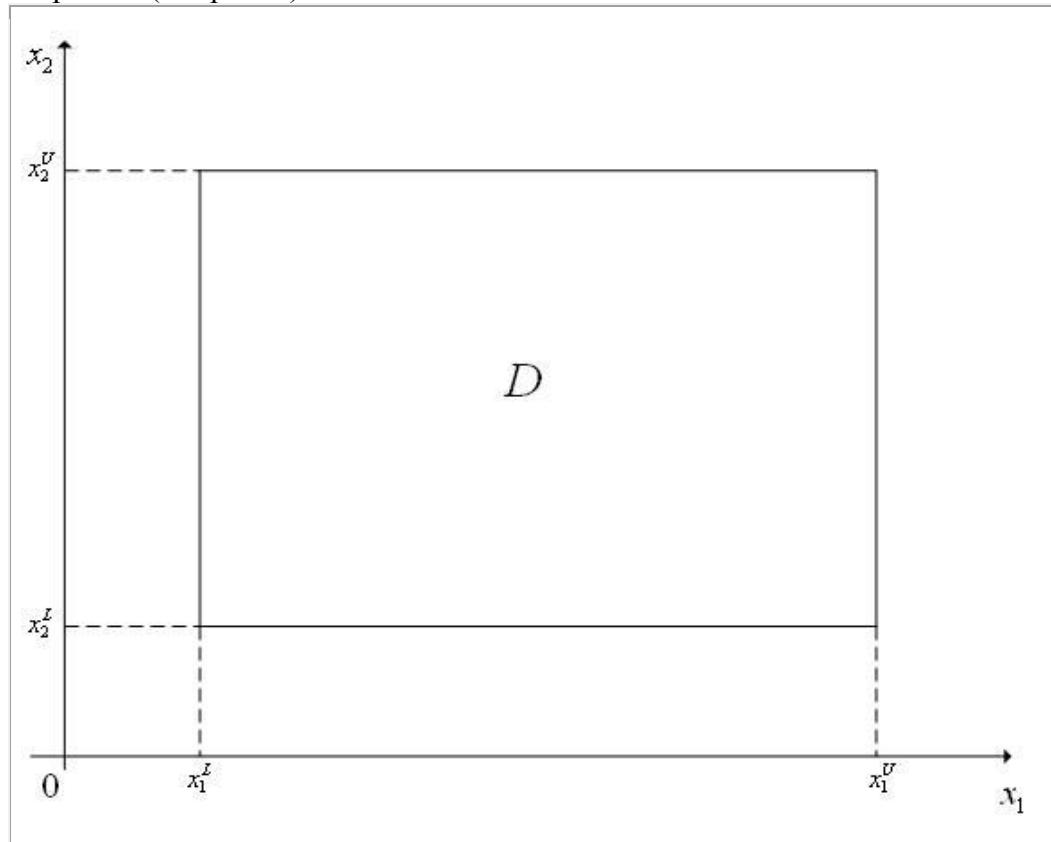


Рис. 1. Область допустимых значений  $D$  в виде гиперпараллелепипеда;  $n = 2$

Метод комплексов в многомерной задаче безусловной оптимизации рассмотрен в параграфе 8.2. В данном параграфе рассматривается модификация этого метода для решения многомерной задачи условной оптимизации - модифицированный метод комплексов.

Основные операции метода комплексов.

Напомним, что комплексом называется многогранник с  $N > n + 1$  вершинами (не обязательно выпуклый). Рекомендуется использовать комплекс с  $N = 2n$  вершинами. Так же, как при решении задачи безусловной оптимизации, при решении задачи (1) методом комплексов используются следующие операции:

генерация случайного комплекса;

отражение вершины комплекса с растяжением;

сжатие комплекса.

Генерация случайного комплекса. Координаты вершин случайного комплекса с  $N$  вершинами могут быть найдены по формуле

$$\mathbf{X}_i = \mathbf{X}_0 + l \frac{\boldsymbol{\Psi}_i}{\|\boldsymbol{\Psi}_i\|}, i \in [1, N], \quad (3)$$

где  $\mathbf{X}_0$  — произвольная начальная точка,  $i$  — номер вершины комплекса,  $l > 0$  — скаляр, определяющий размер комплекса,  $\Psi_i$  — реализация  $n$ -мерного случайного вектора,  $\|\cdot\|$  — некоторая векторная норма. Обычно в качестве координат вектора  $\Psi_i$  используют независимые случайные величины, равномерно распределенные в интервале  $[-1, 1]$ . Положим, что задан комплекс  $C^r$  с  $N$  вершинами  $\mathbf{X}_i^r, i \in [1, N]$ , и его вершину  $\mathbf{X}_k^r$  необходимо отразить через центр тяжести комплекса с растяжением. В новом комплексе  $C^{r+1}$  все вершины, кроме  $k$ -ой, совпадают с соответствующими вершинами исходного комплекса  $C^r$ , а  $k$ -я вершина находится на прямой, проходящей через центр тяжести этого комплекса и его вершину  $\mathbf{X}_k^r$  (см. рис. 2).

Обозначим координаты вершин нового комплекса  $\mathbf{X}_i^{r+1}, i \in [1, N]$ . Тогда имеем

$$\mathbf{X}_i^{r+1} = \mathbf{X}_i^r, i \in [1, N], i \neq k; \mathbf{X}_k^{r+1} = \mathbf{X}_c^r + \alpha(\mathbf{X}_c^r - \mathbf{X}_k^r), \quad (4)$$

где  $\alpha$  — коэффициент растяжения (рекомендуемое значение — 1.3),  $\mathbf{X}_c^r$  — вектор координат центра тяжести комплекса  $C^r$ :

$$\mathbf{X}_c^r = \frac{1}{N} \sum_{i=1}^m \mathbf{X}_i^r \quad (5)$$

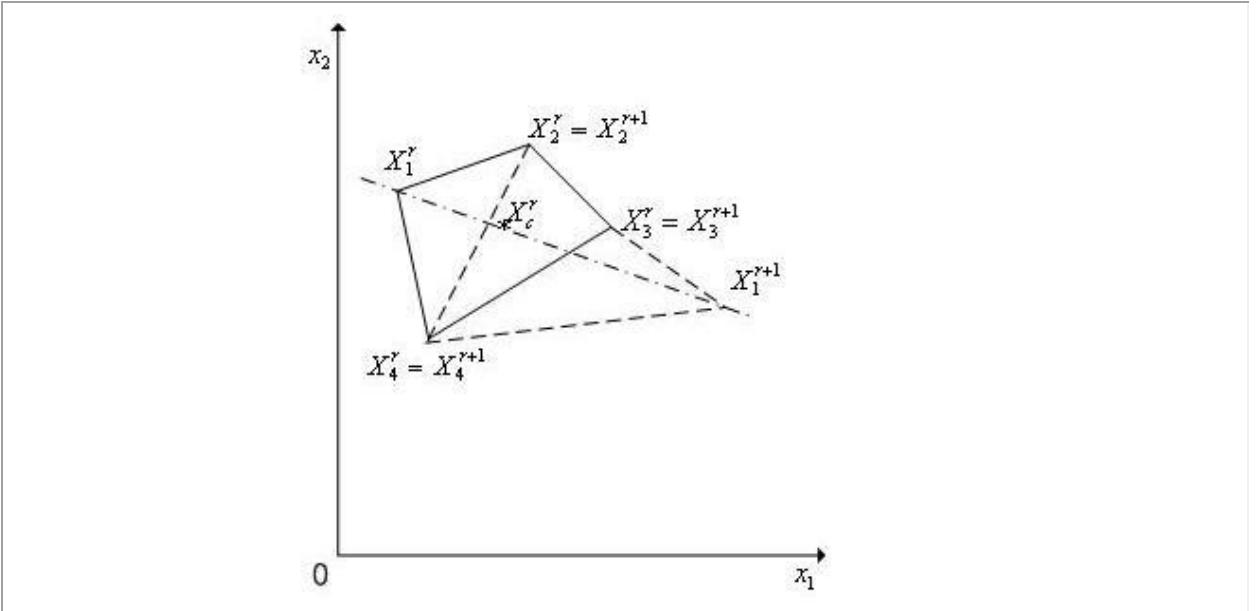


Рис. 2. Отражение вершины комплекса  $C$  через центр его тяжести с растяжением. Пунктиром показан новый комплекс  $C^{r+1}$ .

Сжатие комплекса. Положим, что задан комплекс  $C^r$  с  $N$  вершинами  $\mathbf{X}_i^r, i \in [1, N]$ , и его  $\mathbf{X}_k^r$  вершину необходимо переместить ближе к центру тяжести комплекса  $C^r$  — выполнить сжатие комплекса. В новом комплексе  $C^{r+1}$  все вершины, кроме  $k$ -ой, совпадают с соответствующими вершинами исходного комплекса  $C^r$ , а  $k$ -я вершина находится на прямой, проходящей через центр тяжести этого комплекса и его вершину  $\mathbf{X}_k^r$  (см. рис. 3). Обозначим координаты вершин нового комплекса  $\mathbf{X}_i^{r+1}, i \in [1, N]$ . Тогда имеем

$$\mathbf{X}_i^{r+1} = \mathbf{X}_i^r, i \in [1, N], i \neq k; \mathbf{X}_k^{r+1} = \mathbf{X}_c^r + \beta(\mathbf{X}_k^r - \mathbf{X}_c^r), \quad (6)$$

где —  $\beta$  коэффициент растяжения (рекомендуемое значение — 2),  $\mathbf{X}_c^r$  — вектор координат центра тяжести комплекса  $C^r$  (см. (5)).

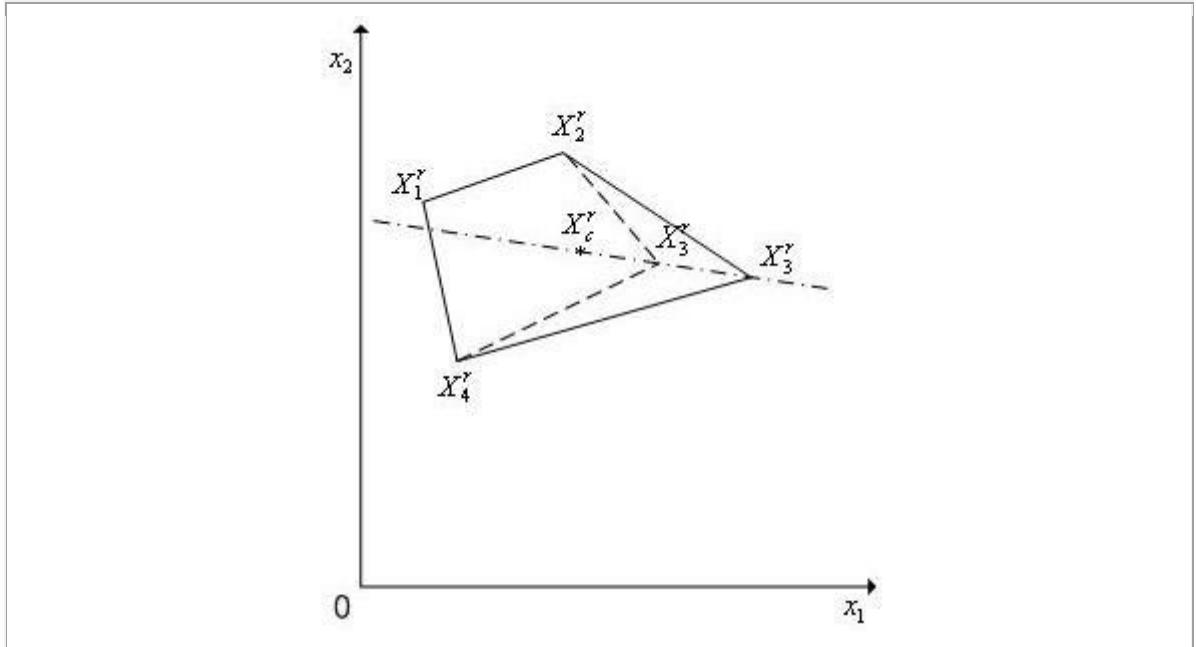


Рис. 3. Сжатие комплекса  $C^r$ . Пунктиром показан новый комплекс  $C^{r+1}$ . Упрощенная схема модифицированного метода комплексов.

Задаем начальную точку  $\mathbf{X}_0^0 \in D$ , исходя из которой должен быть построен комплекс  $C^0$ , величину  $\beta^0$ , а также коэффициенты  $\alpha, \beta$ ; полагаем счетчик числа итераций  $r = 0$ . Строим начальный комплекс  $C^0$ :

поочередно для  $i = 1, 2, \dots, N$  по формуле (3) находим координаты вершин комплекса  $\mathbf{X}_i^r$ ; комплекса  $C^r$ ;

если вершина  $\mathbf{X}_p^r$  является недопустимой (выходит за границы области  $D$ ), то по формуле, аналогичной формуле (6), выполняем сжатие уже построенного комплекса с  $P$  вершинами, вдоль направления  $(\mathbf{X}_p^r, (\mathbf{X}_c^r)_p)$ , где  $(\mathbf{X}_c^r)_p$  — центр тяжести уже найденных  $(p - 1)$ -ой вершин комплекса (см. рис. 4);

если после сжатия комплекса вершина  $\mathbf{X}_p^r$  по-прежнему является недопустимой, повторяем описанную процедуру сжатия;

вычисляем значения  $\Phi(\mathbf{X}_i^r)$  функции  $\Phi(\mathbf{X})$  во всех вершинах построенного комплекса  $C^r$ .

Находим максимальное из значений функции  $\Phi(\mathbf{X})_v$  вершинах комплекса  $C^r$   
 $\Phi(\mathbf{X}_k^r) = \max_{i \in [1, N]} \Phi(\mathbf{X}_i^r)$

По формулам (4), (5) отражаем с растяжением вершину  $\mathbf{X}_k^r$  комплекса  $C^r$  — получаем вершину и новый  $\mathbf{X}_k^{r+1}$  комплекс  $C^{r+1}$ :

если точка  $\mathbf{X}_k^{r+1}$  является не допустимой (выходит за границы области  $D$ ) и  $\Phi(\mathbf{X}_i^{r+1}) > \Phi(\mathbf{X}_i^r)$ , то по формуле (6), выполняем сжатие комплекса  $C^{r+1}$  вдоль направления  $(\mathbf{X}_k^{r+1}, \mathbf{X}_c^{r+1})$ , где —  $\mathbf{X}_c^{r+1}$  центр тяжести комплекса  $C^{r+1}$ , до тех пор, пока точка  $\mathbf{X}_k^{r+1}$  не станет допустимой (см. рис. 5). Переходим к п.5;

если точка  $\mathbf{X}_k^{r+1}$  является допустимой (не выходит за границы области  $D$ ) и  $\Phi(\mathbf{X}_i^{r+1}) < \Phi(\mathbf{X}_i^r)$ , то переходим к шагу 5;

если точка  $\mathbf{X}_k^{r+1}$  является не допустимой, но  $\Phi(\mathbf{X}_i^{r+1}) < \Phi(\mathbf{X}_i^r)$ , то переходим к п. 6. Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то в качестве точки  $\mathbf{X}^*$  принимаем вершину комплекса  $C^{r+1}$ , к которой функция  $\Phi(\mathbf{X})$  имеет наименьшее значение, вычисляем соответствующие значения  $\Phi(\mathbf{X})$  и завершаем итерации. Иначе — переходим к п. 3

Если  $(\mathbf{X}_k^{r+1})_i < \mathbf{X}_i^L$ , то полагаем  $(\mathbf{X}_k^{r+1})_i = \mathbf{X}_i^L$ ; если  $(\mathbf{X}_k^{r+1})_i > \mathbf{X}_i^U$  то полагаем  $(\mathbf{X}_k^{r+1})_i = \mathbf{X}_i^U$  (см. рис. 6). Переходим к п.3 •

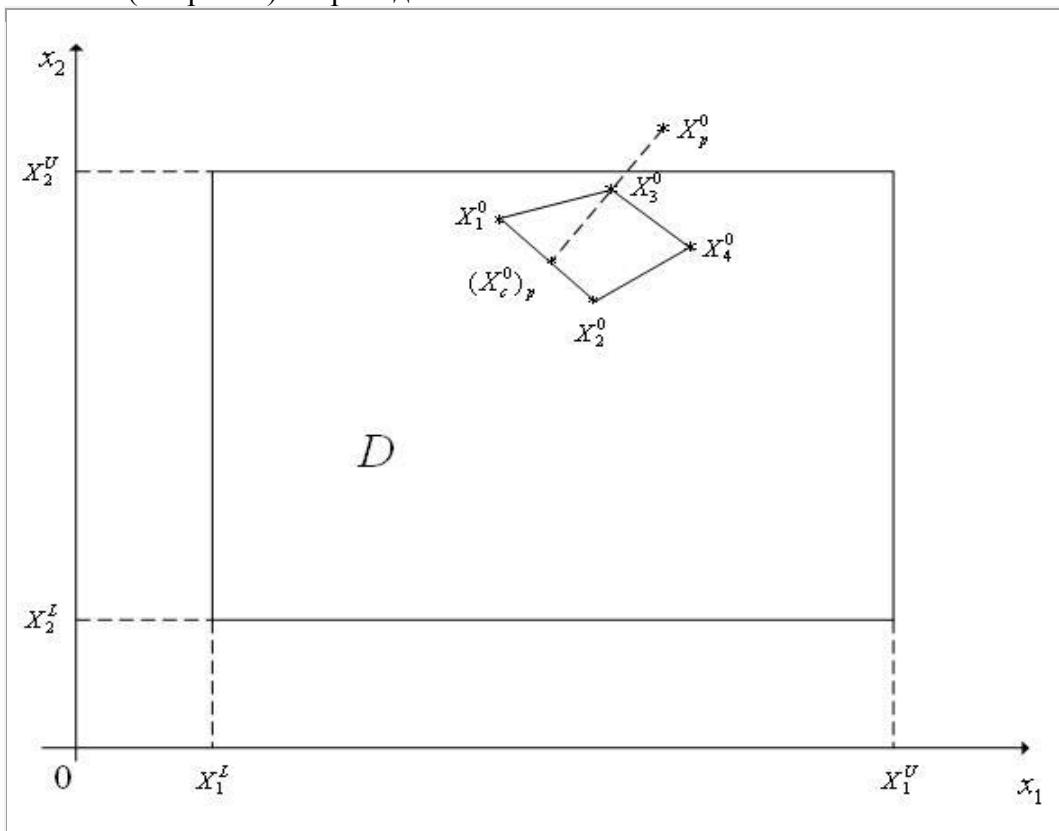


Рис. 4. Построение комплекса С0.

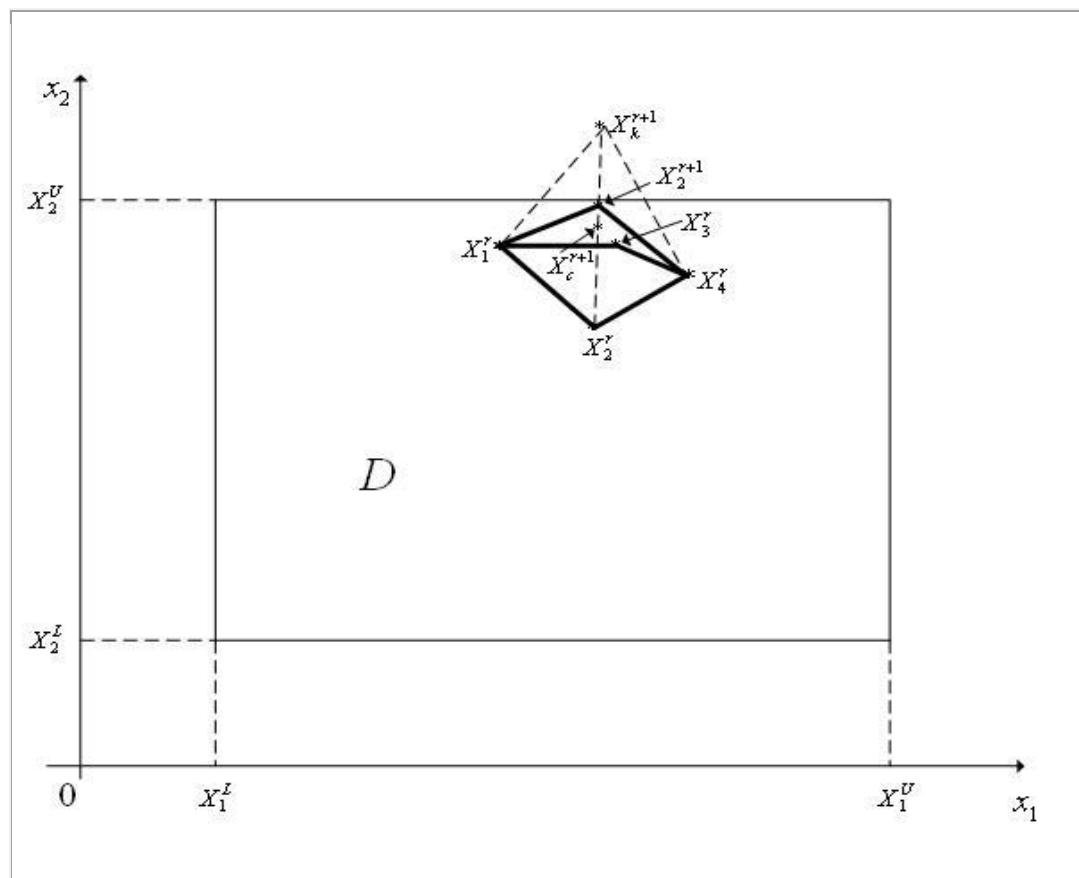


Рис. 5. Построение комплекса Cr+1.

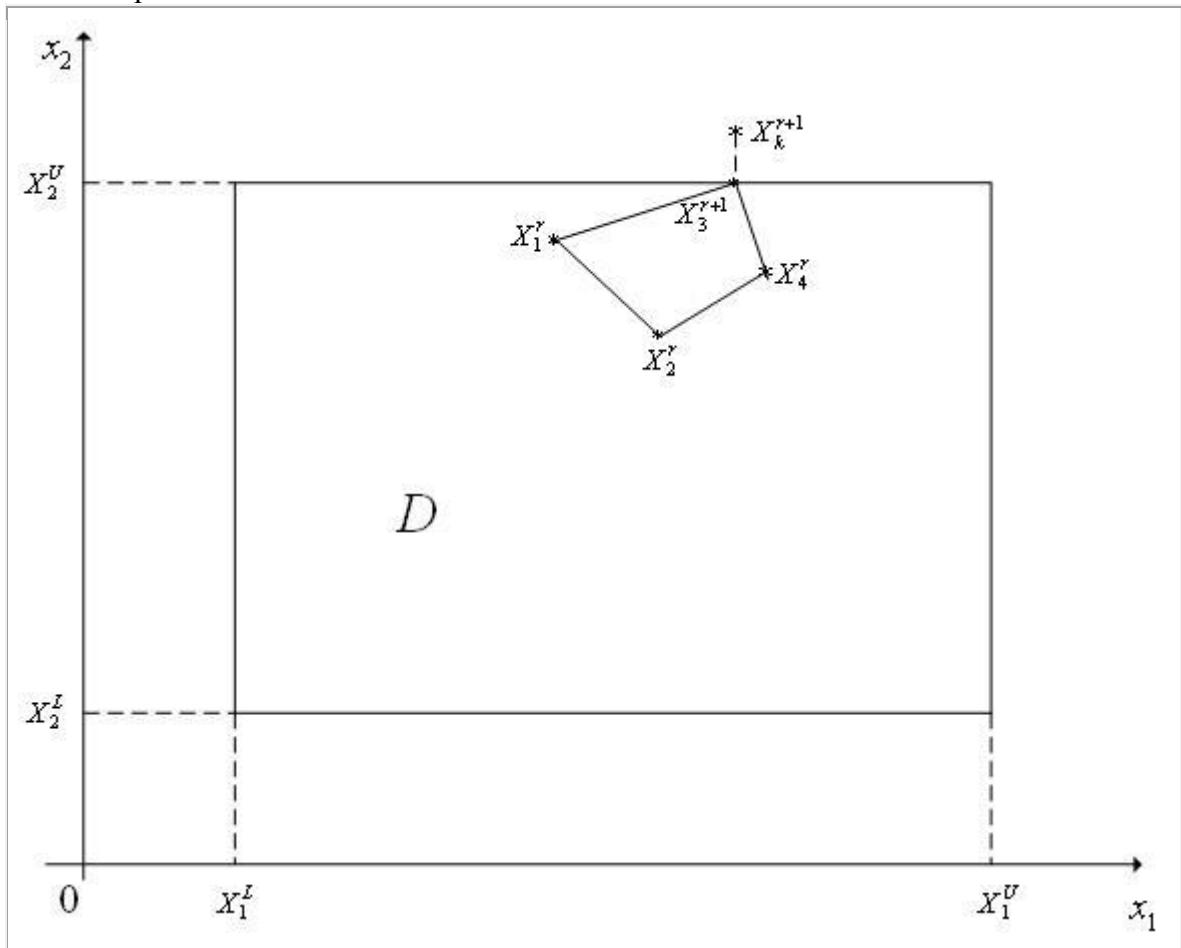


Рис. 6. Построение комплекса Cr+1.

На рис. 4 точка  $\mathbf{X}_o^p$  оказалась за границей области D. После операции сжатия комплекса вершины  $\mathbf{X}_o^p$  комплекса вдоль направления  $(\mathbf{X}_o^p, (\mathbf{X}_c^0)_p)$  получаем вершину  $\mathbf{X}_3^0$ . Здесь  $(\mathbf{X}_c^0)_p$  — центр тяжести комплекса.

На рис. 5 полагается, что  $\Phi(\mathbf{X}_k^{r+1}) > \Phi(\mathbf{X}_2^r)$ . Точка  $\mathbf{X}_k^{r+1}$  оказалась границей области D.

После операции сжатия комплекса вершины  $\mathbf{X}_k^{r+1}$  комплекса вдоль направления  $(\mathbf{X}_k^{r+1}, \mathbf{X}_c^{r+1})$  получаем вершину  $\mathbf{X}_2^{r+1}$ .

На рис. 6 полагается, что  $\Phi(\mathbf{X}_k^{r+1}) < \Phi(\mathbf{X}_2^r)$ . Точка  $\mathbf{X}_k^{r+1}$  оказалась за границей области D - нарушено ограничение  $\mathbf{X}_k^{r+1} < \mathbf{X}_2^U$ . Точка  $\mathbf{X}_3^{r+1}$  получена проектированием точки  $\mathbf{X}_k^{r+1}$  на прямую  $\mathbf{X}_2 = \mathbf{X}_2^U$ .

В качестве критерия окончания поиска может использоваться следующее условие: максимальная длина ребра комплекса  $C^r$  не превышает  $\varepsilon_x$  — требуемую точность решения по  $\mathbf{X}$ . Может использоваться также следующее аналогичное условие: максимальная разность значений функции  $\Phi(\mathbf{X})$  в двух вершинах комплекса  $C^r$  не превышает  $\varepsilon_\Phi$  — требуемую точность решения по  $\Phi$ .

Могут использоваться также более сложные условия окончания поиска, учитывающие текущий размер комплекса или в некотором смысле среднее значение функции  $\Phi(\mathbf{X})$  в его вершинах (см. параграф 8.2).

Изложенная схема метода комплексов приводит к "уплощению" комплекса вблизи границы области допустимых значений D, что может значительно уменьшить эффективность метода. С этой целью преодоления этого недостатка через фиксированное количество итераций находятся максимальная и минимальная диагонали комплекса и, если их отношение превышает заданное, то по рассмотренной схеме производится построение нового комплекса.

#### 9.4. Метод линейной аппроксимации

Сделаем ряд дополнительных допущений. Пусть множество допустимых значений  $D$  определяется только ограничениями типа неравенств и ограничивающие функции  $g_j(\mathbf{X}), j \in [1, l]$  являются непрерывными, дифференцируемыми и выпуклыми:

$$\min_{\mathbf{X} \in D \subset \mathbb{R}^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

$$D = \{\mathbf{X} \mid g_j(\mathbf{X}) \geq 0, j \in [1, l]\} \quad (2)$$

Пусть функция  $\Phi(\mathbf{X})$  также непрерывна, дифференцируема и выпукла во множестве  $D$ . Суть метода линейной аппроксимации.

Метод линейной аппроксимации использует на каждой итерации линейную аппроксимацию целевой функции  $\Phi(\mathbf{X})$  и ограничивающих функций  $g_j(\mathbf{X}), j \in [1, l]$  в окрестности текущей точки  $\mathbf{X}^r$

$$\tilde{\Phi}^r(\mathbf{X}) = \Phi(\mathbf{X}^r) + (\nabla \Phi(\mathbf{X}^r), (\mathbf{X} - \mathbf{X}^r)), \quad (3)$$

$$\tilde{g}_j^r(\mathbf{X}) = g_j(\mathbf{X}^r) + (\nabla g_j(\mathbf{X}^r), (\mathbf{X} - \mathbf{X}^r)), j \in [1, l] \quad (4)$$

Вместо задачи (1) на каждой итерации решается вспомогательная задача линейного программирования

$$\min_{\mathbf{x} \in \tilde{D}^r} \tilde{\Phi}^r(\mathbf{x}) = \tilde{\Phi}^r(\mathbf{x}^{r+1}) \quad (5)$$

$$\text{где } \tilde{D}^r = \{\mathbf{X} \mid \tilde{g}_j^r \geq 0, j \in [1, l]\}.$$

В изложенном виде метод может привести к выходу точки  $\mathbf{X}^{r+1}$  за пределы допустимой области (см. прим. 1).

Пример 1

Рассмотрим следующую двумерную задачу условной оптимизации с тремя ограничениями типа неравенств (первое ограничение — нелинейное, второе и третье ограничения — линейные):

$$\min_{\mathbf{X} \in D \subset \mathbb{R}^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*,$$

где

$$\Phi(\mathbf{X}) = x_1^2 + (x_2 - 6)^2 - 12,$$

$$D = \{\mathbf{X} \mid g_j(\mathbf{X}) \geq 0, j \in [1, 3]\};$$

$$g_1(\mathbf{X}) = -x_1^2 - x_2^2 + 25, g_2(\mathbf{X}) = x_1, g_3(\mathbf{X}) = x_2$$

Положим, что текущая точка есть  $\mathbf{X}^r = (2, 4)$ . Линеаризуем целевую функцию  $\Phi(\mathbf{X})$  и ограничивающую функцию  $g_1(\mathbf{X})$  в окрестности этой точки. Поскольку  $\Phi(\mathbf{X}^r) = 2^2 + (4 - 6)^2 - 12 = -4$ ,  $\nabla \Phi(\mathbf{X}) = \begin{pmatrix} 2x_1 \\ 2(x_2 - 6) \end{pmatrix}$ ,  $\nabla \Phi(\mathbf{X}^r) = \begin{pmatrix} 4 \\ -4 \end{pmatrix}$

по формуле (3) имеем

$$\tilde{\Phi}^r(\mathbf{X}) = -4 + \left( \begin{pmatrix} -4 \\ -4 \end{pmatrix}, \begin{pmatrix} x_1 - 2 \\ x_2 - 4 \end{pmatrix} \right) = -4x_1 - 4x_2 + 4;$$

Аналогично для ограничивающих функций по формуле (4) имеем:

$$\tilde{g}_1^r(\mathbf{X}) = 5 + \left( \begin{pmatrix} -4 \\ -8 \end{pmatrix}, \begin{pmatrix} x_1 - 2 \\ x_2 - 4 \end{pmatrix} \right) = -4x_1 - 8x_2 + 45;$$

$$\tilde{g}_2^r(\mathbf{X}) = g_2(\mathbf{X}) = x_1;$$

$$\tilde{g}_3^r(\mathbf{X}) = g_3(\mathbf{X}) = x_2;$$

Пример иллюстрирует рис. 1, на котором линии уровня целевой функции  $\Phi(\mathbf{X}) = x_1^2 + (x_2 - 6)^2 - 12$  получены с помощью следующей MATLAB-программы:

```
x = -2 : 0.1 : 6;
```

```
y = x;
```

```
[X, Y] = meshgrid(x);
```

```
Z = X.^2 + (Y - 6).^2 - 12;
```

```
V = [-10, -5, 0, 5, 10, 20, 40, 80];
```

```
[C, h] = contour(X, Y, Z, V);
```

```
clabel(C, h);
```

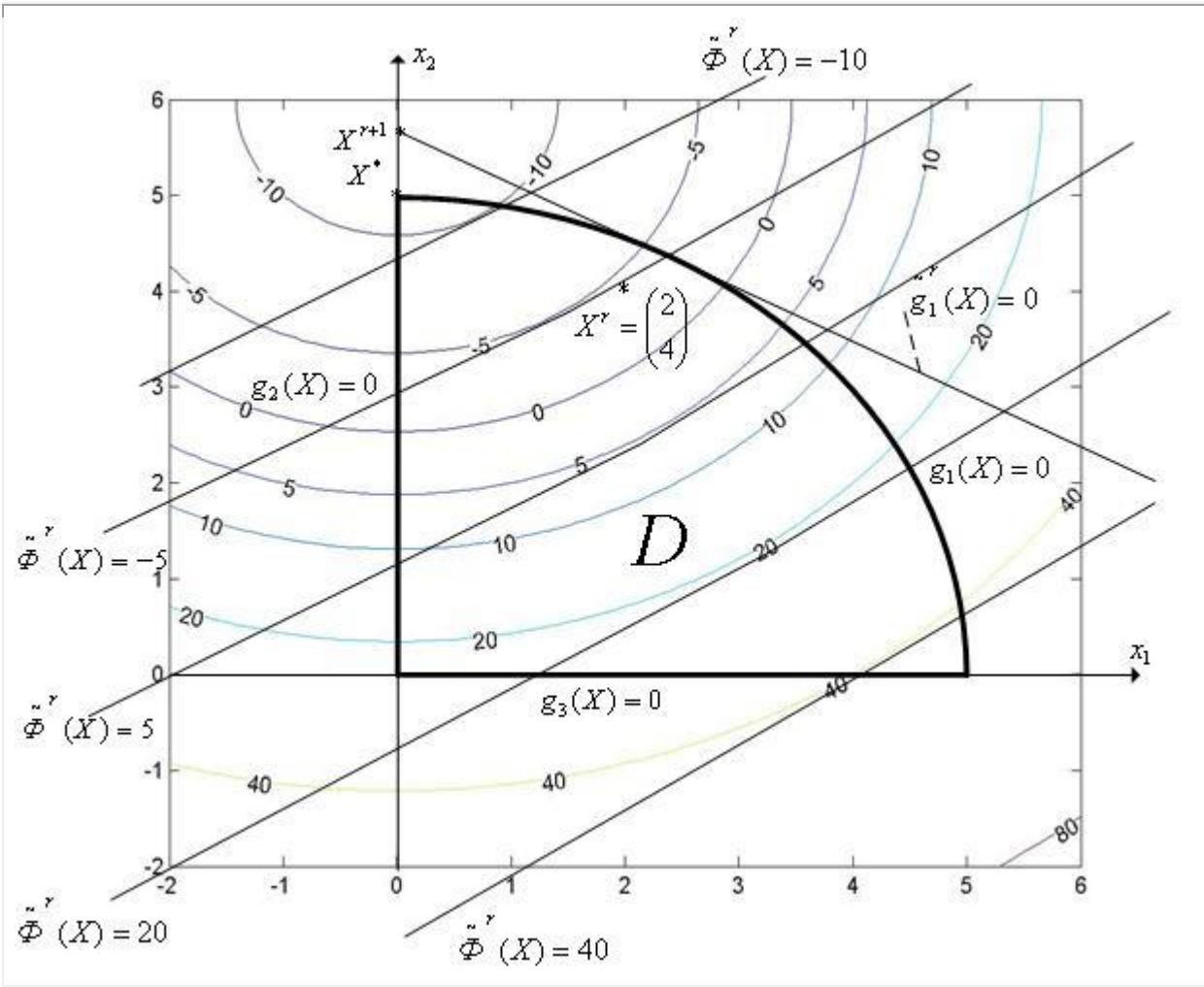


Рис. 1. Точка \$X^{r+1}\$ лежит вне области допустимых значений \$D\$.

Примечание 1

Прямая  $\tilde{g}_1^r$  представляет собой след от пересечения плоскости, которая является касательной к поверхности  $z = g_1(\mathbf{X})$  в точке  $\mathbf{X}^r$ , с плоскостью  $0x1x2$ . Эта прямая не обязательно является касательной к линии  $g_1(\mathbf{X}) = 0$  — прямая  $\tilde{g}_1^r(\mathbf{X}) = 0$  может пересекать кривую  $g_1(\mathbf{X}) = 0$ , быть касательной к ней или не иметь с ней общих точек. Аналогично, линия уровня  $\Phi^r(\mathbf{X}) = c$  функции  $\Phi(\mathbf{X})$  представляет собой след от пересечения плоскости, которая является касательной к поверхности  $z = \Phi(\mathbf{X})$  в точке  $\mathbf{X}^r$ , с плоскостью  $z = c$ .

Чтобы избежать выхода текущей точки за границы области допустимых значений, следующее приближение  $\mathbf{X}^{r+1}$  к точке минимума  $\mathbf{X}^*$  функции  $\Phi(\mathbf{X})$  из множества  $D$  находится по формуле

$$\mathbf{X}^{r+1} = \mathbf{X}^r + \lambda^r (\tilde{\mathbf{X}}^r - \mathbf{X}^r), \quad (6)$$

где  $\tilde{\mathbf{X}}^r$  — решение вспомогательной задачи линейного программирования (5).

Величина шага  $\lambda^r \in [0, 1]$  в формуле (6) в разных вариантах метода линейной аппроксимации может определяться разными способами. Приведем два из множества возможных способов.

1-й способ выбора величины шага  $\lambda^r$ . Величина  $\lambda^r$  находится как решение задачи одномерной оптимизации функции  $\Phi(\mathbf{X}^r + \lambda(\tilde{\mathbf{X}}^r - \mathbf{X}^r))$  на отрезке  $[0, 1]$ :

$$\Phi(\mathbf{X}^{r+1}) = \Phi(\mathbf{X}^r + \lambda^r(\tilde{\mathbf{X}}^r - \mathbf{X}^r)) = \min_{\lambda^r \in [0, 1]} \Phi(\mathbf{X}^r + \lambda(\tilde{\mathbf{X}}^r - \mathbf{X}^r)). \quad (7)$$

2-й способ выбора величины шага  $\lambda^r$ . Полагаем  $\lambda^r = 1$  и по формуле (6) находим вектор  $\mathbf{X}^{r+1}$ . Вычисляем значение  $\Phi(\mathbf{X}^{r+1})$  целевой функции в полученной точке. Если условие

$$\Phi(\mathbf{X}^{r+1}) < \Phi(\mathbf{X}^r) \quad (8)$$

не выполнено, то уменьшаем величину шага  $\lambda^r$  (например, в два раза) и повторно проверяем выполнение условия (8). Дробление шага  $\lambda^r$  и вычисление  $\mathbf{X}^{r+1}$  производим до выполнения условия (8).

Схема метода линейной аппроксимации.

Рассмотрим вариант метода, в котором используется 1-й способ выбора величины шага  $\lambda^r$ .

Задаем начальную точку  $\mathbf{X}^0 \in D$  и полагаем счетчик числа итераций  $r = 0$ .

Вычисляем градиенты функций  $\Phi(\mathbf{X}), g_j(\mathbf{X}), j \in [1, l]$  в точке  $\mathbf{X}^r$ .

Решаем задачу линейного программирования (5) — находим точку  $\tilde{\mathbf{X}}^r$ .

Решаем одномерную задачу минимизации (7) — находим величину шага  $\lambda^r$  и вектор  $\mathbf{X}^{r+1}$ .

Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем  $\mathbf{X}^* \approx \mathbf{X}^{r+1}, \Phi^* \approx \Phi(\mathbf{X}^{r+1})$  и завершаем итерации. Иначе — полагаем  $r = r + 1$  и переходим к п.2 •

В качестве критерия окончания поиска можно использовать стандартные условия окончания итераций

$$\|\mathbf{X}^{r+1} - \mathbf{X}^r\| = \lambda^r \leq \varepsilon_X,$$

$$|\Phi(\mathbf{X}^{r+1}) - \Phi(\mathbf{X}^r)| \leq \varepsilon_\Phi$$

или условие

$$\|\nabla \Phi(\mathbf{X}^{r+1})\| \leq \varepsilon_V,$$

где  $\varepsilon_V$  — константа, определяющая требуемую точность решения по градиенту функции  $\Phi(\mathbf{X})$ .

Отметим следующие трудности, возникающие при использовании метода линейной аппроксимации:

Если функция  $\Phi(\mathbf{X})$  имеет высокую степень нелинейности, то на основе решения вспомогательной задачи минимизации (5) направление поиска может быть выбрано слишком неточно (см. рис. 2), что приводит к медленной сходимости метода.

Метод требует, чтобы точка  $\mathbf{X}^0$  принадлежала множеству допустимых значений  $D$ . Если это требование не выполнено, то прежде приходится использовать какой-либо метод поиска точки, принадлежащей множеству допустимых значений.

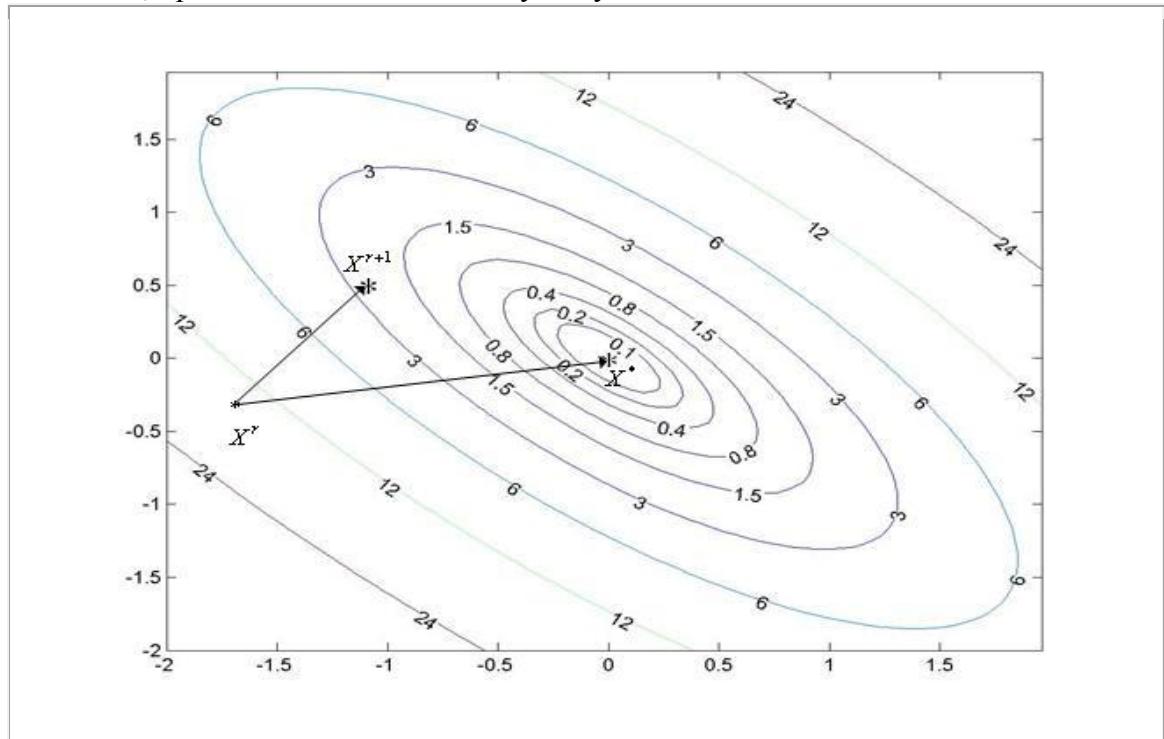


Рис. 2. Направление поиска ( $\mathbf{X}^{r+1}-\mathbf{X}^r$ ), которое обеспечивает метод на основе линейной аппроксимации, далеко от оптимального направления ( $\mathbf{X}^*-\mathbf{X}^r$ ).

Возможны модификации метода линейной аппроксимации, при которых необходимые производные вычисляются с помощью конечных разностей.

## 9.5. Метод проекции градиента

Рассмотрим многомерную задачу локальной условной оптимизации

$$\min_{\mathbf{X} \in D \subset \mathbb{R}^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где множество допустимых значений определяется только ограничениями типа неравенств

$$D = \{\mathbf{X} \mid g_j(\mathbf{X}) \geq 0, j \in [1, l]\} \quad (2)$$

и целевая функция  $\Phi(\mathbf{X})$  и ограничивающие функции  $g_j(\mathbf{X}), j \in [1, l]$  являются непрерывными и дифференцируемыми функциями, а ограничивающие функции еще и выпуклы.

Проектирование точки на множество.

Идея метода проекции градиента состоит в том, что если на некоторой итерации точка

$$\mathbf{X}^{r+1} = \mathbf{X}^r + \lambda^r \mathbf{S}^r, \quad (3)$$

полученная с помощью градиентного метода наискорейшего спуска (см. главу 7), оказывается вне множества допустимых значений  $D$ , то она возвращается на это множество. Возврат производится с помощью процедуры "проекция точки на множество".

Напомним, что в формуле (3)  $\lambda^r$  — длина шага на  $r$ -ой итерации в направлении  $\mathbf{S}^r$ ;

$$\mathbf{S}^r = -\frac{\nabla \Phi^r}{\|\nabla \Phi^r\|} -$$

единичный вектор направления антиградиента функции  $\Phi(\mathbf{X})$  в точке  $\mathbf{X}^r$ ,  $\|\cdot\|$  — некоторая векторная норма, например, евклидова.

Определение. Проекцией точки  $\mathbf{X} \in \mathbb{R}^n$  на замкнутое множество  $D \subset \mathbb{R}^n$  называется

ближайшая к  $\mathbf{X}$  точка  $\hat{\mathbf{X}}$  множества  $D$ . Т.е. точка  $\hat{\mathbf{X}} \in D$  называется проекцией точки  $\mathbf{X} \in \mathbb{R}^n$  на замкнутое множество  $D \subset \mathbb{R}^n$ , если

$$p(\mathbf{X}, \hat{\mathbf{X}}) = \min_{\mathbf{Y} \in D} p(\mathbf{X}, \mathbf{Y}) \quad (4)$$

где  $p(\mathbf{A}, \mathbf{B})$  — расстояние между точками  $\mathbf{A} \in \mathbb{R}^n, \mathbf{B} \in \mathbb{R}^n$  в некоторой метрике, например,  $p(\mathbf{A}, \mathbf{B}) = \|\mathbf{A} - \mathbf{B}\|_*$ .

Проекцию  $\hat{\mathbf{X}}$  точки  $\mathbf{X}$  на замкнутое множество  $D$  будем обозначать  $P_D(\mathbf{X}) = \hat{\mathbf{X}}$  (см. рис. 1). Очевидно, что  $P_D(\mathbf{X}) = \mathbf{X}$ , если  $\mathbf{X} \in D$ .

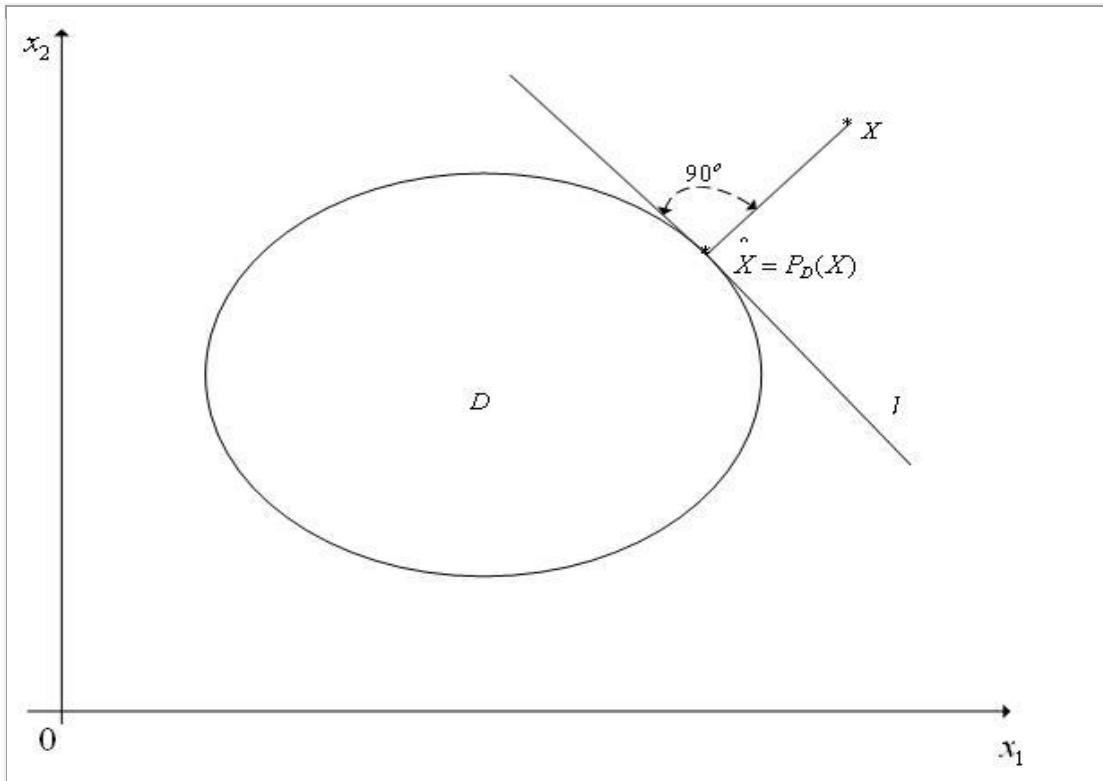


Рис. 1. К определению проекции точки на множество. Прямая  $l$  является касательной к границе области  $D$  в точке  $P_D(X)=X$ .

Можно показать, что если  $D$  — замкнутое выпуклое множество пространства  $\mathbb{R}^n$ , то для любой точки  $\mathbf{X} \in \mathbb{R}^n$  существует единственная ее проекция на это множество.

Задача (4) поиска проекции точки на множество также является многомерной задачей условной оптимизации и ее решении может вызвать в общем случае значительные затруднения.

Задача (4) становится задачей квадратичного программирования, если множество  $D$  задается лишь линейными ограничениями типа неравенств и если функция  $\rho(\mathbf{X}, \mathbf{Y})$  является квадратичной функцией  $\mathbf{Y}$ , например, если  $\rho(\mathbf{X}, \mathbf{Y}) = \|\mathbf{X} - \mathbf{Y}\|$ .

Наибольший практический интерес представляет ситуация, когда множество  $D$  таково, что задача (4) может быть решена в явном виде. Приведем несколько наиболее практически важных примеров таких множеств.

Схема комбинации метода проекции градиента с методом дробления шага.

Метод проекции градиента может быть скомбинирован со многими градиентными методами (см. главу 7). Рассмотрим комбинацию метода проекции градиента с градиентным методом дробления шага.

Напомним, что в градиентном методе с дроблением шага величина шага  $\lambda^r$  находится из условия

$$\Phi(\mathbf{X}^r) - \Phi(\mathbf{X}^{r+1}) < 0.5\lambda^r \|\nabla \Phi^r\| \quad (5)$$

Схема метода:

Задаем начальную точку  $\mathbf{X}^0$ , начальную величину шага  $\lambda^0$  и коэффициент дробления шага  $v \in (0, 1)$ . Полагаем счетчик числа итераций  $r = 0$ .

По формуле (3) вычисляем координаты точки  $\tilde{\mathbf{X}}^{r+1}$  и проекцию  $P_D(\tilde{\mathbf{X}}^{r+1}) = \mathbf{X}^{r+1}$  этой точки на множество  $D$ .

Вычисляем величину  $\Phi(\mathbf{X}^{r+1})$  — значение функции  $\Phi(\mathbf{X})$  в точке  $\mathbf{X}^{r+1}$ .

Если условие дробления шага выполнено (см. параграф 7.1), то переходим к следующему пункту. Иначе – переходим к п.6.

Полагаем  $\lambda^r = v\lambda^r$  и переходим к п.2.

Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем  $\mathbf{X}^* \approx \mathbf{X}^{r+1}$ ,  $\Phi^* \approx \Phi(\mathbf{X}^{r+1})$  и завершаем итерации. Иначе – полагаем  $r = r+1$  переходим к п.2 •

В качестве критерия окончания поиска можно использовать одно из стандартных условий окончания итераций

$$\|\mathbf{X}^{r+1} - \mathbf{X}^r\| = \lambda^r \leq \varepsilon_X,$$

$$|\Phi(\mathbf{X}^{r+1}) - \Phi(\mathbf{X}^r)| \leq \varepsilon_\Phi,$$

или условие  $\|\nabla \Phi^r\| \leq \varepsilon_V$ , где — константа, определяющая требуемую точность решения по градиенту функции  $\Phi(\mathbf{X})$ .

Комбинацию метода проекции градиента и градиентного метода с дроблением шага иллюстрирует рис. 2, на котором показан фрагмент линий уровня функции Химмельбау.

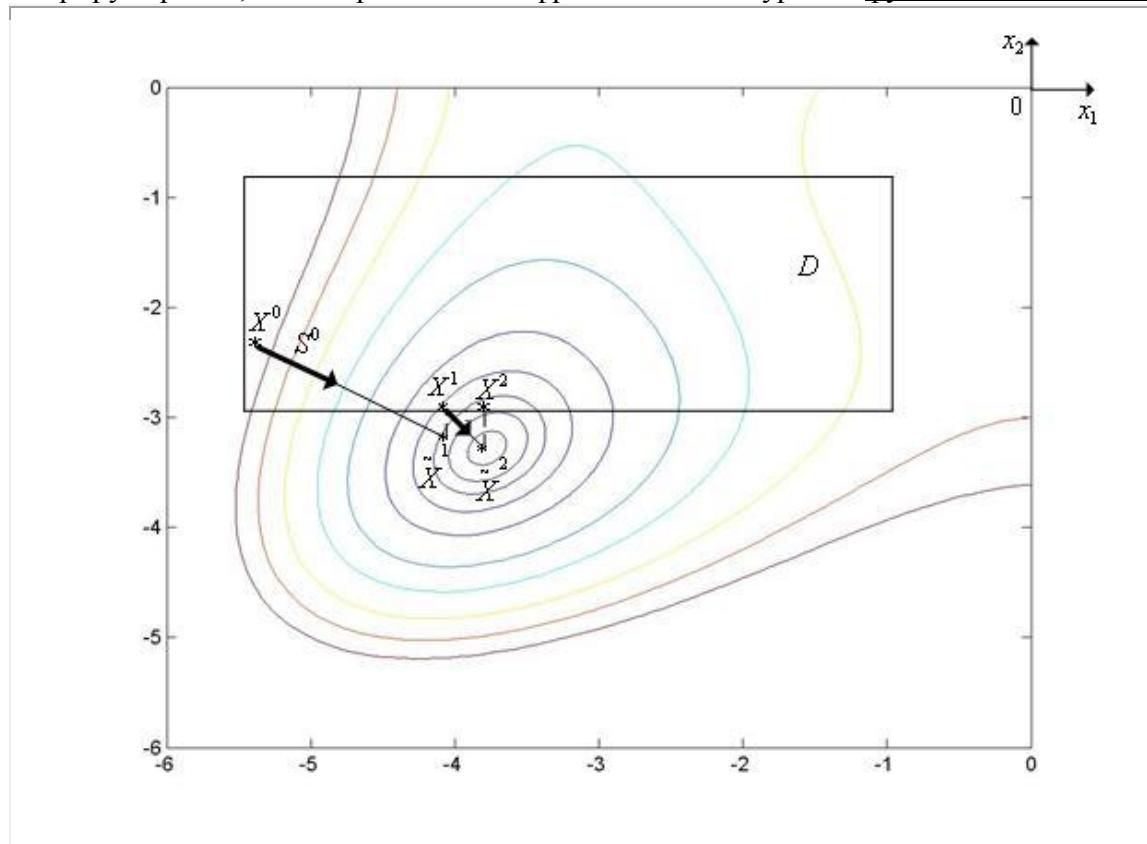


Рис. 2. Траектория поиска минимума функции Химмельбау комбинацией метода проекции градиента и градиентного метода с дроблением шага.

Известны модификации метода проекции градиента, ориентированные на решение задач условной оптимизации с ограничениями типа равенств.

## 10.1. Метод сведения к совокупности вложенных задач глобальной одномерной минимизации

Рассматривается следующая многомерная задача глобальной условной оптимизации: найти минимум критерия оптимальности  $\Phi(\mathbf{X})$ , определенного во множестве  $D$  пространства  $\mathbb{R}^n$ ,

$$\min_{\mathbf{X} \in D \subset \mathbb{R}^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где множество допустимых значений

$$D = \{\mathbf{X} \mid h_i(\mathbf{X}) = 0, g_j(\mathbf{X}) \geq 0, i \in [1, m], j \in [1, l]\}. \quad (2)$$

Положим, что множество допустимых значений  $D$  задается только с помощью ограничений типа неравенств и представляет собой гиперпараллелепипед

$$D = \{\mathbf{X} \mid x_i^L \leq x_i \leq x_i^U, i \in [1, n]\}. \quad (3)$$

Метод сведения к совокупности вложенных одномерных задач глобальной оптимизации состоит в решении вместо задачи (1), (3) следующей совокупности вложенных одномерных задач условной оптимизации.

$$\min_{\mathbf{X} \in D} \Phi(\mathbf{X}) = \min_{\mathbf{X} \in [x_1^L, x_1^U]} \min_{\mathbf{X} \in D(x_1)} \dots \min_{\mathbf{X} \in D(x_1, x_2, \dots, x_{n-1})} \Phi(x_1, x_2, \dots, x_n), \quad (4)$$

где множества  $D(x_i), D(x_1, x_2), \dots, D(x_1, x_2, \dots, x_{n-1})$ , представляют собой соответствующие сечения множества  $D$  (см. ниже).

Поясним смысл метода с помощью примера.

Пример 1

Положим, что  $\Phi(\mathbf{X}) = \Phi(x_1, x_2)$  и  $D = \{\mathbf{X} \mid x_1^L \leq x_1 \leq x_1^U, x_2^L \leq x_2 \leq x_2^U\}$ , т.е.  $n = 2$ . Вложенные одномерные задачи глобальной оптимизации (4) в этом случае можно представить в виде (см. рис. 1)

$$\min_{\mathbf{X} \in D} \Phi(\mathbf{X}) = \min_{x_1 \in [x_1^L, x_1^U]} \Phi(x_1), \quad (5)$$

$$\Phi(x_1) = \min_{x_2 \in D(x_1)} \Phi(x_1, x_2) \quad (6)$$

где  $D(x_1)$  — сечение области  $D$  прямой, параллельной оси  $0x$ . Задача (5) представляет собой одномерную задачу глобальной оптимизации критерия оптимальности  $\Phi(x_1)$  по параметру  $x_1 \in [x_1^L, x_1^U]$ , для вычисления значения которого при данном фиксированном  $x_1$  необходимо решить одномерную задачу глобальной оптимизации критерия оптимальности  $\Phi(x_1, x_2)$  по параметру  $x_2 \in D(x_1)$

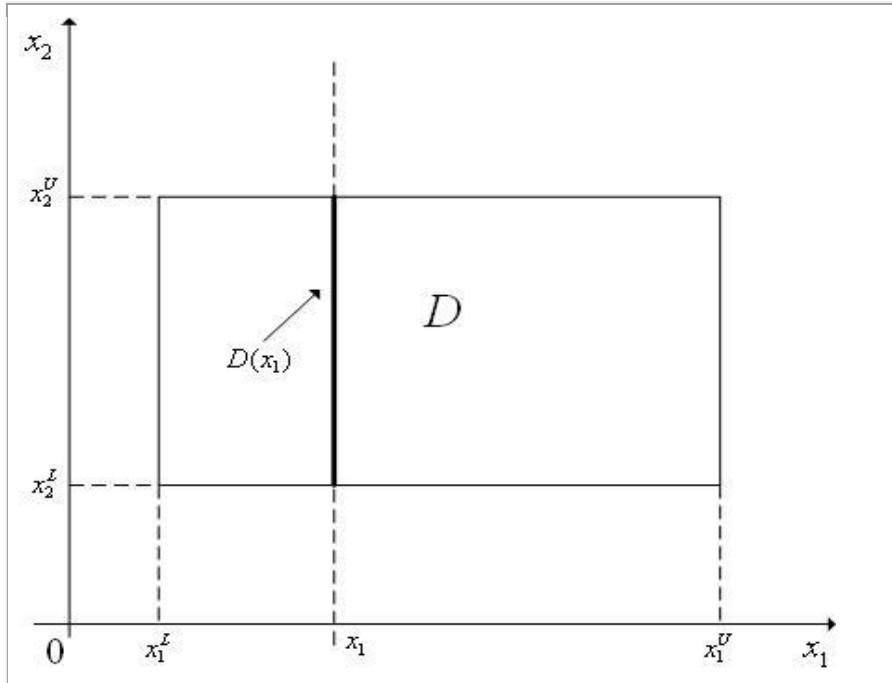


Рис. 1. К прим. 1. При решении задачи (5) вычисление значения критерия оптимальности  $\Phi(X)$  при некотором  $x = x_1$  требует решения задачи минимизации (6) на множестве  $D(x_1)$ . Положим, что для решения всех вложенных одномерных задач глобальной оптимизации (4) используется метод случайного поиска. Обозначим  $N_k$  число испытаний, необходимых для отыскания методом перебора с заданной точностью глобального минимума функции  $\Phi(x_1, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_n)$  по параметру  $x_k \in [x_k^L, x_k^U]$  (когда параметры  $x_1, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_n$  фиксированы). Тогда общее количество испытаний для решения задачи (1), (3) равно, очевидно,

$$N = \prod_{k=1}^n N_k.$$

Поэтому при  $n > 3$  такой алгоритм становится неэффективным. При  $n \leq 3$  надежность алгоритма достаточно высока, а затраты на поиск значительно меньше затрат на полный перебор на той же сетке.

Метод решения многомерной задачи глобальной условной оптимизации путем сведения к совокупности вложенных одномерных задач глобальной оптимизации может быть скомбинирован со всеми рассмотренными в главе 5 методами решения одномерных задач глобальной оптимизации. Рассмотрим комбинацию этого метода с методом случайного поиска для двумерной задачи (1), (3).

Схема комбинации метода с методом случайного поиска ( $n = 2$ ).

Задаем величины  $N_1, N_2$  – количества испытаний при решении задач (5), (6), соответственно. Полагаем  $r = 1$ .

Генерируем с помощью какого-либо программного генератора случайных чисел, равномерно распределенных в интервале  $[x_1^L, x_1^U]$ , случайное число  $x_1^0 = x_1^*$ .

Методом случайного поиска решаем задачу (6) при  $x_1 = x_1^*$  – находим точку  $x_2^0 = x_2^*$  и вычисляем значение критерия оптимальности  $\Phi^* = \Phi(x_1^*, x_2^*)$ .

Аналогично п.2 генерируем случайное число  $x_1^r$ .

Методом случайного поиска решаем задачу (6) при  $x_1 = \tilde{x}_1^r$  – находим точку  $\tilde{x}_1^r$  и вычисляем значение критерия оптимальности  $\Phi^r = \Phi(\tilde{x}_1^r, \tilde{x}_2^r)$ .

Если  $\Phi^r < \Phi^*$ , то выполняем присваивания  $x_1^* = \tilde{x}_1^r, x_2^* = \tilde{x}_2^r, \Phi^* = \Phi^r$ .

Если  $r_1 < N_1$ , то выполняем присваивание  $r_1 = r_1 + 1$  и переходим на п.4. Иначе принимаем точку  $(x_1^*, x_2^*)$  в качестве приближенного значения точки глобального минимума функции  $\Phi(\mathbf{X})$  в области  $D$  или каким-либо из рассмотренных ранее методов организуем в окрестности указанной точки поиск локального минимума функции  $\Phi(\mathbf{X})$  и заканчиваем вычисления ●

Отметим еще раз, что рассмотренный метод, как и любой другой метод глобальной оптимизации, при отсутствии априорной информации о свойствах минимизируемой функции не гарантирует нахождение глобального минимума.

Комбинацию рассматриваемого метода с методом случайного поиска для двумерной задачи иллюстрирует рис. 2, на котором показан фрагмент линий уровня функции Химмельблау ( $n = 2$ ). Принято, что  $X^*$  — точка минимума функции  $\Phi(X)$  в области  $D$  после ( $r = 1$ )-ой итерации. Точки на прямой  $x_1 = x_{r1}$  случайным образом сгенерированы на  $r$ -ой итерации. После завершения  $r$ -ой итерации, очевидно,  $X^* = (x_{r1}, x_{r1})$ .

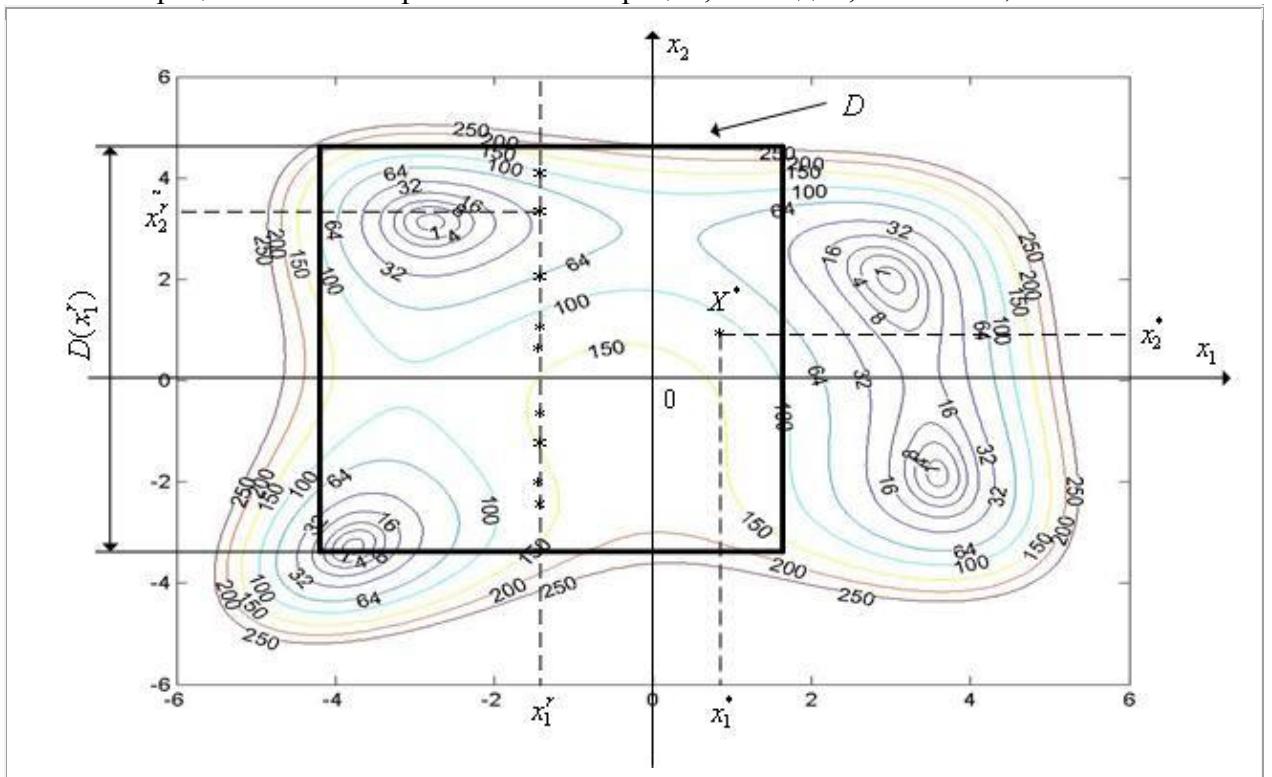


Рис. 2. Итерация номер  $r$  комбинации метода сведения с методом случайного поиска для двумерной задачи.

Линии уровня на рис. 1 получены с помощью следующей MATLAB-программы:  
 $x = -6 : 0.05 : 6;$   
 $y = x;$

```
[X, Y] = meshgrid(x);
Z = (X.^2 + Y - 11).^2 + (X + Y.^2 - 7).^2;
V = [1, 4, 8, 16, 32, 64, 100, 150, 200, 250];
[C, h] = contour(X, Y, Z, V);
clabel(C, h);
```

## 10.2. Метод сведения к задаче одномерной глобальной оптимизации с помощью развертки Пеано

Рассмотрим многомерную задачу глобальной условной оптимизации

$$\min_{\mathbf{X} \in D \subset \mathbb{R}^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где множество допустимых значений  $D$  задается только с помощью ограничений типа неравенств и представляет собой гиперкуб с длиной ребра, равной 1,

$$D = \{\mathbf{X} \mid -0.5 \leq x_i \leq 0.5, i \in [1, n]\}. \quad (2)$$

Отметим, что произвольный гиперпараллелепипед с помощью линейного преобразования может быть сведен к гиперкубу (2), так что рассмотрение в качестве множества  $D$  гиперкуба (2), а не гиперпараллелепипеда, не сужает общности рассуждений.

Рассматриваемый метод основан на использовании непрерывного отображения гиперкуба  $D$  на отрезок вещественной оси.

Разбиение гиперкуба. Развертка Пеано.

Шаг 1 ( $s = 1$ ). Координатными плоскостями гиперкуб  $D$  разбивается на  $2^n$  гиперкубов

$$\frac{1}{2} = \left(\frac{1}{2}\right)^{s=2}$$

первого разбиения с длиной ребра, равной  $\frac{1}{2}$  (см. рис. 1а). Пронумеруем их с помощью переменной  $z_1 \in [0, 2^n - 1]$  таким образом, чтобы гиперкубы с номерами, отличающимися на единицу, имели общую грань. Соединим центры гиперкубов ломаной  $\Pi_1$  в порядке введенной нумерации. Гиперкуб первого разбиения с номером  $z_1$  обозначим  $D(z_1)$ .

Шаг 2 ( $s = 2$ ). По рассмотренной схеме каждый гиперкуб первого разбиения разобьем плоскостями, параллельными координатным плоскостям и проходящими через его центр,

$$\frac{1}{4} = \left(\frac{1}{2}\right)^{s=2}$$

на  $2^n$  гиперкубов второго разбиения с длиной ребра, равной  $\frac{1}{4}$  (см. рис. 1б).

Пронумеруем полученные гиперкубы с помощью переменной  $z_2 \in [0, 2^n - 1]$  по тому же правилу, что и гиперкубы первого разбиения, с тем отличием, что нулевой гиперкуб второго разбиения, входящий в гиперкуб  $D(z_1)$ , должен иметь общую грань с  $(2^n - 1)$ -м гиперкубом второго разбиения, входящим в гиперкуб  $D(z_1 - 1)$ . Соединим центры гиперкубов ломаной  $\Pi_2$  в порядке введенной нумерации. Обозначим гиперкубы второго разбиения  $D(z_1, z_2)$ .

.....

Шаг  $s$ . Аналогично шагу 2 разбиваем гиперкубы  $(s - 1)$ -го разбиения на гиперкубы  $s$ -го

$$\left(\frac{1}{2}\right)^s$$

разбиения с длиной ребра, равной  $\left(\frac{1}{2}\right)^s$ , нумеруем их с помощью переменной  $z_s \in [0, 2^n - 1]$ , соединяя центры гиперкубов ломаной  $\Pi_s$  в порядке введенной нумерации и обозначаем  $D(z_1, z_2, \dots, z_s)$  •

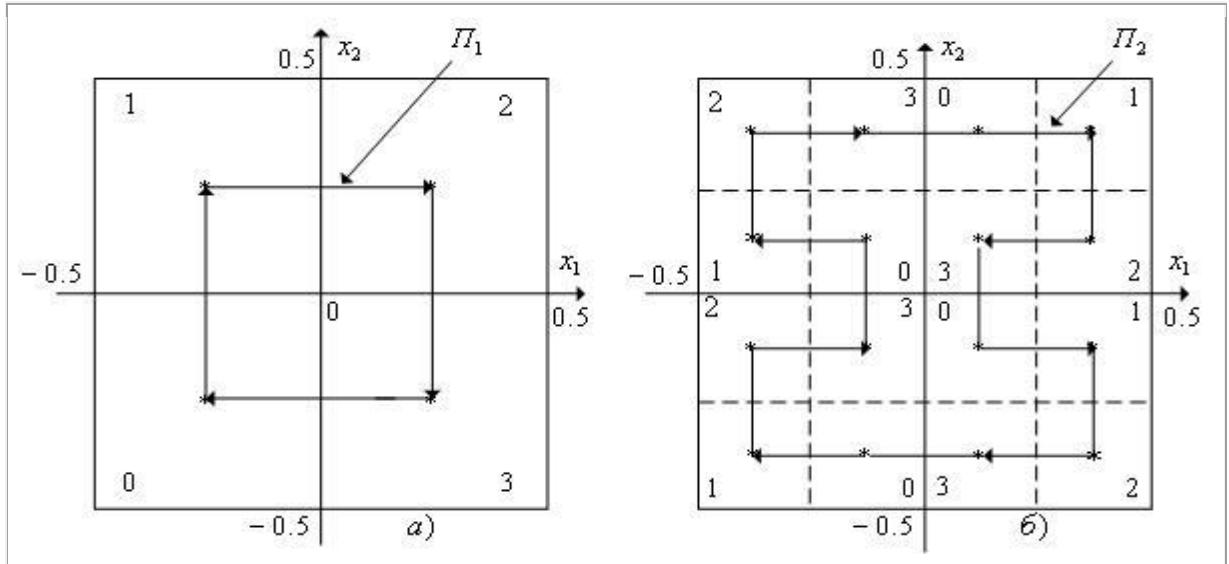


Рис. 1. К разбиению гиперкуба ( $n = 2$ ). а) Первое разбиение. б) Второе разбиение. Стрелками показано направление нумерации гиперкубов.

Ломаная  $\Pi_s$  называется разверткой Пеано. В пределе при  $s \rightarrow \infty$  ломаная  $\Pi_s$  называется кривой Пеано. Кривая Пеано обладает тем свойством, что проходит через все точки гиперкуба и имеет в каждой точке излом  
Разбиение отрезка  $[0, 1]$ .

Шаг 1 ( $s=1$ ). Разобьем отрезок  $[0, 1]$  на  $2^n$  равных частей длиной  $\left(\frac{1}{2^n}\right)^{s=2}$  (см. рис. 2а), пронумеруем их слева направо с помощью переменной  $z_1 \in [0, 2^n - 1]$  и обозначим  $\Delta(z_1)$ .

Шаг 2 ( $s=2$ ). Каждый из отрезков  $\Delta(z_1)$ ,  $z_1 \in [0, 2^n - 1]$  разобьем на  $2^n$  равных частей длиной  $\left(\frac{1}{2^n}\right)^{s=2}$  (см. рис. 2б), пронумеруем их слева направо с помощью переменной  $z_2 \in [0, 2^n - 1]$  и обозначим  $\Delta(z_1, z_2)$ . ....

Шаг  $s$ . Аналогично шагу 2 каждый из отрезков  $(s-1)$ -го разбиения разобьем на  $2^n$  равных частей длиной  $\left(\frac{1}{2^n}\right)^s$ , пронумеруем их слева направо с помощью переменной  $z_n \in [0, 2^n - 1]$  и обозначим  $\Delta(z_1, z_2, \dots, z_n)$  •

Определим отображение точки  $v$  отрезка  $[0, 1]$  на гиперкуб  $D$  следующим образом: если точка  $v \in \Delta(z_1, z_2, \dots, z_s)$ , то соответствующая точка  $X_D(z_1, z_2, \dots, z_s)$  является центром гиперкуба  $D(z_1, z_2, \dots, z_s)$ . Обозначим введенное отображение  $P(v)$ . Таким образом, если  $v \in \Delta(z_1, z_2, \dots, z_s)$ , то  $P(v) = X_D(z_1, z_2, \dots, z_s) \in D(z_1, z_2, \dots, z_s)$  (см. рис. 3).

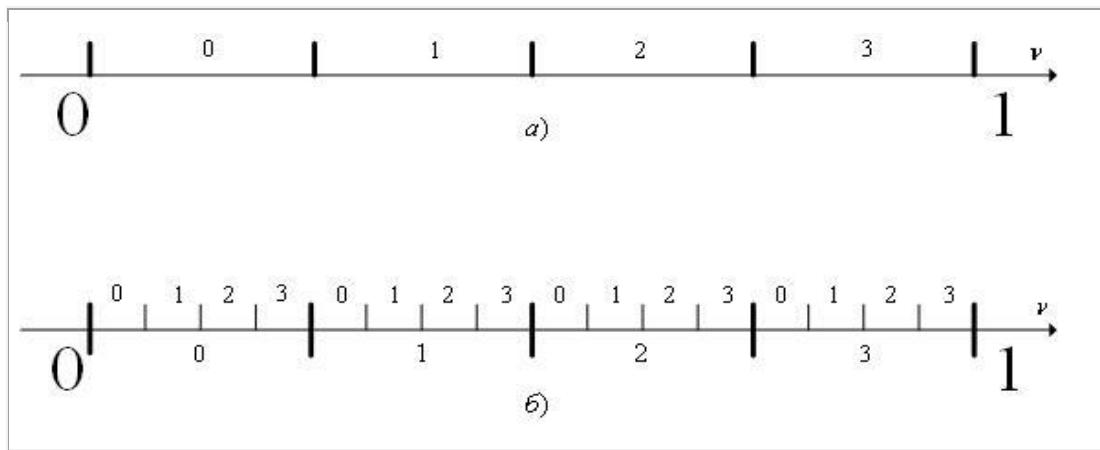


Рис. 2. К разбиению отрезка  $[0, 1]$ . а) Первое разбиение. б) Второе разбиение.

Отображение отрезка  $[0, 1]$  на гиперкуб.

На рис. 3 любая точка  $v_1 \in \Delta(0, 2)$  отображается в центр гиперкуба  $D(0, 2)$  - точку  $X_{D(0, 2)}$ . Аналогично, любая точка  $v_2 \in \Delta(1, 3)$  отображается в точку  $X_{D(1, 3)}$  и любая точка  $v_3 \in \Delta(2, 1)$  отображается в точку  $X_{D(2, 1)}$ .

В пределе при  $s \rightarrow \infty$  введенное отображение отображает отрезок  $[0, 1]$  на кривую Пеано. Можно показать, что в пределе при  $s \rightarrow \infty$  построенное отображение является непрерывным и взаимнооднозначным.

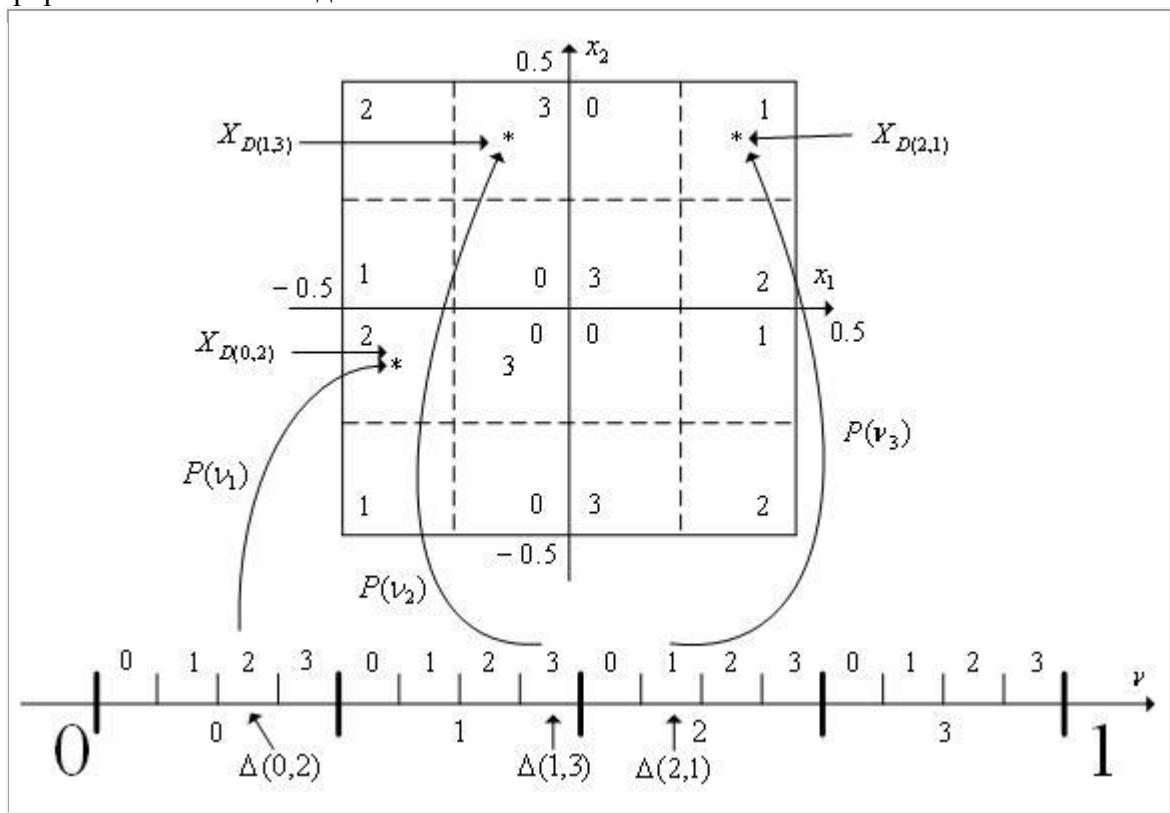


Рис. 3. К отображению отрезка  $[0, 1]$  на гиперкуб.

Пусть  $0, \alpha_1 \alpha_2 \dots \alpha_k \dots$  — двоичное представление числа  $v$ , т.е.  $v_b = 0, \alpha_1 \alpha_2 \dots \alpha_k \dots$

Утверждение 1 Если  $v \in \Delta(z_1, z_2, \dots, z_s)$ , то первые  $s$  двоичных цифр числа  $v_b = 0, \alpha_1 \alpha_2 \dots \alpha_k \dots$  определяют разбиение  $\Delta(z_1, z_2, \dots, z_s)$  отрезка  $[0, 1]$ :

$$z_1 = \langle \alpha_1 \alpha_2 \dots \alpha_n \rangle;$$

$$z_2 = \langle \alpha_{n+1} \alpha_{n+2} \dots \alpha_{2n} \rangle;$$

...

$$z_s = \langle \alpha_{(s-1)n+1} \alpha_{(s-1)n+2} \dots \alpha_{sn} \rangle.$$

Здесь  $\langle \cdot \rangle$  - операция преобразования двоичного числа в десятичное •

Пример 1

Пусть область  $D$  представляет собой квадрат ( $n = 2$ ). На отрезке  $[0,1]$  рассмотрим точки  $v_1 = 0.26$ ,  $v_2 = 0.49$ ,  $v_3 = 0.74$  – см. рис. 4.

Преобразуем  $v_1 = 0.26$  в двоичную систему счисления:

$0.26 * 2 = 0.52$  — запоминаем целую часть 0;

$0.52 * 2 = 1.04$  — запоминаем целую часть 1;

$0.04 * 2 = 0.08$  — запоминаем целую часть 0;

$0.08 * 2 = 0.16$  — запоминаем целую часть 0;

$0.16 * 2 = 0.32$  — запоминаем целую часть 0;

$0.32 * 2 = 0.64$  — запоминаем целую часть 0;

$0.64 * 2 = 1.28$  — запоминаем целую часть 1;

Итого,  $v_1 = 0.26 = 0.0100001\dots_b$ ,  $z_1 = \langle \alpha_1 \alpha_2 \rangle = \langle 01 \rangle = 1_d$ ,  $z_2 = \langle \alpha_3 \alpha_4 \rangle = \langle 00 \rangle = 0_d$ . Таким образом,  $v_1 \in \Delta(z_1, z_2) = \Delta(1, 0)$ .

Аналогично  $v_2 = 0.49 = 0.0111111\dots_b$ ,  $z_1 = \langle \alpha_1 \alpha_2 \rangle = \langle 01 \rangle = 1_d$ ,  $z_2 = \langle \alpha_3 \alpha_4 \rangle = \langle 11 \rangle = 3_d$ .

Таким образом,  $v_2 \in \Delta(z_1, z_2) = \Delta(1, 3)$ .

И  $v_3 = 0.74 = 0.1011101\dots_b$ ,  $z_1 = \langle \alpha_1 \alpha_2 \rangle = \langle 10 \rangle = 2_d$ ,  $z_2 = \langle \alpha_3 \alpha_4 \rangle = \langle 11 \rangle = 3_d$ . Таким образом,  $v_3 \in \Delta(z_1, z_2) = \Delta(2, 3)$

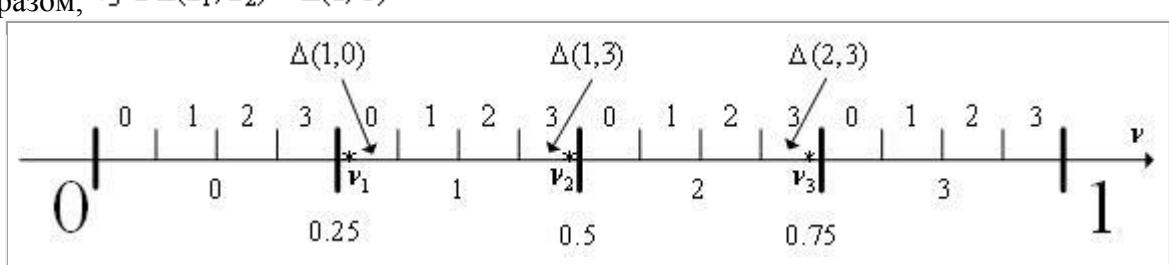


Рис. 4. К прим. 1.

Определим на отрезке  $[0, 1]$  функцию  $\Phi(v) = \Phi(\mathbf{X}) = P(v)$ . Отметим, что если функция  $\Phi(\mathbf{X})$  является непрерывной функцией, то функция  $\Phi(v)$  также непрерывна. Однако эта функция является негладкой и многоэкстремальной, даже если исходная функция  $\Phi(\mathbf{X})$  гладкая и унимодальная.

Таким образом, с помощью развертки Пеано многомерная задача глобальной условной оптимизации (1), (2) сводится к одномерной задаче условной глобальной оптимизации

$$\min_{v \in [0,1]} \Phi(v) = \Phi(v^*) = \Phi_*.$$

Решение задачи многомерной глобальной условной оптимизации с помощью развертки Пеано.

Метод решения многомерной задачи глобальной условной оптимизации с использованием развертки Пеано называется методом развертки Пеано и может быть скомбинирован со всеми рассмотренными в главе 5 методами решения одномерных задач глобальной оптимизации. При этом тот факт, что фактически решается задача не одномерной, а многомерной оптимизации, вносит следующие особенности в указанные методы:

Должна быть задана требуемая точность  $\varepsilon_x$  решения исходной задачи (1), (2) по  $\mathbf{X}$ . Исходя из этой точности, предварительно должно быть определено  $s$  — количество разбиений области  $D$  (см. ниже).

Вычисления значений критерия оптимальности  $\Phi(v)$ ,  $v \in [0, 1]$  должны производиться по следующей схеме:

для заданного  $v \in [0, 1]$  находим  $s$  цифр его двоичного представления

$$v_0 = 0, \alpha_1 \alpha_2 \dots \alpha_s \dots;$$

определяем числа  $z_1 = \langle \alpha_1, \alpha_2, \dots, \alpha_s \rangle$ ,  $z_2 = \langle \alpha_{n+1} \alpha_{n+2} \dots \alpha_{2n} \rangle$ ,  $\dots$ ,

$$z_s = \langle \alpha_{(s-1)n+1} \alpha_{(s-1)n+2} \dots \alpha_{sn} \rangle;$$

в гиперкубе  $D(z_1, z_2, \dots, z_s)$  выбираем его центр  $\mathbf{X}_{D(z_1, z_2, \dots, z_s)}$ ;

вычисляем значение критерия оптимальности  $\Phi(\mathbf{X})$  в этой точке, которое и принимаем за значение  $\Phi(v)$ .

При заданной точность  $\varepsilon_x$  решения задачи (1), (2) по  $\mathbf{X}$  необходимое количество  $s$  разбиений гиперкуба  $D$  может быть найдено из следующих соображений. Гиперкуб  $s$ -го

разбиения  $D(z_1, z_2, \dots, z_s)$  имеет длину ребра, равную  $\left(\frac{1}{2}\right)^s$ . Максимальное расстояние точек этого гиперкуба до его центра равно половине диагонали гиперкуба, которая,

очевидно, равна корню квадратному из суммы квадратов  $n$  ребер гиперкуба, т.е.  $n\left(\frac{1}{2}\right)^{2s}$ .

Таким образом,  $s$  может быть найдено из условия

$$n\left(\frac{1}{2}\right)^{2s} \leq \varepsilon_x$$

Отметим еще раз, что рассмотренный метод, как и любой другой метод глобальной оптимизации, при отсутствии априорной информации о свойствах минимизируемой функции не гарантирует нахождение глобального минимума.

### 10.3. Метод Монте-Карло

Рассмотрим многомерную задачу глобальной условной оптимизации

$$\min_{\mathbf{X} \in D \subset \mathbb{R}^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) = \Phi^*, \quad (1)$$

где множество допустимых значений

$$D = \{\mathbf{X} \mid h_i(\mathbf{X}) = 0, g_j(\mathbf{X}) \geq 0, i \in [1, m], j \in [1, l]\} \quad (2)$$

определяется как ограничениями типа неравенств, так и ограничениями типа равенств  $\Phi^*$   $(3)$

Метод Монте-Карло относится к классу прямых методов случайного поиска.

Схема метода Монте-Карло.

Задаем общее количество испытаний  $N$  и полагаем счетчик числа итераций  $r = 1$ .

С помощью какого-либо программного генератора случайных чисел генерируем  $n$  компонент вектора  $\mathbf{X}^1 \in D$ .

Вычисляем  $\Phi(\mathbf{X}^1)$  и полагаем  $\mathbf{X}^* = \mathbf{X}^1, \Phi^* = \Phi(\mathbf{X}^1), r = r + 1$ .

Аналогично п.2 генерируем случайную точку  $\mathbf{X}^r \in D$ . Вычисляем соответствующее значение критерия оптимальности  $\Phi(\mathbf{X}^r) = \Phi^*$ .

Выполняем следующие присваивания:

$$\mathbf{X}^* = \begin{cases} \mathbf{X}^r, & \text{если } \Phi^r < \Phi^*, \\ \mathbf{X}^*, & \text{иначе.} \end{cases}$$

Если  $r < N$  полагаем  $r = r + 1$  и переходим на п.4, иначе принимаем  $\mathbf{X}^*, \Phi^*$  в качестве приближенного решения задачи и заканчиваем вычисления •

Отметим, что в простейшем случае точки  $\mathbf{X}^r \in D$  генерируются равномерно распределенными в области  $D$ . С целью сокращения вычислительных затрат и при наличии априорной информации о положении точки глобального минимума, целесообразно использовать законы распределения, в которых вероятность генерации точки в окрестности предполагаемого глобального минимума выше, чем вне этой окрестности.

Для локализации с помощью метода Монте-Карло глобального минимума с высокой вероятностью и точностью, требуется очень большое количество испытаний  $N$ . Поэтому метод Монте-Карло обычно комбинируют с каким-либо детерминированным методом локальной оптимизации.

Комбинация метода Монте-Карло с детерминированным методом локальной оптимизации.

Задаем общее количество исходных случайных точек  $N$ .

С помощью какого-либо программного генератора случайных чисел генерируем  $N$  случайных точек  $\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^N$ , принадлежащие множеству  $D$ .

Полагаем  $r = 1$ .

Исходя из точки  $\mathbf{X}^r$ , каким-либо многомерным методом условной оптимизации (см. главу 9) находим локальный минимум  $(\mathbf{X}^*)^r$  функции  $\Phi(\mathbf{X})$  в окрестности этой точки и вычисляем  $\Phi((\mathbf{X}^*)^r) = (\Phi^*)^r$ .

Если  $r < N$  полагаем  $r = r + 1$  и переходим на п.4, иначе – переходим к следующему пункту.

Находим минимальное из чисел  $(\Phi^*)^r, r \in [1, N]$ . Пусть

$$\min_{r \in [1, N]} (\Phi^*)^r = (\Phi^*)^k$$

Принимаем в качестве приближенного решения задачи  $(X^*)^k, (\Phi^*)^r$  и заканчиваем вычисления •

### 11.1. Постановка задачи многокритериальной оптимизации. Множество Парето

В задачах САПР часто возникает задача обеспечить оптимальность объекта проектирования одновременно по нескольким критериям оптимальности  $\phi_k(\mathbf{X}), k \in [1, s]$ . Обычно эти критерии противоречивы и оптимизация по каждому из них приводит к различным значениям вектора варьируемых параметров  $\mathbf{X}^*$ . Поэтому выделяется отдельный класс задач многокритериальной оптимизации. Постановка задачи многокритериальной оптимизации.

Будем называть каждый из скалярных критерииев оптимальности  $\phi_k(\mathbf{X}), k \in [1, s]$  частным критерием оптимальности. Совокупность частных критериев оптимальности  $\Phi(\mathbf{X}) = (\phi_1(\mathbf{X}), \phi_2(\mathbf{X}), \dots, \phi_s(\mathbf{X}))$  будем называть векторным критерием оптимальности. Положим, что ставится задача минимизации каждого из частных критериев оптимальности  $\phi_1(\mathbf{X}), \phi_2(\mathbf{X}), \dots, \phi_s(\mathbf{X})$  в одной и той же области допустимых значений  $D_x \in R^n$ .

Решение задачи многокритериальной оптимизации в общем случае не является оптимальным ни для одного из частных критериев, а оказывается некоторым компромиссом для вектора  $\Phi(\mathbf{X})$  в целом.

Задачу многокритериальной оптимизации будем записывать в виде

$$\min_{\mathbf{X} \in D_x} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*), \quad (1)$$

где  $D_x$  — множество допустимых значений вектора варьируемых параметров  $\mathbf{X}$ .

Прежде, чем применить тот или иной метод решения задачи (1), обычно производят нормализацию частных критериев, приводя все частные критерии оптимальности  $\phi_k(\mathbf{X}), k \in [1, s]$  к одному масштабу. Чаще всего при этом используют относительные отклонения частных критериев от их минимальных значений:

$$\bar{\phi}_k(\mathbf{X}) = \frac{\phi_k(\mathbf{X}) - \phi_k^*}{\phi_k^{**} - \phi_k^*}, \quad k \in [1, s]$$

где

$$\phi_k^* = \min_{\mathbf{X} \in D_x} \phi_k(\mathbf{X}),$$

$$\phi_k^{**} = \max_{\mathbf{X} \in D_x} \phi_k(\mathbf{X}).$$

Метод относится к классу стохастических методов оптимизации.

Сохраним за нормализованными частными критериями оптимальности обозначения  $\phi_k(\mathbf{X}), k \in [1, s]$ .

Множество Парето.

Введем понятие пространства критериев  $\{\Phi\}$ . Пространство критериев имеет размерность  $s$  (по числу частных критериев) и образуется  $s$  ортогональными осями координат, вдоль которых откладываются значения частных критериев оптимальности  $\phi_k(\mathbf{X}), k \in [1, s]$ .

Векторный критерий оптимальности  $\Phi(\mathbf{X})$  выполняет отображение множества допустимых значений  $D_x \in \{\mathbf{X}\}$  в некоторую область  $D_\Phi \in \{\Phi\}$ , где  $\{\mathbf{X}\}$  — пространство варьируемых параметров (см. рис. 1).

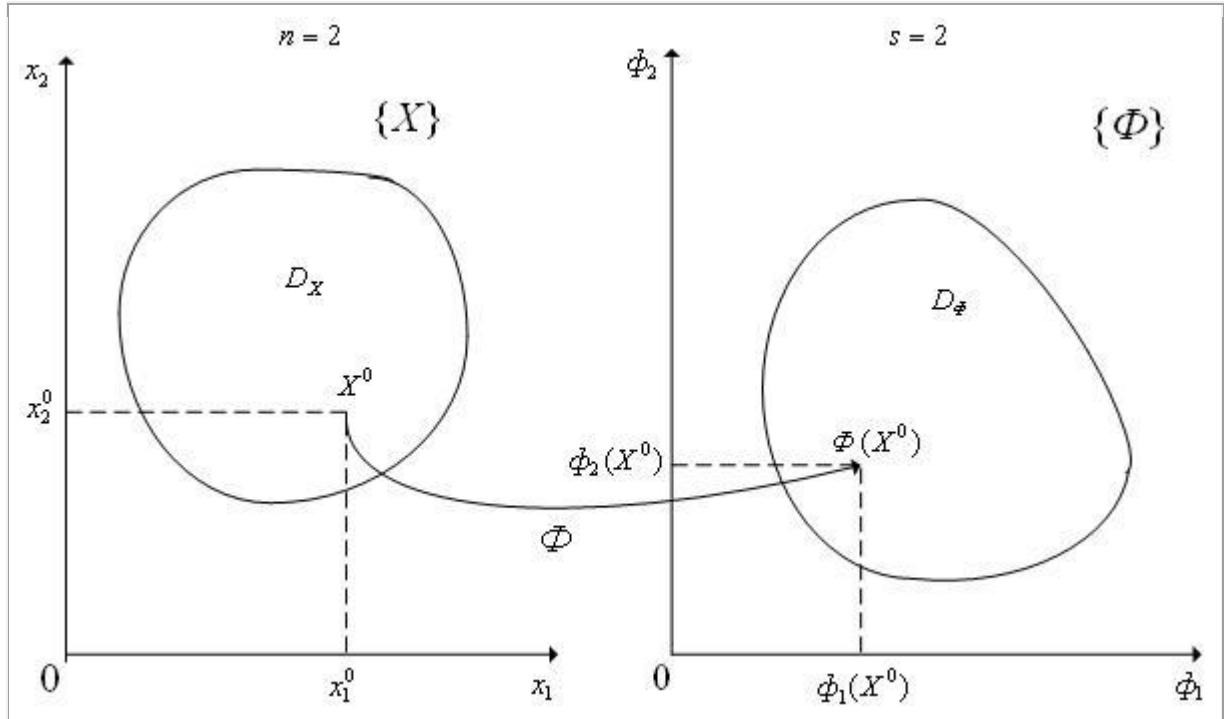


Рис. 1. К отображению векторным критерием оптимальности  $\Phi(X)$  множества допустимых значений  $D_X$  пространства варьируемых параметров  $\{X\}$  в область  $D_\Phi$  пространства критериев  $\{\Phi\}$ . Случай  $n=2, s=2$ .

Введем на множестве  $D_X$  отношение предпочтения.

Отношение предпочтения  $\succ$ . Будем говорить, что вектор  $\mathbf{X}^1 \in D_X$  предпочтительнее вектора  $\mathbf{X}^2 \in D_X$ , и писать  $\mathbf{X}^1 \succ \mathbf{X}^2$ , если среди равенств и неравенств  $\phi_k(\mathbf{X}^1) \leq \phi_k(\mathbf{X}^2), k \in [1, s]$  имеется хотя бы одно строгое неравенство (см. рис. 2).

Аналогично на множестве  $D_\Phi$  введем отношение доминирования: будем говорить, что векторный критерий оптимальности  $\Phi(\mathbf{X}^1) \in D_\Phi$  доминирует векторный критерий оптимальности  $\Phi(\mathbf{X}^2) \in D_\Phi$ , и писать  $\Phi(\mathbf{X}^1) \triangleright \Phi(\mathbf{X}^2)$ , если  $\mathbf{X}^1 \succ \mathbf{X}^2$  (см. рис. 2).

Примечание 1

Введенные отношение предпочтения и отношение доминирования являются транзитивными, т.е.

если  $\mathbf{X}^1 \succ \mathbf{X}^2$  и  $\mathbf{X}^2 \succ \mathbf{X}^3$ , то  $\mathbf{X}^1 \succ \mathbf{X}^3$ ;  
если  $\Phi(\mathbf{X}^1) \triangleright \Phi(\mathbf{X}^2)$  и  $\Phi(\mathbf{X}^2) \triangleright \Phi(\mathbf{X}^3)$ , то  $\Phi(\mathbf{X}^1) \triangleright \Phi(\mathbf{X}^3)$ .

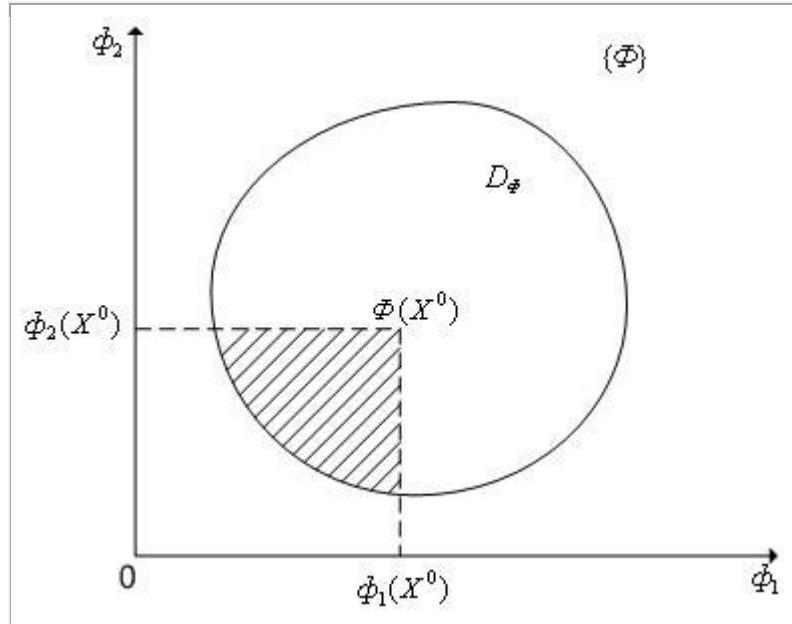


Рис. 2. К понятию отношения предпочтения и отношения доминирования ( $s = 2$ ). Для всех точек заштрихованной области  $\Phi(X_1) \succ \Phi(X_2)$ , т.е. заштрихованной области пространства критериев соответствуют векторы варьируемых параметров  $X \in D\Phi$ , для которых  $X_1 \succ X_2$ .

Выделим из множества  $D\Phi$  подмножество  $D^*_\Phi \subset D\Phi$  точек, для которых нет точек, их доминирующих. Множество  $D^*_X \subset D_X$ , соответствующее  $D^*_\Phi$ , называется множеством Парето (переговорным множеством, областью компромисса) — см. рис. 3. Поскольку

множество  $D\Phi$  на рисунке является выпуклым, то множество  $D^*_\Phi$  — есть часть границы множества  $D\Phi$  — дуга АВ, в которой точка А соответствует  $(\phi_2)_{\min}$ , а точка В —  $(\phi_1)_{\min}$ . Среди точек  $\Phi(X_1) \in D^*_\Phi$ ,  $\Phi(X_2) \in D^*_\Phi$  нет более предпочтительных, поскольку  $\phi_1(X_1) > \phi_1(X_2)$ , но  $\phi_2(X_1) > \phi_2(X_2)$ .

Таким образом, если  $X \in D^*_X$ , то  $\Phi(X) \in D^*_\Phi$ .

Другими словами множество Парето можно определить как множество, в котором значение любого из частных критериев оптимальности можно улучшить только за счет ухудшения других частных критериев — любое из решений, принадлежащее множеству Парето, не может быть улучшено одновременно по всем частным критериям.

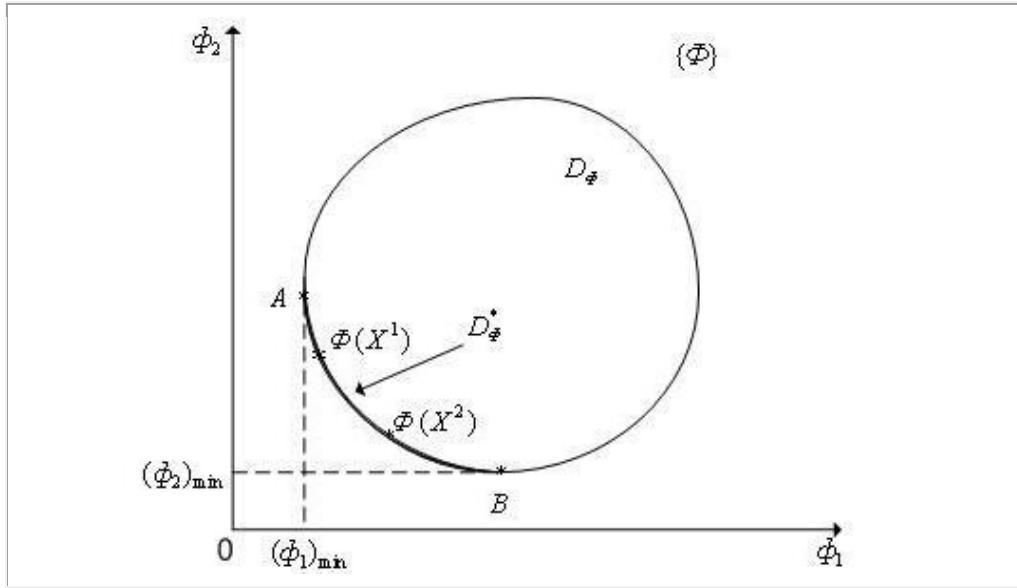


Рис. 3. К определению множества Парето ( $s = 2$ ).

Пример 1

Пусть множество допустимых значений  $D_X = \{\mathbf{X} \mid -1 \leq x_i \leq 1, i \in [1, 2]\}$  (см. рис. 4) и заданы два частных критерия оптимальности  $\phi_1(\mathbf{X}) = x_1^2 + x_2^2$ ,  $\phi_2(\mathbf{X}) = (x_1 - 1)^2 + (x_2 - 1)^2$ . Можно показать, что множество  $D_\Phi$  при этом имеет вид, представлены на рис. 5, а множество Парето  $D_X^*$  представляет собой отрезок прямой, соединяющий точки  $(0, 0)$ ,  $(1, 1)$  — см. рис. 4.

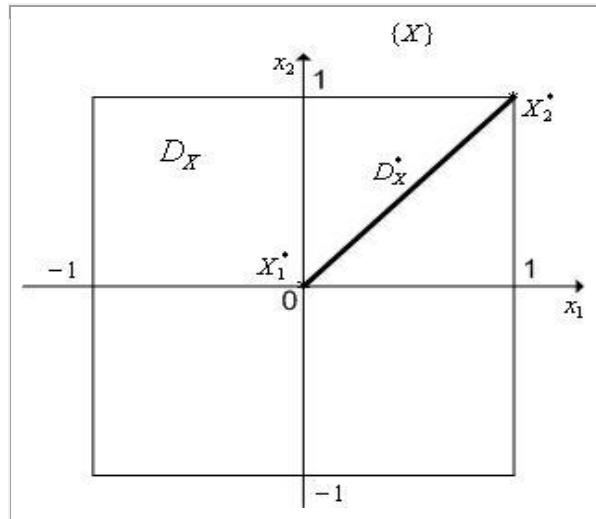


Рис. 4. К прим. 1.

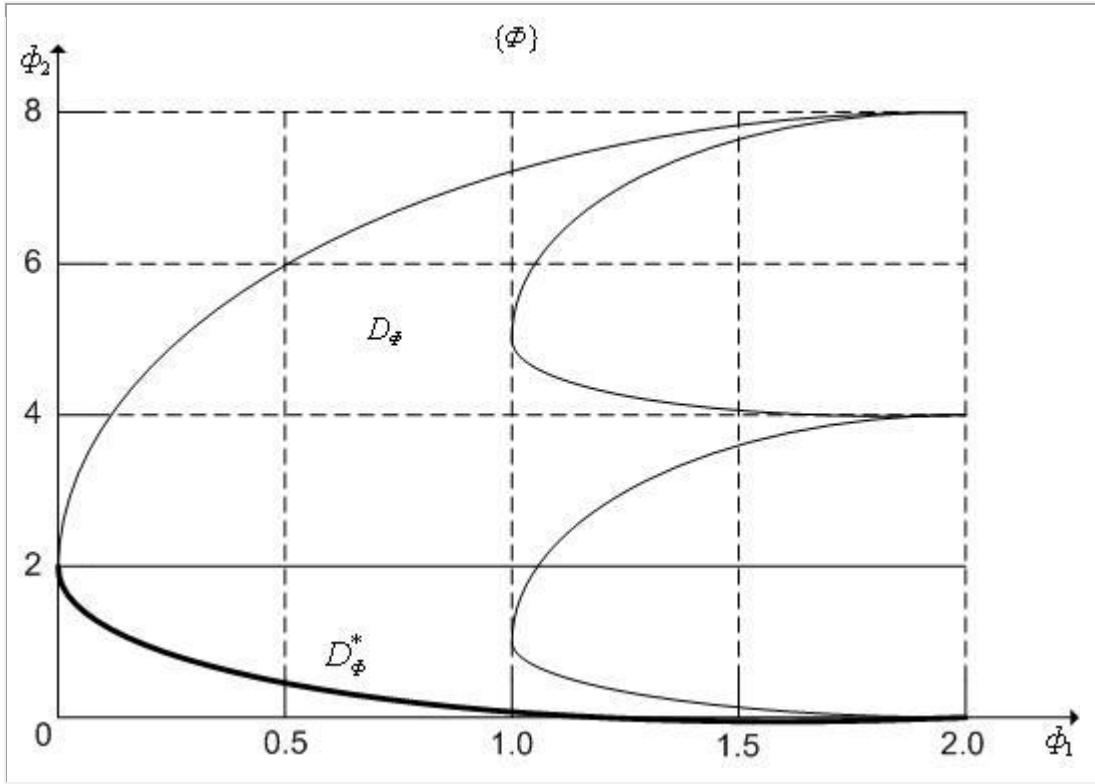


Рис. 5. К прим. 1.

Заметим, что точки  $(0,0)$ ,  $(1,1)$  являются точками минимума частных критериев оптимальности  $\phi_1(\mathbf{X})$ ,  $\phi_2(\mathbf{X})$ , соответственно.

Роль множества Парето при решении задач многокритериальной оптимизации определяется следующей теоремой.

Теорема 1. Если для некоторых весовых множителей  $\lambda_k$ ,  $k \in [1, s]$  и вектора  $\mathbf{X}^* \in D_X$  имеет место равенство

$$\sum_{k=1}^s \lambda_k \phi_k(\mathbf{X}^*) = \min_{\mathbf{X} \in D_X} \sum_{k=1}^s \lambda_k \phi_k(\mathbf{X}) \quad (2)$$

то вектор  $\mathbf{X}^*$  оптimalен по Парето, т.е.  $\mathbf{X}^* \in D_X$ .

Доказательство. Пусть вектор  $\mathbf{X}^*$  не оптimalен по Парето. Тогда существует такой вектор  $\mathbf{X} \in D_X$ , что

$$\phi_k(\mathbf{X}) \leq \phi_k(\mathbf{X}^*), \quad k \in [1, s], \quad (3)$$

причем хотя бы одно из неравенств строгое. Умножая каждое из неравенств (3) на  $\lambda_k > 0$  и складывая, получим

$$\sum_{k=1}^s \lambda_k \phi_k(\mathbf{X}) < \sum_{k=1}^s \lambda_k \phi_k(\mathbf{X}^*)$$

что противоречит условию теоремы •

Теорема 1 доказана для случая неотрицательных весовых коэффициентов  $\lambda_k$ ,  $k \in [1, s]$ , хотя не содержит этого условия. Однако доказательство возможно и для произвольных весовых коэффициентов.

Теорема показывает, что выбор определенной точки из множества Парето эквивалентен указанию весов для каждого из частных критериев оптимальности. На этом факте

основано большое количество численных методов решения задач многокритериальной оптимизации.

Заметим, что теорема 1 задает лишь необходимое условие оптимальности по Парето вектора  $\mathbf{X}^* \in D_X$ . Т.е. из того факта, что точка  $\mathbf{X}^*$  принадлежит множеству Парето, не следует, что эта точка обязательно удовлетворяет условию (2).

В постановке задачи многокритериальной оптимизации (1) фиксируется лишь множество допустимых значений  $D_X$  и вектор частных критериев  $\Phi(\mathbf{X}) = (\phi_1(\mathbf{X}), \phi_2(\mathbf{X}), \dots, \phi_s(\mathbf{X}))$ . Этой информации недостаточно для однозначного решения задачи (1). Указанная информация позволяет лишь выделить соответствующее множество Парето (можно сказать, что решением задачи многокритериальной оптимизации в постановке (1) является множество Парето). Для однозначного решения задачи (1) нужна дополнительная информация.

## 11.2. Метод весовых множителей решения задачи многокритериальной оптимизации

Рассмотрим задачу многокритериальной оптимизации

$$\min_{\mathbf{X} \in D_{\mathbf{X}}} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*), \quad (1)$$

где  $\Phi(\mathbf{X}) = (\phi_1(\mathbf{X}), \phi_2(\mathbf{X}), \dots, \phi_s(\mathbf{X}))$  — векторный критерий оптимальности,  $\phi_k(\mathbf{X}), k \in [1, s]$  — частные критерии оптимальности (скалярные),  $D_{\mathbf{X}}$  — множество допустимых значений вектора варьируемых параметров.

Для решения задачи многокритериальной оптимизации (1) широко используются методы, основанные на сведении задачи многокритериальной оптимизации к задаче однокритериальной оптимизации. Рассмотрим один из методов этой группы методов — метод весовых множителей.

В метод весовых множителей дополнительной информацией (относительно информации, заданной в постановке задачи (1)) является информация об относительной важности частных критериев. Метод требует, чтобы эта информация была formalизована в значениях весовых множители  $\lambda_k, k \in [1, s]$ . В этом случае в качестве скалярного критерия используется критерий

$$\varphi(\mathbf{X}) = \sum_{k=1}^s \lambda_k \phi_k(\mathbf{X}), \lambda_k \geq 0, \sum_{k=1}^s \lambda_k = 1. \quad (2)$$

Т.е. вместо задачи (1) решается многомерная задача условной оптимизации со скалярным критерием оптимальности (2)

$$\min_{\mathbf{X} \in D_{\mathbf{X}}} \varphi(\mathbf{X}) = \min_{\mathbf{x} \in D_{\mathbf{x}}} \sum_{i=1}^s \lambda_i \phi_i(\mathbf{x}) = \varphi(\mathbf{x}^*). \quad (3)$$

Напомним следующее: из теоремы 1 (см. параграф 1 данной главы) вытекает, что вектор  $\mathbf{X}^*$ , являющийся решением задачи условной оптимизации (3), принадлежит множеству Парето задачи (1), а обратное утверждение неверно — вектор  $\mathbf{x}^*$ , принадлежащий множеству Парето задачи (1), не обязательно удовлетворяет условию (3).

Существуют различные способы выбора весовых множителей  $\lambda_k, k \in [1, s]$ . Одним из таких способов является назначение коэффициентов в зависимости от относительной важности соответствующих частных критериев оптимальности, например, согласно табл. 1.

Таблица 1

Относительная важность критерия	Определение относительной важности критериев
1	Равная важность
3	Умеренное (слабое) превосходство
5	Сильное (существенное) превосходство
7	Очевидное превосходство
9	Абсолютное (подавляющее) превосходство
2,4,6,8	Промежуточные решения между двумя соседними оценками

Шкала относительной важности частных критериев.

Для того чтобы при выборе весовых множителей  $\lambda_k, k \in [1, s]$  избавиться от влияния масштабов частных критериев оптимальности, в методе весовых множителей целесообразно использовать нормализованные критерии.

Дадим геометрическую интерпретацию метода. Введем в рассмотрение ( $s * 1$ ) вектор  $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_s)$ . Тогда критерий оптимальности (2) можно записать в виде скалярного произведения

$$\varphi(\mathbf{X}) = \langle \Lambda, \Phi \rangle, \quad (4)$$

а задачу (3) в виде

$$\min_{\mathbf{X} \in D_X} \varphi(\mathbf{X}) = \min_{\mathbf{X} \in D_X} \langle \Lambda, \Phi \rangle = \varphi(\mathbf{X}^*). \quad (5)$$

Уравнение  $\langle \Lambda, \Phi \rangle = c$ , где  $c$  — некоторая константа, определяет в пространстве критериев  $\{\Phi\}$  гиперплоскость. При этом решение задачи (5) можно интерпретировать как поиск такого значения  $c$ , при котором гиперплоскость  $\langle \Lambda, \Phi \rangle = c$  будет касательной к множеству  $D_{\Phi}^*$  задачи (1). Компоненты вектора  $\Lambda$  определяют искомую точку касания этой гиперплоскости с множеством  $D_{\Phi}^*$  (см. рис. 1). На рис. 1 для любой точки  $\Phi^*$  множества  $D_{\Phi}^*$  (дуги A,B) найдется вектор весовых множителей  $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_s)$ , при котором эта точка удовлетворяет условию (5).

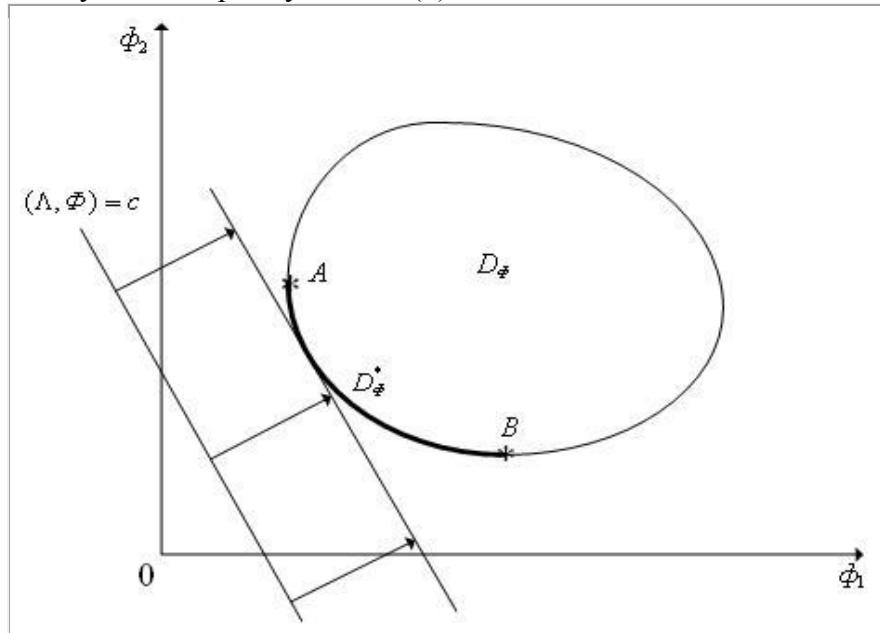


Рис. 1. Геометрическая интерпретация метода весовых множителей: случай двух критериев; множество  $D_{\Phi}$  выпукло.

Множество  $D_{\Phi}$  задачи (1) может быть не выпуклым. В этом случае не все точки множества  $D_{\Phi}^*$  могут быть достигнуты с помощью изменения весовых множителей  $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_s)$  (см. рис. 2). На рис. 2 ни для одной точки  $\Phi^*$  множества  $D_{\Phi}^*$ , принадлежащей дуге A1B1, невозможно найти вектор весовых множителей  $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_s)$ , при котором эта точка удовлетворяет условию (5).

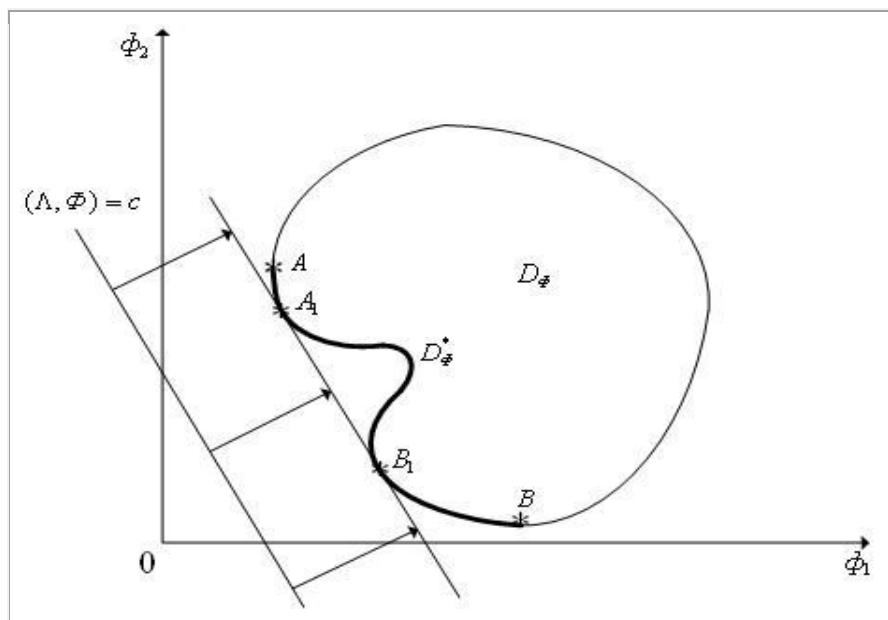


Рис. 2. Геометрическая интерпретация метода весовых множителей: случай двух критериев; множество  $D\Phi$  не выпукло.

### 11.3. Метод эпсилон-ограничений решения задачи многокритериальной оптимизации

Рассмотрим задачу многокритериальной оптимизации

$$\min_{\mathbf{X} \in D_X} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*), \quad (1)$$

где  $\Phi(\mathbf{X}) = (\phi_1(\mathbf{X}), \phi_2(\mathbf{X}), \dots, \phi_s(\mathbf{X}))$  — векторный критерий оптимальности,  $\phi_k(\mathbf{X}), k \in [1, s]$  — частные критерии оптимальности (скалярные),  $D_X$  — множество допустимых значений вектора варьируемых параметров.

Метод  $\varepsilon$ -ограничений относится к группе методов, основанных на сведении задачи многокритериальной оптимизации к однокритериальной многомерной задаче условной оптимизации.

В метод  $\varepsilon$ -ограничений в качестве скалярного критерия оптимальности  $\varphi(\mathbf{X})$  используется самый важный из частных критериев оптимальности  $\phi_p(\mathbf{X})$ , а остальные частные критерии учитываются с помощью ограничений типа неравенств вида  $\phi_k(\mathbf{X}) \leq \varepsilon_k, k \in [1, s], k \neq p$ .

Дополнительной информацией в методе  $\varepsilon$ -ограничений является информация о номере самого важного из частных критериев, а также информация о максимально допустимых значениях частных критериев.

Таким образом, в методе  $\varepsilon$ -ограничений вместо задачи (1) решается задача условной оптимизации со скалярным критерием оптимальности  $\varphi_p(\mathbf{X})$ :

$$\min_{\mathbf{X} \in \tilde{D}} \varphi(\mathbf{X}) = \min_{\mathbf{X} \in \tilde{D}} \phi_p(\mathbf{X}) = \varphi(\mathbf{X}^*). \quad (2)$$

где

$$\tilde{D} = D_X \cap D_p, D_p = \{\mathbf{X} \mid \phi_k(\mathbf{X}) \leq \varepsilon_k, k \in [1, s], k \neq p\}. \quad (3)$$

Метод  $\varepsilon$ -ограничений в значительной мере свободен от указанного выше недостатка метода весовых множителей в случае, когда множество  $D_\Phi$  не выпукло (см. рис. 1). На рис. 1 точка A2 в данном методе является доступной.

Недостатком метода  $\varepsilon$ -ограничений является трудность выбора максимально допустимых значений частных критериев  $\varepsilon_k, k \in [1, s], k \neq p$ , которые гарантировали бы достижимость некоторого решения. Кроме того, жесткость ограничений  $\phi_k(\mathbf{X}) \leq \varepsilon_k, k \in [1, s], k \neq p$  далеко не всегда адекватна представлениям ЛПР (лица, принимающего решения) о наилучшем решении. Отметим также трудность построения в явном или неявном виде множества  $D_P$ .

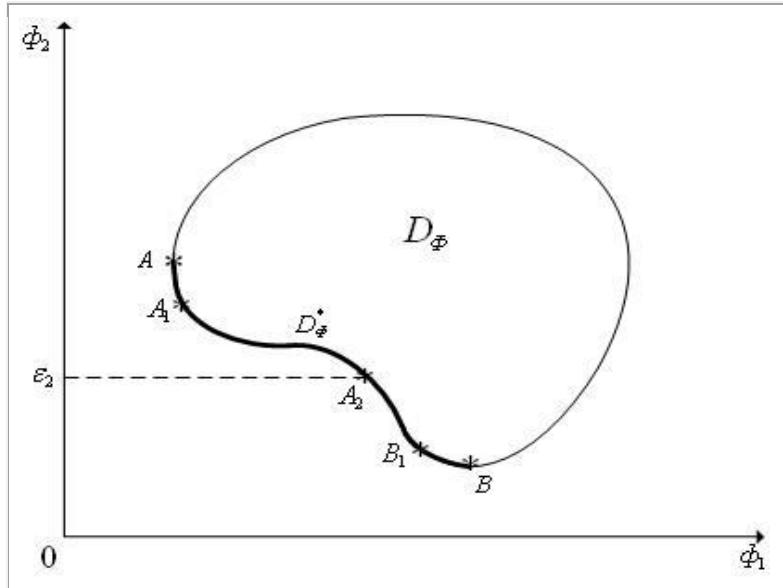


Рис. 1. Геометрическая интерпретация метода  $\varepsilon$ -ограничений: случай двух критериев; множество  $D\Phi$  не выпукло; самым важным является критерий  $\phi_1(X)$ ; на критерий  $\phi_2(X)$  наложено ограничение  $\phi_2(X) \leq \varepsilon_2$ .

#### 11.4. Метод справедливого компромисса для решения задач многокритериальной оптимизации

Рассмотрим задачу многокритериальной оптимизации

$$\min_{\mathbf{X} \in D_X} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*), \quad (1)$$

где  $\Phi(\mathbf{X}) = (\phi_1(\mathbf{X}), \phi_2(\mathbf{X}), \dots, \phi_s(\mathbf{X}))$  — векторный критерий оптимальности,  $\phi_k(\mathbf{X}), k \in [1, s]$  — частные критерии оптимальности (скалярные),  $D_X$  — множество допустимых значений вектора варьируемых параметров.

Положим, что все частные критерии  $\phi_k(\mathbf{X}), k \in [1, s]$  имеют одинаковую важность! Справедливый компромисс.

Справедливым компромиссом будем называть такой компромисс, при котором относительный уровень снижения качества по одному или нескольким частным критериям не превосходит относительного уровня повышения качества по остальным частным критериям (меньше или равен).

Для формализации понятия справедливого компромисса нам понадобится ввести отношение превосходства  $\succ$  на множестве Парето (не путать с отношением доминирования  $\succeq$ , используемым при построении множества Парето!).

Пусть во множестве Парето задачи (1) даны две точки  $\mathbf{X}^1 \in D_X^*, \mathbf{X}^2 \in D_X^*$  и значения всех частных критериев оптимальности в них  $\phi_k(\mathbf{X}^1), \phi_k(\mathbf{X}^2), k \in [1, s]$ . Введем меру относительного изменения (снижения — знак "минус" или повышения — знак "плюс") качества решения по каждому из этих критериев

$$\Delta \tilde{\phi}_k(\mathbf{X}^1, \mathbf{X}^2) = \frac{\Delta \phi_k(\mathbf{X}^1, \mathbf{X}^2)}{\max_{\mathbf{X} \in (\mathbf{X}^1, \mathbf{X}^2)} \phi_k(\mathbf{X})}, \quad k \in [1, s], \quad (2)$$

где  $\Delta \phi_k(\mathbf{X}^1, \mathbf{X}^2) = \phi_k(\mathbf{X}^1) - \phi_k(\mathbf{X}^2)$  — абсолютные изменения значений частных критериев оптимальности  $\phi_k(\mathbf{X}), k \in [1, s]$  при переходе от решения  $\mathbf{X}^1$  к решению  $\mathbf{X}^2$ . Вычислим максимальное снижение качества решения при переходе от решения  $\mathbf{X}^1$  к решению  $\mathbf{X}^2$

$$\Delta \tilde{\phi}_{\min}(\mathbf{X}^1, \mathbf{X}^2) = \min_{k \in [1, s]} \Delta \tilde{\phi}_k(\mathbf{X}^1, \mathbf{X}^2). \quad (3)$$

Аналогично вычислим максимальное повышение качества решения при переходе от решения  $\mathbf{X}^1$  к решению  $\mathbf{X}^2$

$$\Delta \tilde{\phi}_{\max}(\mathbf{X}^1, \mathbf{X}^2) = \max_{k \in [1, s]} \Delta \tilde{\phi}_k(\mathbf{X}^1, \mathbf{X}^2). \quad (4)$$

Будем говорить, что решение  $\mathbf{X}^2$  превосходит решение  $\mathbf{X}^1$ , и писать

$$\mathbf{X}^2 \succ \mathbf{X}^1, \text{ если } |\Delta \tilde{\phi}_{\max}(\mathbf{X}^1, \mathbf{X}^2)| > |\Delta \tilde{\phi}_{\min}(\mathbf{X}^1, \mathbf{X}^2)|. \quad (5)$$

С другой стороны, будем говорить, что решение  $\mathbf{X}^1$  превосходит решение  $\mathbf{X}^2$ , и писать

$$\mathbf{X}^1 \succ \mathbf{X}^2, \text{ если } |\Delta\tilde{\phi}_{\max}(\mathbf{X}^1, \mathbf{X}^2)| \leq |\Delta\tilde{\phi}_{\min}(\mathbf{X}^1, \mathbf{X}^2)|. \quad (6)$$

Пример 1

Пусть заданы четыре частных критерия оптимальности  $\phi_k(\mathbf{X}), k \in [1, 4]$ , ( $s = 4$ ) и решения  $\mathbf{X}^1, \mathbf{X}^2$  принадлежащие множеству Парето задачи (1). Положим, что критерии  $\phi_k(\mathbf{X}), k \in [1, 4]$  в точках  $\mathbf{X}^1, \mathbf{X}^2$  имеют следующие значения:

Таблица 1

	$\phi_1(\mathbf{X})$	$\phi_2(\mathbf{X})$	$\phi_3(\mathbf{X})$	$\phi_4(\mathbf{X})$
$\mathbf{X}^1$	1	2	3	5
$\mathbf{X}^2$	3	2	0	4

По формулам (2), (3), (4) последовательно имеем

$$\Delta\tilde{\phi}_1(\mathbf{X}^1, \mathbf{X}^2) = \frac{1 - 3}{3} = -\frac{2}{3},$$

$$\Delta\tilde{\phi}_2(\mathbf{X}^1, \mathbf{X}^2) = \frac{2 - 2}{2} = 0,$$

$$\Delta\tilde{\phi}_3(\mathbf{X}^1, \mathbf{X}^2) = \frac{3 - 0}{3} = 1,$$

$$\Delta\tilde{\phi}_4(\mathbf{X}^1, \mathbf{X}^2) = \frac{5 - 4}{5} = \frac{1}{5},$$

$$\Delta\tilde{\phi}_{\min}(\mathbf{X}^1, \mathbf{X}^2) = \min(-\frac{2}{3}, 0, 1, \frac{1}{5}) = -\frac{2}{3},$$

$$\Delta\tilde{\phi}_{\max}(\mathbf{X}^1, \mathbf{X}^2) = \max(-\frac{2}{3}, 0, 1, \frac{1}{5}) = 1.$$

Поскольку  $|\Delta\tilde{\phi}_{\max}(\mathbf{X}^1, \mathbf{X}^2)| > |\Delta\tilde{\phi}_{\min}(\mathbf{X}^1, \mathbf{X}^2)|$ , т.е. максимальное относительное повышение качества решения превышает максимальное относительное снижение качества решения, то решение  $\mathbf{X}^2$  превосходит решения  $\mathbf{X}^1$ :  $\mathbf{X}^2 \succ \mathbf{X}^1$ .

Таким образом, дополнительной информацией в методе справедливого компромисса является информация об одинаковой важности всех частных критериев  $\phi_p(\mathbf{X})$ , а также информация о справедливом компромиссе, formalизованная отношением превосходства  $\succ$ .

Упрощенная схема метода справедливого компромисса.

Рассмотрим основные этапы метода справедливого компромисса.

Полагаем счетчик числа итераций  $r = 1$ .

Тем или иным способом выбираем из множества Парето задачи (1) решение  $\mathbf{X}^0 = \tilde{\mathbf{X}}^*$ . Вычисляем значения всех частных критериев оптимальности  $\phi_k(\mathbf{X}), k \in [1, s]$  в точке  $\mathbf{X}^0$ .

Тем или иным способом выбираем из множества Парето задачи (1) решение  $\mathbf{X}^r$ .

Вычисляем значения всех частных критериев оптимальности  $\phi_k(\mathbf{X}), k \in [1, s]$  в точке  $\mathbf{X}^r$ .

По формулам (2), (3), (4), (5), (6) из числа решений  $\mathbf{X}^{r+1}$ ,  $\mathbf{X}^r$  находим превосходящее решение. Обозначим его  $\tilde{\mathbf{X}}^*$ .

Если условие окончания итераций выполнено (см. ниже), то принимаем точку  $\tilde{\mathbf{X}}^*$  в качестве приближенного решения задачи (1) и заканчиваем вычисления. Иначе полагаем  $r = r + 1$  и переходим к п.3 ●

В простейшем случае выбор решений может быть произведен случайным образом. В качестве условия окончания итераций в этом случае может быть использовано достижение заданного количества итераций.

Выбор решений  $\mathbf{X}^r \in D_X^*$  может быть произведен также с помощью полного перебора узлов какой-либо сетки, покрывающей множество  $D_X^*$ .

Поскольку метод справедливого компромисса использует относительные изменения частных критериев оптимальности, этот метод инвариантен к масштабу измерения частных критериев, т.е. не требуется их нормализация.

## 11.5. Метод приближения к идеальному решению для решения задач многокритериальной оптимизации

Рассмотрим задачу многокритериальной оптимизации

$$\min_{\mathbf{X} \in D_X} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*), \quad (1)$$

где  $\Phi(\mathbf{X}) = (\phi_1(\mathbf{X}), \phi_2(\mathbf{X}), \dots, \phi_s(\mathbf{X}))$  — векторный критерий оптимальности,  $\phi_k(\mathbf{X}), k \in [1, s]$  — частные критерии оптимальности (скалярные),  $D_X$  — множество допустимых значений вектора варьируемых параметров.

Метод приближения к идеальному решению относится к группе методов, основанных на сведении задачи многокритериальной оптимизации к задаче однокритериальной оптимизации.

Идеальным решением задачи многокритериальной оптимизации (1) называется вектор  $\Phi^* = (\phi_1^*, \phi_2^*, \dots, \phi_s^*)$ , где

$$\phi_k^* = \min_{\mathbf{X} \in D_X} \phi_k(\mathbf{X}), k \in [1, s] -$$

минимальное значение частного критерия оптимальности  $\phi_k(\mathbf{X})$  во множестве  $D_X$ .

Напомним, что векторы  $\mathbf{X}_k^*, k \in [1, s]$ , доставляющие минимумы соответствующим критериям оптимальности  $\phi_k(\mathbf{X})$ , вообще говоря, различны.

Введем в рассмотрение скалярный критерий оптимальности

$$\varphi(\mathbf{X}) = \|\bar{\Phi}(\mathbf{X}) - \bar{\Phi}^*\|, \quad (2)$$

где  $\|\cdot\|$  — некоторая векторная норма, например, евклидова;

$$\bar{\Phi}(\mathbf{X}) = \left( \frac{\phi_1(\mathbf{X})}{\phi_1^*}, \frac{\phi_2(\mathbf{X})}{\phi_2^*}, \dots, \frac{\phi_s(\mathbf{X})}{\phi_s^*} \right)^T \text{ — нормированный векторный критерий оптимальности;}$$

$$\bar{\Phi}^* = \left( \frac{\phi_1^*}{\phi_1^*}, \frac{\phi_2^*}{\phi_2^*}, \dots, \frac{\phi_s^*}{\phi_s^*} \right)^T = (1, 1, \dots, 1) \text{ — нормированное идеальное решение (единичный } (s * 1) \text{-вектор).}$$

В методе приближения к идеальному решению вместо задачи (1) решается задача условной оптимизации со скалярным критерием оптимальности (2)

$$\min_{\mathbf{X} \in D_X} \varphi(\mathbf{X}) = \min_{\mathbf{X} \in D_X} \|\bar{\Phi}(\mathbf{X}) - \bar{\Phi}^*\| = \varphi(\mathbf{X}^*). \quad (3)$$

Заметим, что если  $\|\cdot\|$  — евклидова норма, то критерий оптимальности (2) является квадратичной функцией компонент вектора  $\mathbf{X}$ . Поэтому если, дополнительно, множество  $D_X$  является выпуклым, то задача (3) представляет собой задачу квадратичного программирования. Этот факт значительно упрощает решение задачи (3).

Дополнительной информацией в методе приближения к идеальному решению является информация, заключенная в способе сворачивания векторного критерия оптимальности  $\Phi(\mathbf{X}) = (\phi_1(\mathbf{X}), \phi_2(\mathbf{X}), \dots, \phi_s(\mathbf{X}))$  в скалярный критерий  $\varphi(\mathbf{X})$ .

Поскольку метод приближения к идеальному решению использует нормированные частные критерии оптимальности, этот метод инвариантен к масштабу измерения частных критериев.

## 11.6. Метод последовательных уступок для решения задач многокритериальной оптимизации

Рассмотрим задачу многокритериальной оптимизации

$$\min_{\mathbf{X} \in D_X} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*), \quad (1)$$

где  $\Phi(\mathbf{X}) = (\phi_1(\mathbf{X}), \phi_2(\mathbf{X}), \dots, \phi_s(\mathbf{X}))$  — векторный критерий оптимальности,  $\phi_k(\mathbf{X}), k \in [1, s]$  — частные критерии оптимальности (скалярные),  $D_X$  — множество допустимых значений вектора варьируемых параметров.

В методе последовательных уступок прежде производится качественный анализ относительной важности частных критериев оптимальности  $\phi_1(\mathbf{X}), \phi_2(\mathbf{X}), \dots, \phi_s(\mathbf{X})$ . На основании этого анализа частные критерии располагаются и нумеруются в порядке убывания важности. Положим, что главным является критерий  $\phi_1(\mathbf{X})$ , следующим по важности — критерий  $\phi_2(\mathbf{X})$ , затем следуют остальные частные критерии  $\phi_3(\mathbf{X}), \phi_4(\mathbf{X}), \dots, \phi_s(\mathbf{X})$ .

Затем для каждого из частных критериев, исключая последний по важности критерий  $\phi_s(\mathbf{X})$ , назначаются уступки  $\Delta_k, k \in [1, s-1]$  — допустимые, с точки зрения лица, принимающего решения (ЛПР), увеличения соответствующих критериев

$\phi_k(\mathbf{X}), k \in [1, s-1]$  относительно их оптимальных значений  $\phi_k^*(\mathbf{X}) = \min_{\mathbf{X} \in D_X} \phi_k(\mathbf{X})$ .

Далее решается задача минимизации критерия  $\phi_1(\mathbf{X})$

$$\phi_1^* = \min_{\mathbf{X} \in D_X} \phi_1(\mathbf{X}) = \phi_1(\mathbf{X}_1^*) \quad (2)$$

и определяется множества допустимых значений  $D_X^1$  — сужение множества допустимых значений:

$$D_X^1 = \{\mathbf{X} \mid \phi_1(\mathbf{X}) \leq \phi_1^* + \Delta_1\} \quad (3)$$

Переходим к следующему по важности критерию  $\phi_2(\mathbf{X})$ . Решается задача минимизации критерия  $\phi_2(\mathbf{X})$

$$\phi_2^* = \min_{\mathbf{X} \in D_X^1} \phi_2(\mathbf{X}) = \phi_2(\mathbf{X}_2^*) \quad (4)$$

и определяется множества допустимых значений  $D_X^2$  — сужение множества допустимых значений:

$$D_X^2 = \{\mathbf{X} \mid \phi_1(\mathbf{X}) \leq \phi_1^* + \Delta_1, \phi_2(\mathbf{X}) \leq \phi_2^* + \Delta_2\} \quad (5)$$

Т.е. минимизация критерия  $\phi_2(\mathbf{X})$  производится при условии, что значения предыдущего критерия  $\phi_1(\mathbf{X})$  не превосходит величины  $\phi_1^* + \Delta_1$ .

И т.д. до предпоследнего по важности критерия  $\phi_{s-1}(\mathbf{X})$ . Решается задача минимизации критерия  $\phi_{s-1}(\mathbf{X})$

$$\phi_{s-1}^* = \min_{\mathbf{X} \in D_X^{s-2}} \phi_{s-1}(\mathbf{X}) = \phi_{s-1}(\mathbf{X}_{s-1}^*) \quad (6)$$

и определяется множества допустимых значений  $D_X^{s-1}$  — сужение множества допустимых значений:

$$D_X^{s-1} = \{\mathbf{X} \mid \phi_1(\mathbf{X}) \leq \phi_1^* + \Delta_1, \phi_2(\mathbf{X}) \leq \phi_2^* + \Delta_2, \dots, \phi_{s-2}(\mathbf{X}) \leq \phi_{s-2}^* + \Delta_{s-2}\} \quad (7)$$

Наконец, переходим к последнему по важности критерию  $\phi_s(\mathbf{X})$ . Решается задача минимизации критерия  $\phi_s(\mathbf{X})$

$$\phi_s^* = \min_{\mathbf{X} \in D_X^{s-1}} \phi_s(\mathbf{X}) = \phi_s(\mathbf{X}_s^*) \quad (8)$$

В качестве решения задачи (1) принимается решение  $\mathbf{X}_s^*$  со значениями частных критериев  $\phi_k(\mathbf{X}_s^*), k \in [1, s]$ .

Достоинством метода последовательных уступок является то, что он легко позволяет контролировать, ценой какой уступки в одном частном критерии приобретается выигрыш в другом частном критерии. Заметим также, что свобода выбора решения, приобретаемая ценой даже незначительных уступок, может оказаться существенной, так как в окрестностях минимума обычно эффективность решений меняется слабо.

Не смотря на идейную простоту метода последовательных уступок, практическое применение этого метода сопряжено со значительными трудностями.

Метод последовательных уступок применим для решения лишь тех задач многокритериальной оптимизации, в которых все частные критерии естественным образом упорядочены по степени важности.

Поскольку взаимосвязь частных критериев обычно неизвестна, заранее назначить величины уступок  $\Delta_k, k \in [1, s]$ , как правило, не удается. Поэтому изучение взаимосвязи частных критериев и назначение величин уступки приходится производить в процессе решения задачи. Практически, для этого вначале оценивают взаимосвязь частных критериев  $\phi_1(\mathbf{X}), \phi_2(\mathbf{X})$ . Для этого задают несколько величин уступок  $\Delta_{11}, \Delta_{12}, \dots$  и

определяют соответствующие значения второго по важности критерия  $\phi_{21}^*, \phi_{22}^*, \dots$ . На основе анализа этой информации лицо, принимающее решение, принимает решение о величине первой уступки  $\Delta_1$ . Затем аналогично оценивают взаимосвязь частных критериев  $\phi_2(\mathbf{X}), \phi_3(\mathbf{X})$  и назначают величину второй уступки  $\Delta_2$ . И так далее до критериев  $\phi_{s-1}(\mathbf{X}), \phi_s(\mathbf{X})$  и уступки  $\Delta_{s-1}$ . Таким образом, фактически приходится решать каждую из задач (2), (4), (5), (6) не однократно, как в изложенной схеме, а многократно.

Изложенная схема метода последовательных уступок предполагает, что каждый частный критерий  $\phi_k(\mathbf{X}), k \in [1, s-1]$  настолько существенно более важен, чем последующий частный критерий, что можно ограничиться учетом только попарных связей критериев и выбирать величину уступки для данного критерия с учетом поведения лишь одного следующего критерия.

Сложной самостоятельной проблемой является отыскание в явном виде множеств  $D_X^1, D_X^2, \dots, D_X^{s-1}$ .

Дополнительной информацией в методе последовательных уступок является информация об относительной важности частных критериев  $\phi_k(\mathbf{X}), k \in [1, s]$ , а также информация об уступках  $\Delta_k, k \in [1, s - 1]$ .