

Раздел 4. Нелинейное программирование. Многомерная условная оптимизация

4.1. Методы последовательной безусловной оптимизации

Рассмотрим следующую многомерную задачу локальной условной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного во множестве D евклидова пространства R^n ,

$$\min_{X \in D \subset R^n} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (4.1.1)$$

где множество допустимых значений

$$D = \left\{ X / h_i(X) = 0, g_j(X) \geq 0, i \in [1, \dots, m], j \in [1, \dots, l] \right\}. \quad (4.1.2)$$

Основная идея методов последовательной безусловной оптимизации состоит в преобразовании задачи условной оптимизации (4.1.1), (4.1.2) к последовательности задач безусловной оптимизации

$$\min_{X \in R^n} Q^\alpha(X) = \min_{X \in R^n} (\Phi(X) + P^\alpha(X)) = Q^\alpha(\tilde{X}), \quad (4.1.3)$$

где $P^\alpha(X)$ функции, которые возрастают вблизи границ области допустимых значений D и тем быстрее, чем больше значение параметра α .

В качестве приближенного решения задачи (4.1.1), (4.1.2) принимается решение \tilde{X} вспомогательной задачи (4.1.3) при достаточно большом α .

Поясним идею методов последовательной безусловной оптимизации примером.

Пример 4.1.1. Пусть

$$\Phi(X) = (x_1 - 3)^2 + (x_2 - 2)^2 \quad (4.1.4)$$

и имеется одно ограничение типа равенств с ограничивающей функцией

$$h_i(X) = x_1 + x_2 - 4. \quad (4.1.5)$$

Положим

$$P^\alpha(X) = \alpha h_1^2(X), \quad (4.1.6)$$

где $\alpha \geq 0$ — вещественная константа.

На рисунках рис. 4.1.1, рис. 4.1.2, рис. 4.1.3 приведены линии уровня функции при $\alpha = 0, 1, 2$ соответственно.

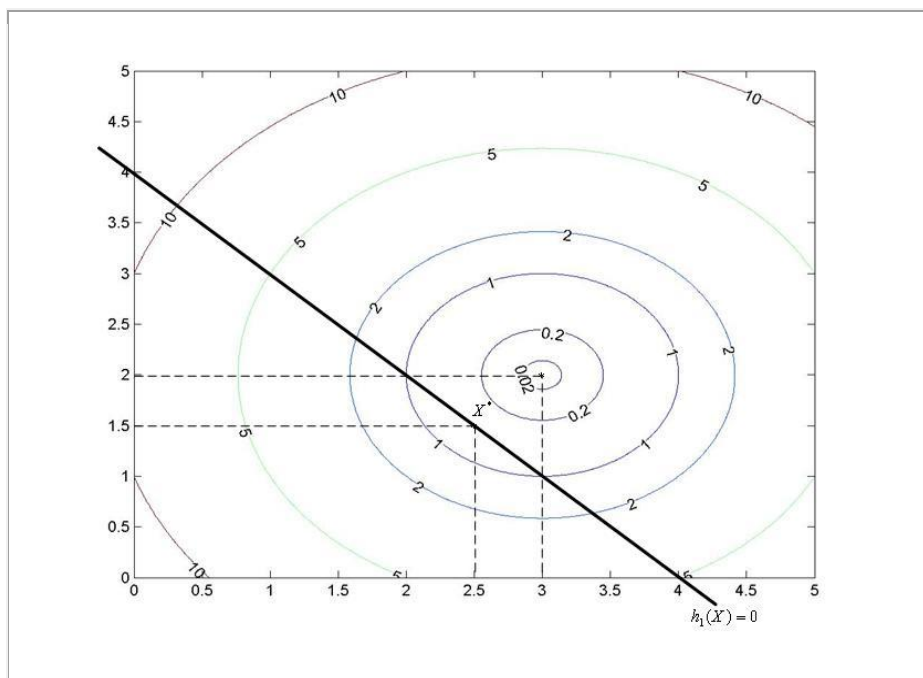


Рис. 4.1.1. К примеру 4.1.1. Точка минимума функции $Q^\alpha(X)$ при $\alpha = 0$ имеет координаты $(3, 2)$. Решением задачи (4.1.3)-(4.1.6) является точка X^* с координатами $(2.5, 1.5)$

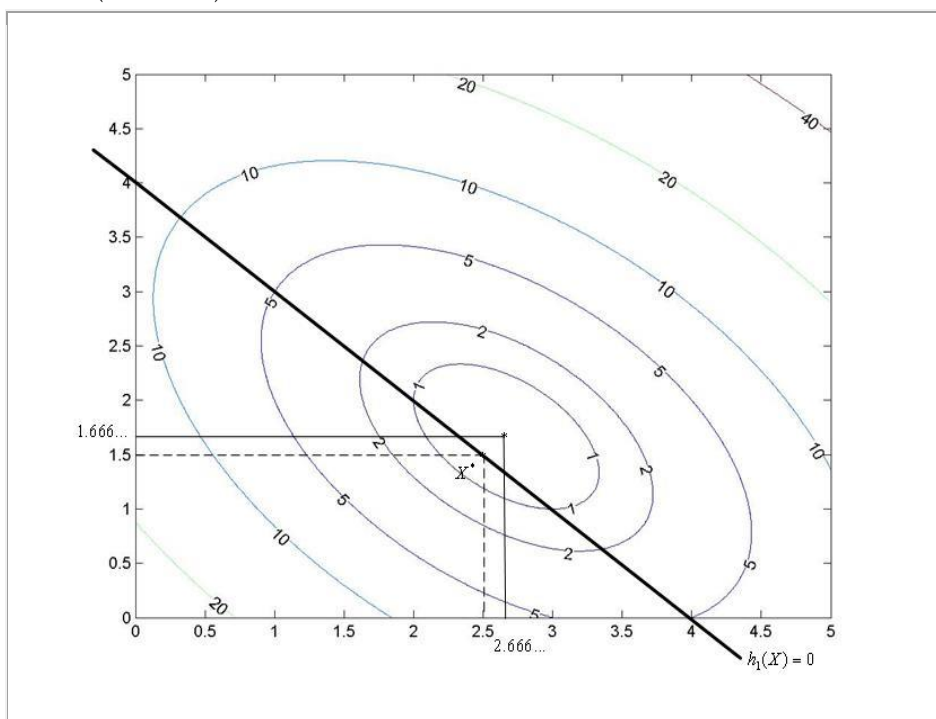


Рис. 4.1.2. К примеру 4.1.1. Точка минимума функции $Q^\alpha(X)$ при $\alpha = 1$ имеет координаты $(2.666..., 1.666...)$. Решением задачи (4.1.3)-(4.1.6) является точка X^* с координатами $(2.5, 1.5)$

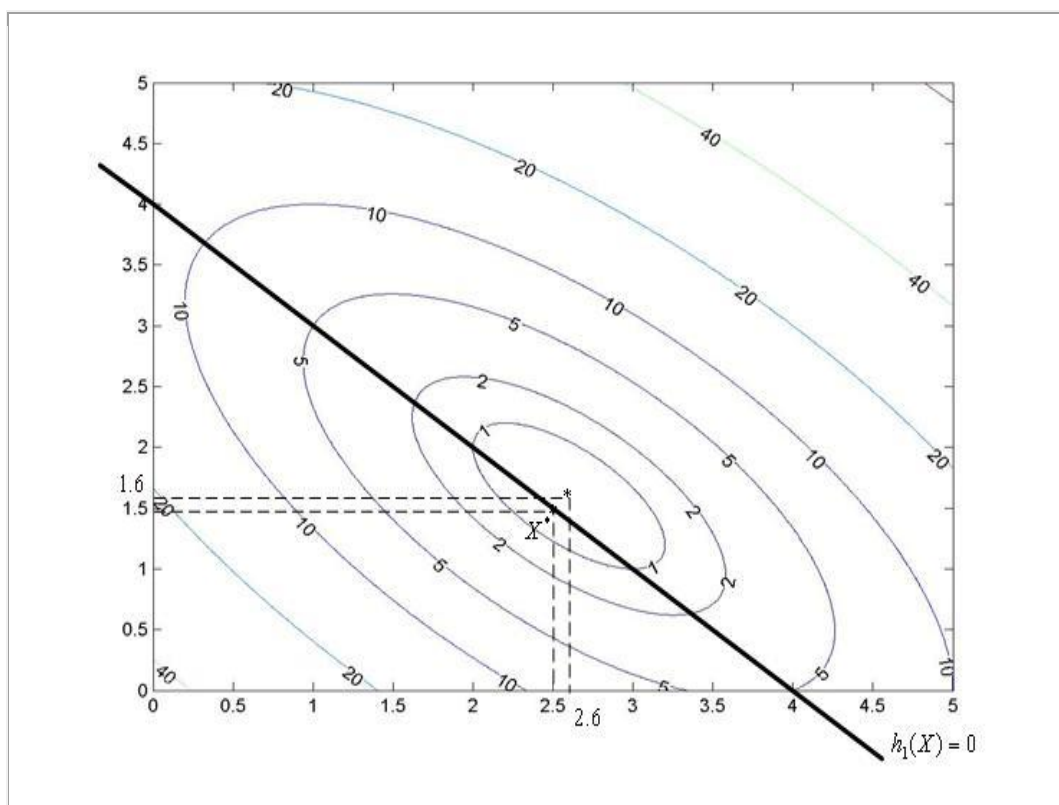


Рис. 4.1.3. К примеру 4.1.1. Точка минимума функции $Q^\alpha(X)$ при $\alpha = 2$ имеет координаты $(2.6, 1.6)$. Решением задачи (4.1.3)-(4.1.6) является точка X^* с координатами $(2.5, 1.5)$

Рисунки показывают, что при увеличении параметра α минимум функции $Q^\alpha(X)$ приближается к решению задачи (4.1.3)- (4.1.6).

Среди методов последовательной безусловной оптимизации выделяют **метод штрафных функций** и **метод барьерных функций**.

В методе штрафных функций функцию $P^\alpha(X)$, которая в этом случае называется **штрафной функцией**, подбирают таким образом, чтобы при больших α функция $Q^\alpha(X)$ мало отличалась от функции $\Phi(X)$ при $X \in D$ и быстро возрастала при удалении точки $X \notin D$ от границы области допустимых значений D . В методе штрафных функций точка X в процессе поиска может выходить за границы области D (см. рис. 4.1.4). Т.е. метод штрафных функций относится к классу **методов внешней точки**.

Рассмотренный выше пример 4.1.1 также иллюстрирует метод штрафных функций.

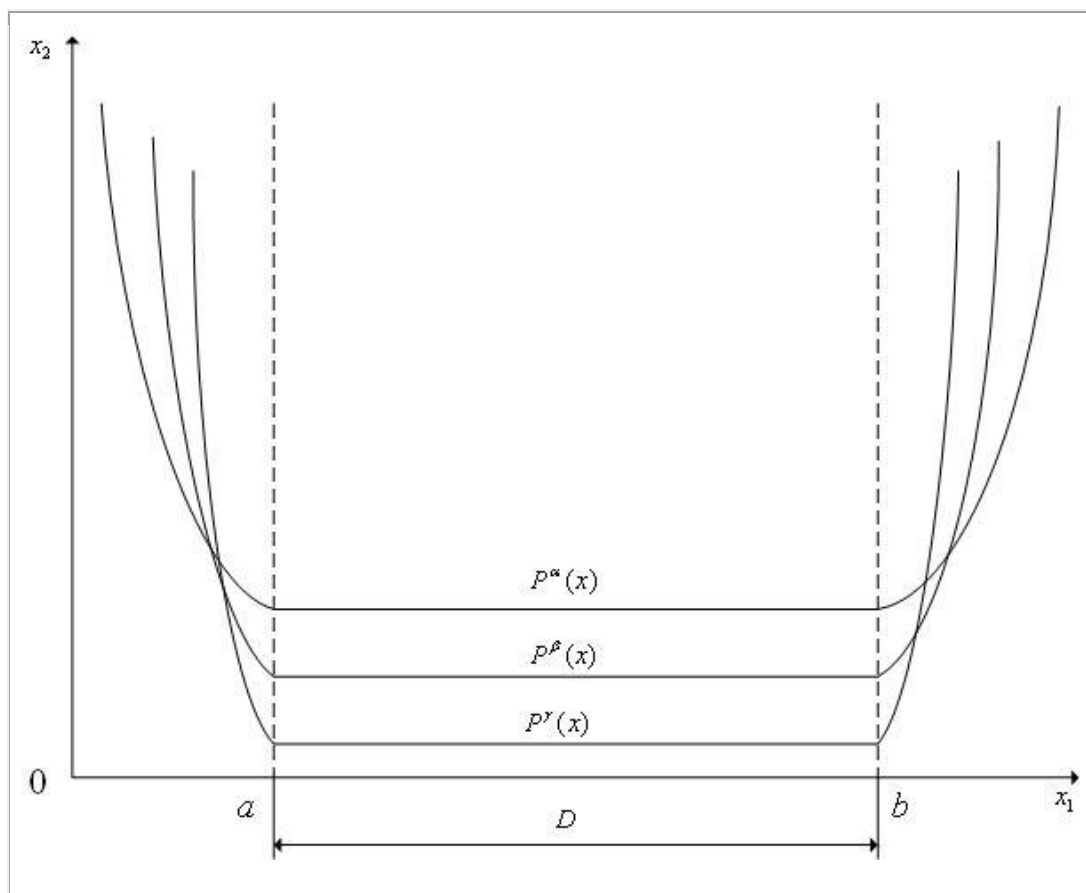


Рис. 4.1.4. К методу штрафных функций ($n=1$). Интервал $[a, b]$ — область допустимых значений D ; $\gamma > \beta > \alpha$

В методе барьерных функций функцию $P^\alpha(X)$, которая в этом случае называется **барьерной функцией**, подбирают таким образом, чтобы при больших α функция $Q^\alpha(X)$ мало отличалась от функции $\Phi(X)$ при $X \in D$ и быстро возрастала при приближении точки $X \in D$ к границе области допустимых значений D . В методе барьерных функций точка X в процессе поиска не может выходить за границы области D (см. рис. 4.1.5). Это означает, что метод барьерных функций относится к классу **методов внутренней точки**.

В вычислительной практике преимущественно используется метод штрафных функций. Поэтому в дальнейшем ограничимся именно им.

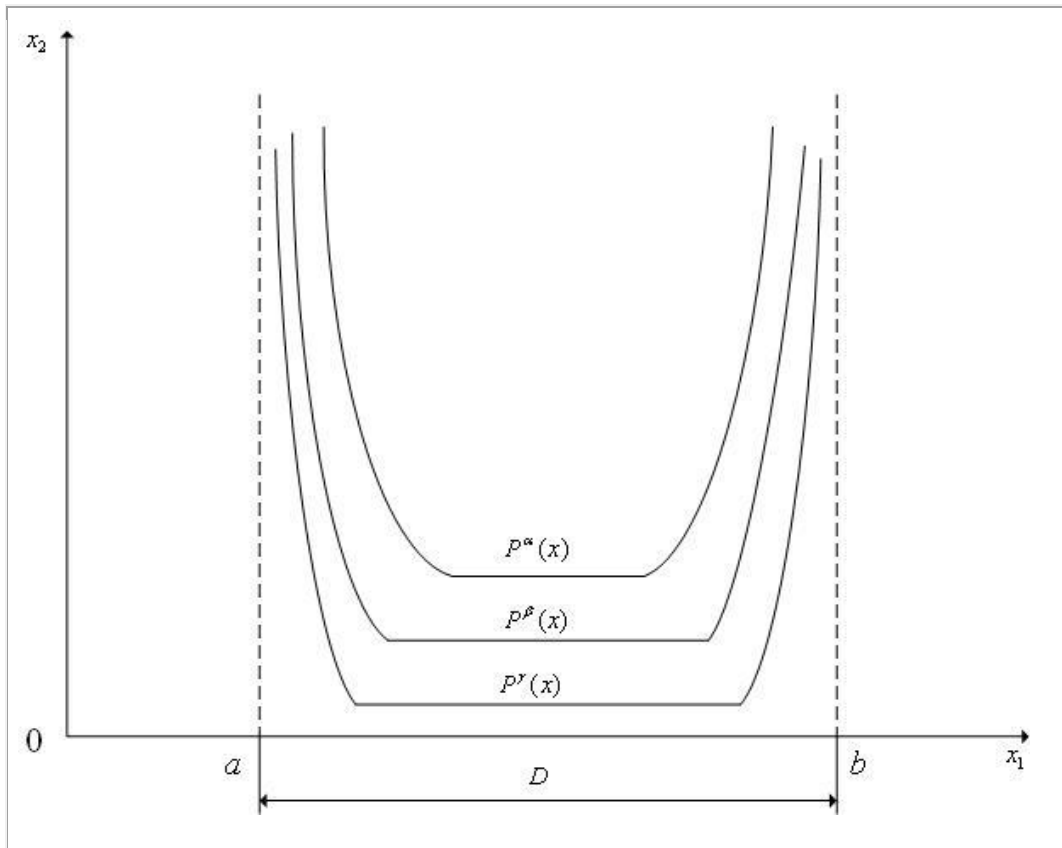


Рис. 4.1.5. К методу барьерных функций ($n=1$). Интервал $[a, b]$ — область допустимых значений D ; $\gamma > \beta > \alpha$

Штрафная функция в общем случае имеет вид

$$P(X, \alpha_h^r, \alpha_g^r) = \alpha_h^r \sum_{i=1}^m \mu_i H(h_i(X)) + \alpha_g^r \sum_{j=1}^l \lambda_j G(g_j(X)), \quad (4.1.7)$$

где $\alpha^r = (\alpha_h^r, \alpha_g^r)$ — двумерный вектор параметров штрафной функции;

μ_i, λ_j — весовые коэффициенты, могущие изменяться в процессе итераций,

$H(h_i(X)), G(g_j(X))$ — функционалы над функциями $h_i(X) \in [1, \dots, m]$,

$g_j(X) \in [1, \dots, l]$, соответственно.

Функционалы $H(h_i(X)), G(g_j(X))$ в формуле (4.1.7) должны удовлетворять очевидным требованиям:

$$H(h_i(X)) \rightarrow 0 \text{ при } h_i(X) \rightarrow 0,$$

$$G(g_j(X)) \rightarrow 0 \text{ при } g_j(X) \rightarrow 0;$$

$$\lim_{r \rightarrow \infty} \left(\alpha_h^r \sum_{i=1}^m \mu_i H(h_i(X)) \right) = \begin{cases} 0, & X \in D, \\ \infty, & X \notin D. \end{cases}$$

$$\lim_{r \rightarrow \infty} \left(\alpha_g^r \sum_{j=1}^l \lambda_j G(g_j(X)) \right) = \begin{cases} 0, & X \in D, \\ \infty, & X \notin D. \end{cases}$$

В качестве функционалов $H(h_i(X))$, $G(g_j(X))$ можно взять расстояния в какой-либо метрике от точки 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 $H(h_i(X))$ обычно используют функционалы $H(h_i(X)) = h_i^2(X)$, в качестве функционалов $G(g_j(X))$ — функционалы $G(g_j(X)) = (g_j^+(X))^2$, где $g_j^+(X) = \begin{cases} 0, & g_j(X) \geq 0, \\ g_j(X), & g_j(X) < 0. \end{cases}$

В качестве критерия окончания итераций в методе последовательной безусловной оптимизации можно использовать неравенство

$$\left\| X^r - X^{r/2} \right\| \leq \varepsilon_x, \quad (4.1.8)$$

где r — четное число итераций, ε_x — требуемая точность решения по X .

Недостатком метода последовательной безусловной оптимизации является значительное усложнение структуры минимизируемой функции (см. рис. 4.1.1) — плата за исключение ограничений.

Схема метода штрафных функций.

1. Задаем начальную точку X^0 и полагаем счетчик числа итераций $r = 0$.

2. Исходя из точки X^r , одним из методов локальной безусловной оптимизации решаем задачу — находим точку X^{r+1} .
3. Проверяем условие окончания поиска (4.1.8). Если условие окончания поиска выполнено, то полагаем $X^* = X^{r+1}$ и завершаем итерации. Иначе — по некоторому правилу увеличиваем значения параметров α_h^r, α_g^r , полагаем $r = r + 1$ и переходим к п.3.

Примечание 4.1.1. В зависимости от метода локальной безусловной оптимизации, который используется для решения задач (4.1.3), метод последовательной безусловной оптимизации может быть детерминированным и случайным, нулевого, первого или второго порядка.

4.2. Метод скользящего допуска

Рассмотрим следующую многомерную задачу условной локальной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного во множестве D евклидова пространства R^n ,

$$\min_{X \in D \subset R^n} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (4.2.1)$$

где множество допустимых значений

$$D = \{X / g_j(X) \geq 0, j \in [1, \dots, l]\}. \quad (4.2.2)$$

Основы метода скользящего допуска.

Метод скользящего допуска существенно использует множество

$$D^r = \{X / T(X) \leq \Delta^r\}. \quad (4.2.3)$$

где неотрицательный скаляр Δ^r — **критерий скользящего допуска**, $T(X)$ — неотрицательно определенный функционал над множеством всех ограничивающих функций $g_j, j \in [1, \dots, l]$.

При этом функционал $T(X)$ должен быть сконструирован таким образом, чтобы $T(X) = 0$ при $X \in D$ и значение $T(X)$ возрастало по мере удаления точки X от границы области допустимых значений D . Критерий скользящего допуска Δ^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 X называется **допустимой точкой**, если $T(X) = 0$, **почти допустимой точкой** — если $0 < T(X) \leq \Delta^r$, **недопустимой точкой** — если $T(X) > \Delta^r$. Поскольку величина Δ^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 точки X^r , выполняем итерацию по решению задачи локальной безусловной оптимизации

$$\min_{X \in R^n} \Phi(X) = \Phi(\tilde{X}^{r+1}) \quad (4.2.4)$$

- находим точку \tilde{X}^{r+1} . Если $T(\tilde{X}^{r+1}) \leq \Delta^{r+1}$ (точка \tilde{X}^{r+1} является допустимой точкой или почти допустимой точкой), то полагаем $X^{r+1} = \tilde{X}^{r+1}$ и заканчиваем данную итерацию.

2. Если $T(\tilde{X}^{r+1}) > \Delta^{r+1}$ (точка \tilde{X}^{r+1} является недопустимой), то отыскиваем точку X^{r+1} , лежащую ближе к границе области D . Для этого с помощью того же базового метода, исходя из точки \tilde{X}^{r+1} , решаем задачу локальной безусловной оптимизации

$$\min_{X \in R^n} T(X) = T(X^{r+1}) \quad (4.2.5)$$

с условием окончания итераций

$$T(X^{r+1}) \leq \Delta^{r+1} \quad (4.2.6)$$

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 осцилляция 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 $(X_1^r, X_2^r, \dots, X_{n+1}^r)$:

$$\begin{cases} \Delta^r = \min \left\{ \Delta^{r-1}, \frac{1}{n+1} \sum_{i=1}^{n+1} |X_i^r - X_c^r| \right\} \\ \Delta^0 = \frac{1}{n+1} \sum_{i=1}^{n+1} |X_i^0 - X_c^0|. \end{cases} \quad (4.2.7)$$

Здесь X_c^r — вектор координат центра тяжести многогранника $(X_1^r, X_2^r, \dots, X_{n+1}^r)$, так что величина $\frac{1}{n+1} \sum_{i=1}^{n+1} |X_i^r - X_c^r|$ — есть среднее расстояние вершин многогранника $(X_1^r, X_2^r, \dots, X_{n+1}^r)$ от его центра тяжести.

Из (4.2.7) следует, что критерий скользящего допуска Δ^r является положительно определенной функцией координат вершин многогранника $\Delta^r = \Delta^r(X_1^r, X_2^r, \dots, X_{n+1}^r)$. С другой стороны, поскольку размер многогранника при приближении к точке минимума X^* уменьшается (в пределе до нуля), то справедливо предельное соотношение $\lim_{r \rightarrow \infty} \Delta^r = 0$.

Для задачи (4.2.1), (4.2.2) в качестве функционала $T(X)$ обычно используют функционал

$$T(X) = \sum_{j=1}^l (g_j^+(X))^2, \quad (4.2.8)$$

где $g_j^+(X) = \begin{cases} 0, & g_j(X) \geq 0, \\ g_j(X), & g_j(X) < 0. \end{cases}$

Из (4.2.8) следует, что функционал $T(X)$ обладает следующим

свойством: $\begin{cases} T(X) = 0, & X \in D, \\ T(X) > 0, & X \notin D. \end{cases}$

Из (4.2.8) вытекает также, что если значение функционала $T(X)$ мало, то точка X находится недалеко от границы области D .

Примечание 4.2.1. 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 X_i^r , $i \in [1, \dots, n+1]$ обозначим S^r .

1. Задаем начальный симплекс S^0 и полагаем счетчик числа итераций $r=0$.
2. С помощью метода Нелдера-Мида, исходя из симплекса S^r , выполняем одну итерацию по решению задачи локальной безусловной оптимизации (4.2.4) — находим симплекс \tilde{S}^{r+1} с вершинами \tilde{X}_i^r , $i \in [1, \dots, n+1]$.
3. Вычисляем значения функционала $T(X)$ во всех вершинах симплекса \tilde{S}_i^{r+1} и значение критерия скользящего допуска Δ^r . Находим вершину симплекса \tilde{S}^{r+1} , в которой значение функционала $T(X)$ максимально, т.е.

вершину, которая расположена дальше всех от границы области D . Обозначим эту вершину \tilde{X}_T^{r+1} .

4. Если $T(\tilde{X}_T^{r+1}) \leq \Delta^{r+1}$ (точка \tilde{X}_T^{r+1} 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 $X_i^{r+1} = \tilde{X}_i^{r+1}$, $i \in [1, \dots, n+1]$, полагаем $r = r + 1$ и переходим к п.2.

5. Если $T(\tilde{X}_T^{r+1}) > \Delta^{r+1}$ (точка \tilde{X}_T^{r+1} является недопустимой точкой), то с помощью метода Нелдера-Мида, исходя из точки \tilde{X}_T^{r+1} , решаем задачу локальной безусловной оптимизации (4.2.5) с критерием окончания итераций (4.2.6) — находим точку X_T^{r+1} . Формируем новый симплекс S^{r+1} с вершинами $X_i^{r+1} = \tilde{X}_i^{r+1}$, $i \in [1, \dots, n+1]$, $i \neq T$, 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 4.2.1. Рассмотрим двумерную задачу условной оптимизации (4.2.1), когда критерий оптимальности равен $\Phi(X) = -x_1 - x_2$ и множество допустимых значений D определяется ограничениями

$$\begin{cases} g_1(X) = -x_1^2 - x_2^2 + 9 \geq 0, \\ g_2(X) = -x_1 \geq 0, \\ g_3(X) = -x_2 \geq 0. \end{cases} \quad (4.2.9)$$

Рассмотренный пример иллюстрирует тот факт, что поскольку вершина X_1^r симплекса S^r расположена далеко от границы области D , то после операций отражения и растяжения точка \tilde{X}_1^{r+1} может оказаться глубоко в недопустимой области. В результате в процессе минимизации функционала $T(X)$ может получиться точка X_1^{r+1} , которая снова оказывается далеко от границы области D . И т.д.

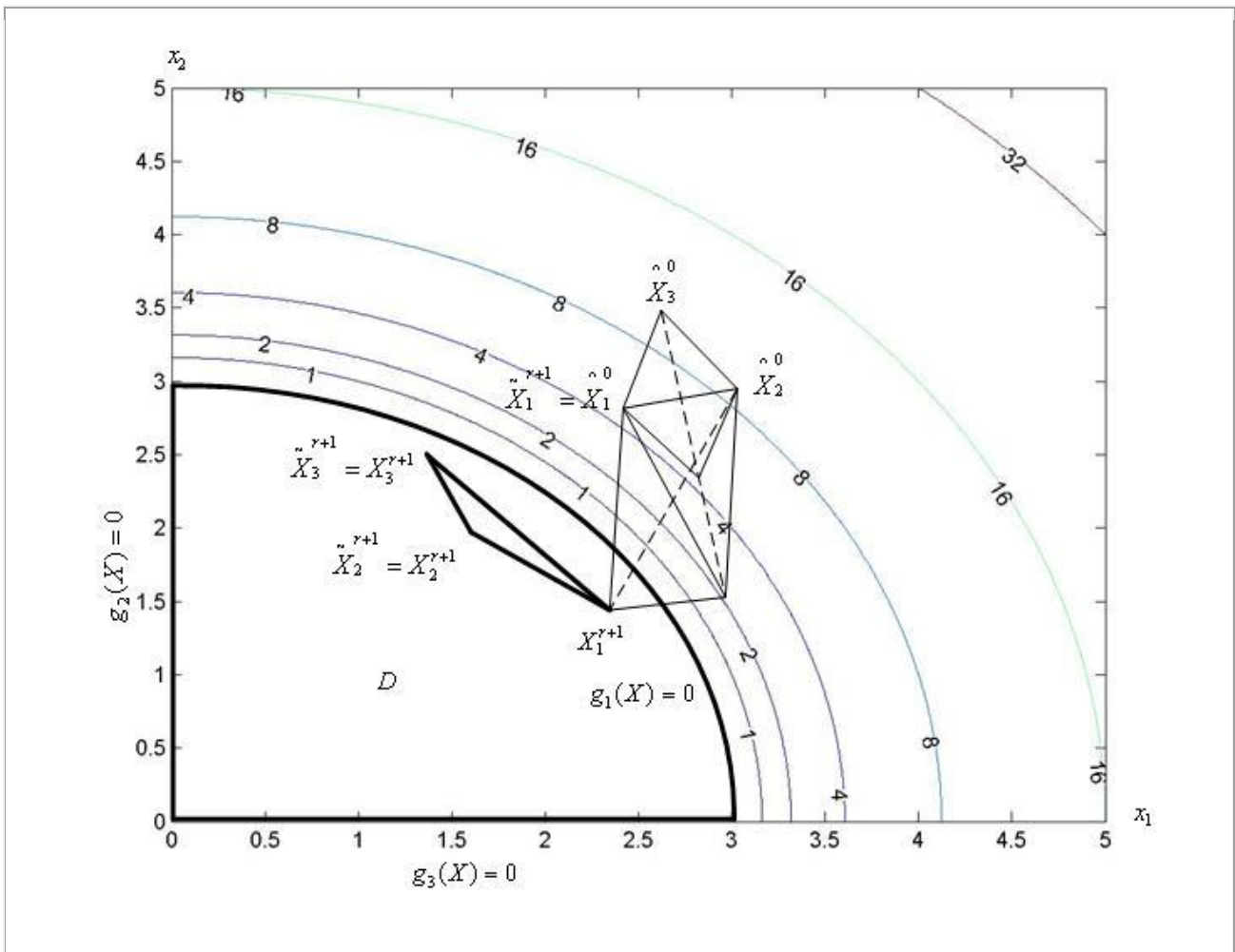


Рис. 4.2.2. К примеру 4.2.1. После успешного отражения вершины X_3^0 выполнено растяжение симплекса и отражение вершины X_2^0

Эффект, рассмотренный в прим. 4.2.1, и называется осцилляцией решения относительно границы области D .

Для ослабления влияния осцилляций в простейшем случае можно вместо точки \tilde{X}_T^{r+1} использовать точку A^r — середину отрезка $[X_T^r, \tilde{X}_T^{r+1}]$.

Чаще с этой целью используют квадратичную интерполяцию функции $T(X)$ на отрезке $[X_T^r, \tilde{X}_T^{r+1}]$ по трем точкам $(X_T^r, \Phi(X_T^r))$, $(A^r, \Phi(A^r))$, $(\tilde{X}_T^{r+1}, \Phi(\tilde{X}_T^{r+1}))$, где A^r — также середина отрезка $[X_T^r, \tilde{X}_T^{r+1}]$ — см. рис. 4.2.3. Обозначим эту интерполирующую функцию $y(x) = \alpha x^2 + \beta x + \gamma$.

Вместо точки \tilde{X}_T^{r+1} в этом случае можно использовать один из нулей функции $y(x)$ либо его приближенное значение, найденное, например, методом касательных.

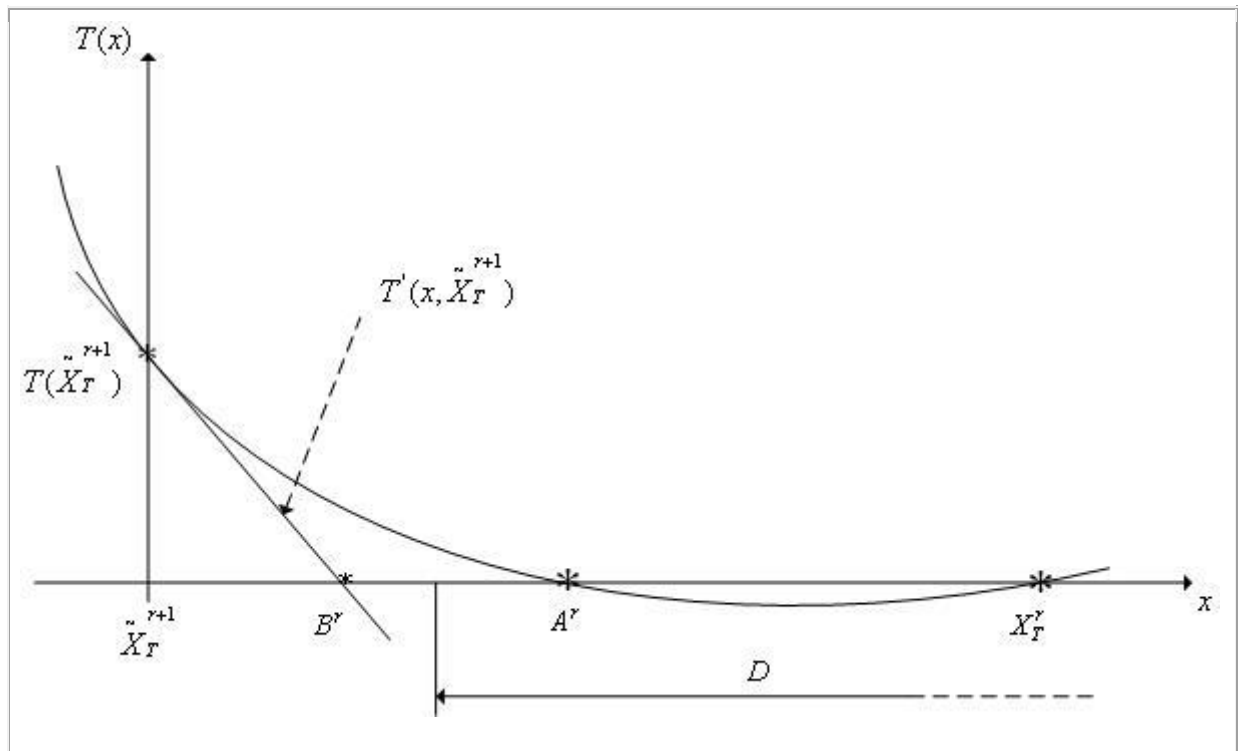


Рис. 4.2.3. Использование квадратичной интерполяции функции $T(X)$ на отрезке по трем точкам для ослабления осцилляций. Случай, когда точка A^r принадлежит области допустимых значений D

4.3. Модифицированный метод комплексов

Рассмотрим многомерную задачу локальной условной оптимизации

$$\min_{X \in D \subset R^n} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (4.3.1)$$

где множество допустимых значений определяется только ограничениями типа неравенств и представляет собой гиперпараллелепипед, т.е.

$$D = \left\{ X / x_i^U \geq x_i \geq x_i^L, i \in [1, \dots, n] \right\}. \quad (4.3.2)$$

Здесь $x_i^L, x_i^U, i \in [1, \dots, n]$ — нижняя и верхняя границы области допустимых значений D по i -му измерению (см. рис. 4.3.1).

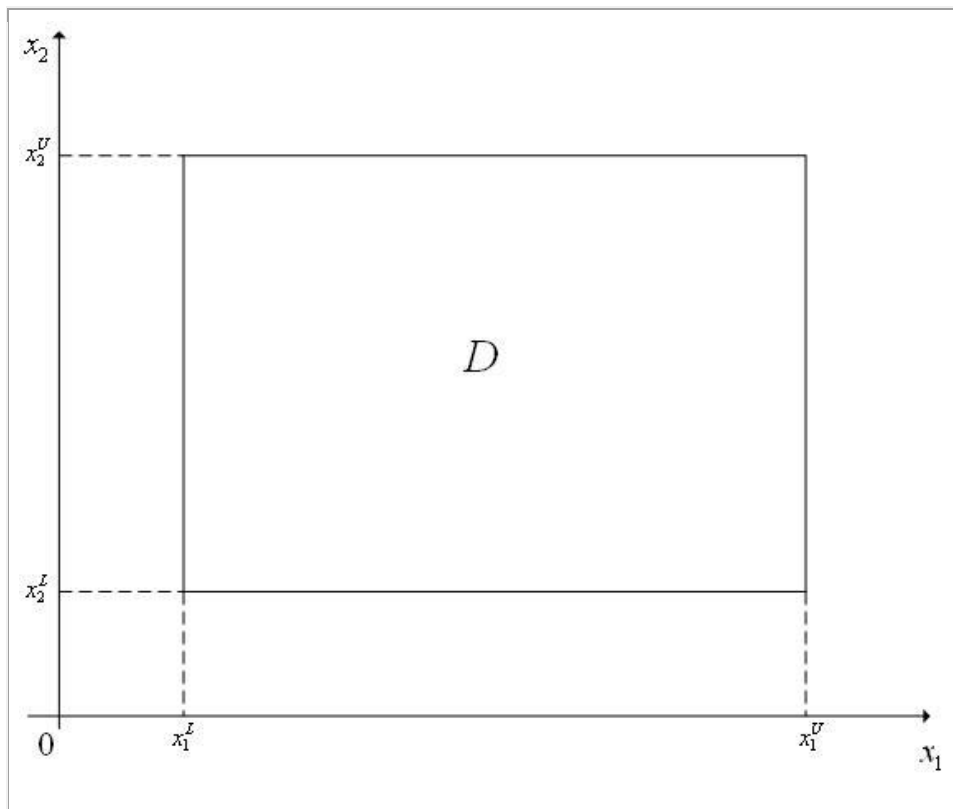


Рис. 4.3.1. Область допустимых значений D в виде гиперпараллелепипеда; $n = 2$

Метод комплексов в многомерной задаче безусловной оптимизации рассмотрен в параграфе 3.10. 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 (4.3.1) 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

$$X_i = X_0 + l \frac{\psi_i}{\|\psi_i\|}, \quad i \in [1, \dots, N], \quad (4.3.3)$$

где X_0 — произвольная начальная точка, i — номер вершины комплекса, $l > 0$ — скаляр, определяющий размер комплекса, ψ_i — реализация n -мерного случайного вектора, $\|\cdot\|$ — некоторая векторная норма. Обычно в качестве координат вектора ψ_i используют независимые случайные величины, равномерно распределенные в интервале $[-1, 1]$.

Отражение вершины комплекса с растяжением.

Положим, что задан комплекс C^r с N вершинами X_i^r , $i \in [1, \dots, N]$, и его вершину 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 X_k^r (см. рис. 4.3.2). Обозначим координаты вершин нового комплекса X_i^{r+1} , $i \in [1, \dots, N]$.

Тогда имеем

$$X_i^{r+1} = X_i^r, \quad i \in [1, \dots, N], \quad i \neq k; \quad X_k^{r+1} = X_c^r + \alpha(X_c^r - X_k^r), \quad (4.3.4)$$

где α — коэффициент растяжения (рекомендуемое значение — 1.3), X_c^r — вектор координат центра тяжести комплекса C^r :

$$X_c^r = \frac{1}{N} \sum_{i=1}^m X_i^r. \quad (4.3.5)$$

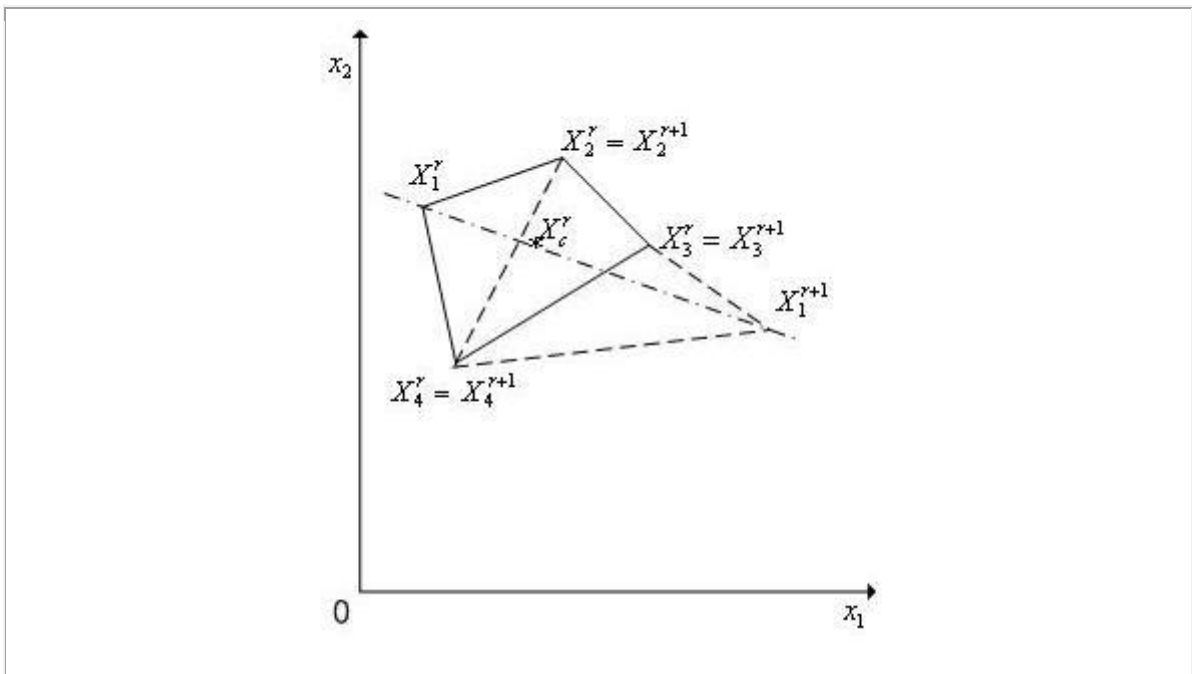


Рис. 4.3.2. Отражение вершины комплекса C^r через центр его тяжести с растяжением. Пунктиром показан новый комплекс C^{r+1}

Сжатие комплекса.

Положим, что задан комплекс C^r с N вершинами X_i^{r+1} , $i \in [1, \dots, N]$, и его 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 X_k^r (см.

рис. 4.3.3). Обозначим координаты вершин нового комплекса X_i^{r+1} , $i \in [1, \dots, N]$. Тогда имеем

$$X_i^{r+1} = X_i^r, \quad i \in [1, \dots, N], \quad i \neq k; \quad X_k^{r+1} = X_c^r + \beta(X_k^r - X_c^r), \quad (4.3.6)$$

где β — коэффициент растяжения (рекомендуемое значение — 2), X_c^r — вектор координат центра тяжести комплекса C^r (см. (4.3.5)).

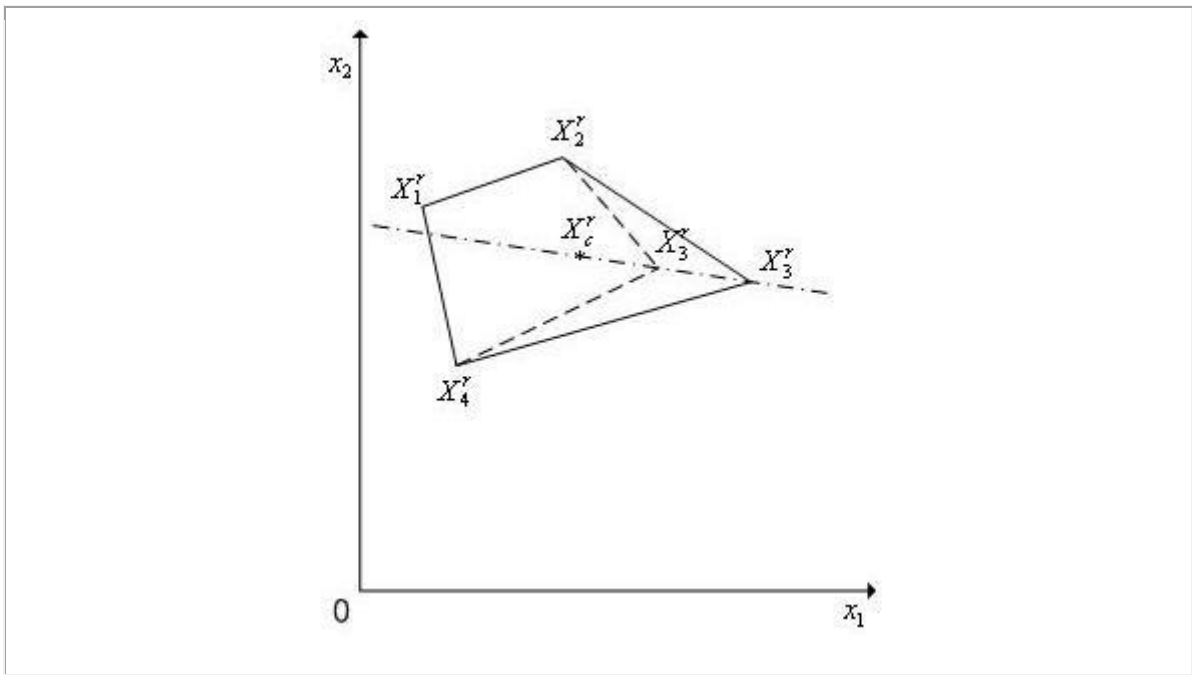


Рис. 4.3.3. Сжатие комплекса C^r . Пунктиром показан новый комплекс C^{r+1}

Упрощенная схема модифицированного метода комплексов.

1. Задаем начальную точку $X_0^0 \in D$, исходя из которой должен быть построен комплекс C^0 , величину l^0 , а также коэффициенты α, β ; полагаем счетчик числа итераций $r = 0$.
2. Строим начальный комплекс C^0 :
 - поочередно для $i = 1, 2, \dots, N$ по формуле (4.3.3) находим координаты вершин комплекса X_i^r комплекса C^r ;
 - если вершина X_p^r является недопустимой (выходит за границы области D), то по формуле, аналогичной формуле (4.3.6), выполняем сжатие уже построенного комплекса с p вершинами, вдоль направления $\left(X_p^r, (X_c^r)_p \right)$,

где $(X_c^r)_p$ — центр тяжести уже найденных $(p-1)$ -ой вершин комплекса (см. рис. 4.3.4);

- если после сжатия комплекса вершина X_p^r по-прежнему является недопустимой, повторяем описанную процедуру сжатия;

- вычисляем значения $\Phi(X_i^r)$ функции $\Phi(X)$ во всех вершинах построенного комплекса C^r .

3. Находим максимальное из значений функции $\Phi(X)$ в вершинах комплекса C^r : $\Phi(X_k^r) = \max_{i \in [1, \dots, N]} \Phi(X_i^r)$.

4. По формулам (4.3.4), (4.3.5) отражаем с растяжением вершину X_k^r комплекса C^r — получаем вершину и новый X_k^{r+1} комплекс C^{r+1} :

- если точка X_k^{r+1} является не допустимой (выходит за границы области D)

и $\Phi(X_i^{r+1}) > \Phi(X_i^r)$, то по формуле (4.3.6), выполняем сжатие комплекса

C^{r+1} вдоль направления (X_k^{r+1}, X_c^{r+1}) , где — X_c^{r+1} центр тяжести

комплекса C^{r+1} , до тех пор, пока точка X_k^{r+1} не станет допустимой (см.

рис. 4.3.5). Переходим к п.5;

- если точка X_k^{r+1} является допустимой (не выходит за границы области D)

и $\Phi(X_i^{r+1}) < \Phi(X_i^r)$, то переходим к шагу 5;

- если точка X_k^{r+1} является не допустимой, но $\Phi(X_i^{r+1}) < \Phi(X_i^r)$, то

переходим к п. 6.

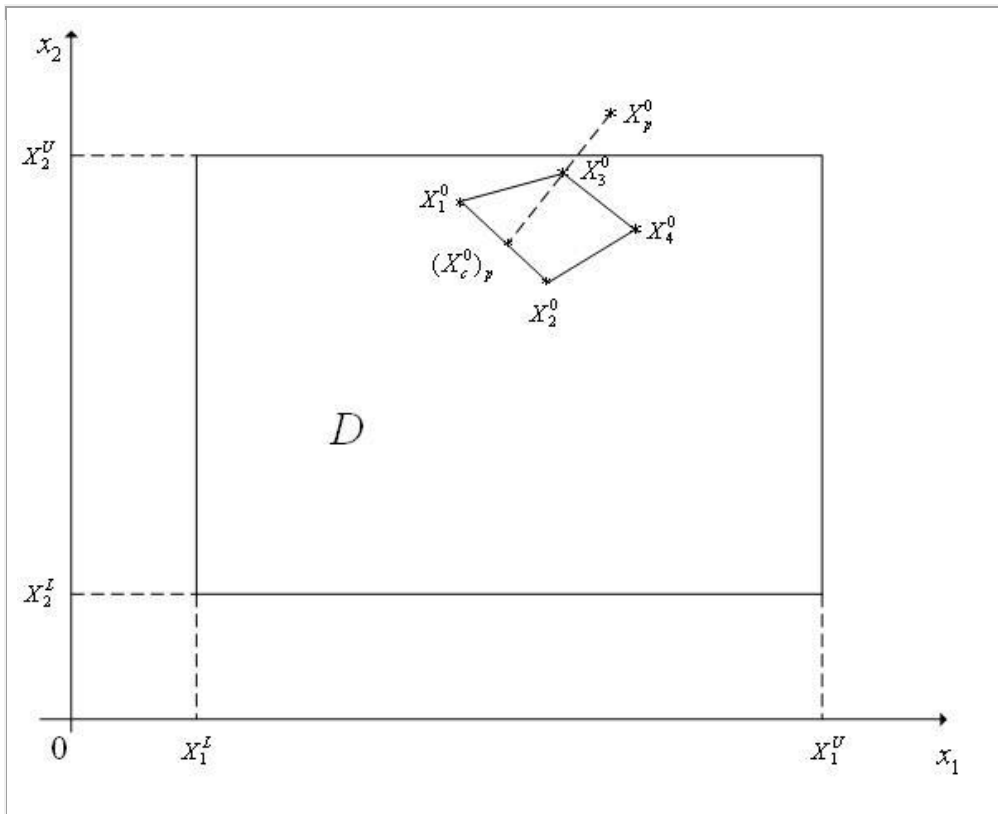


Рис. 4.3.4. Построение комплекса C^0

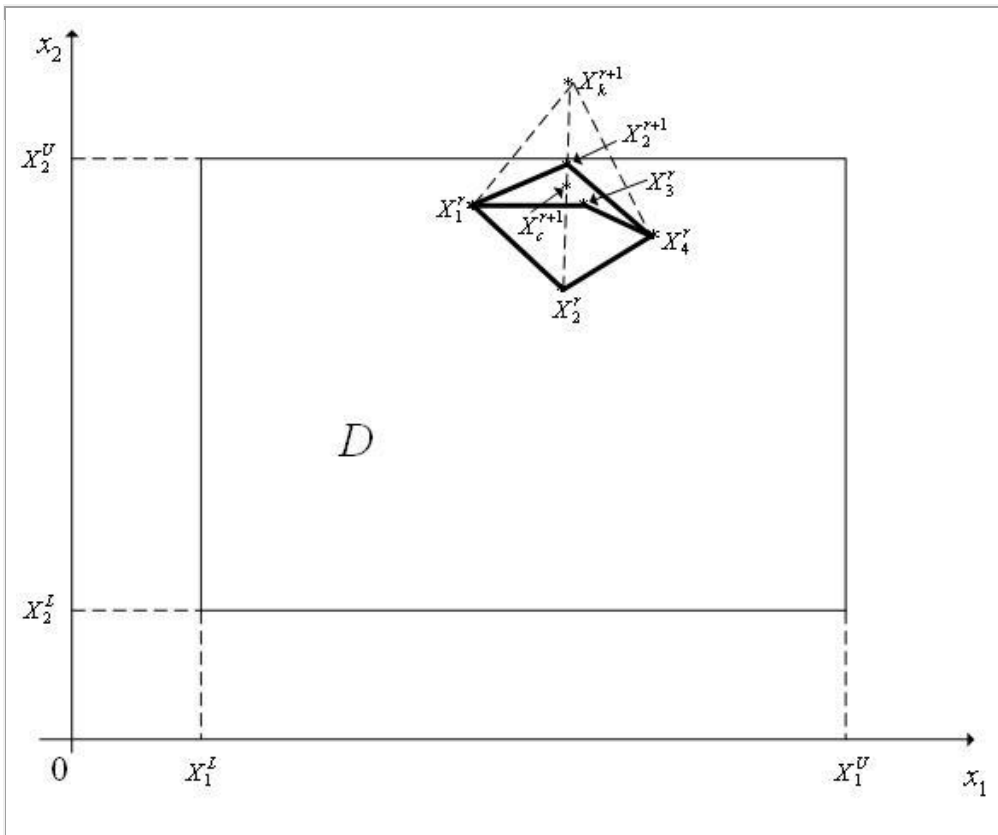


Рис. 4.3.5. Построение комплекса C^{r+1}

5. Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то в качестве точки X^* принимаем вершину комплекса C^{r+1} , к которой функция $\Phi(X)$ имеет наименьшее значение, вычисляем соответствующие значения $\Phi(X)$ и завершаем итерации. Иначе — переходим к п. 3.

6. Если $(X_k^{r+1})_i < X_i^L$, то полагаем $(X_k^{r+1})_i = X_i^L$; если $(X_k^{r+1})_i > X_i^U$ то полагаем $(X_k^{r+1})_i = X_i^U$ (см. рис. 4.3.6). Переходим к п.3.

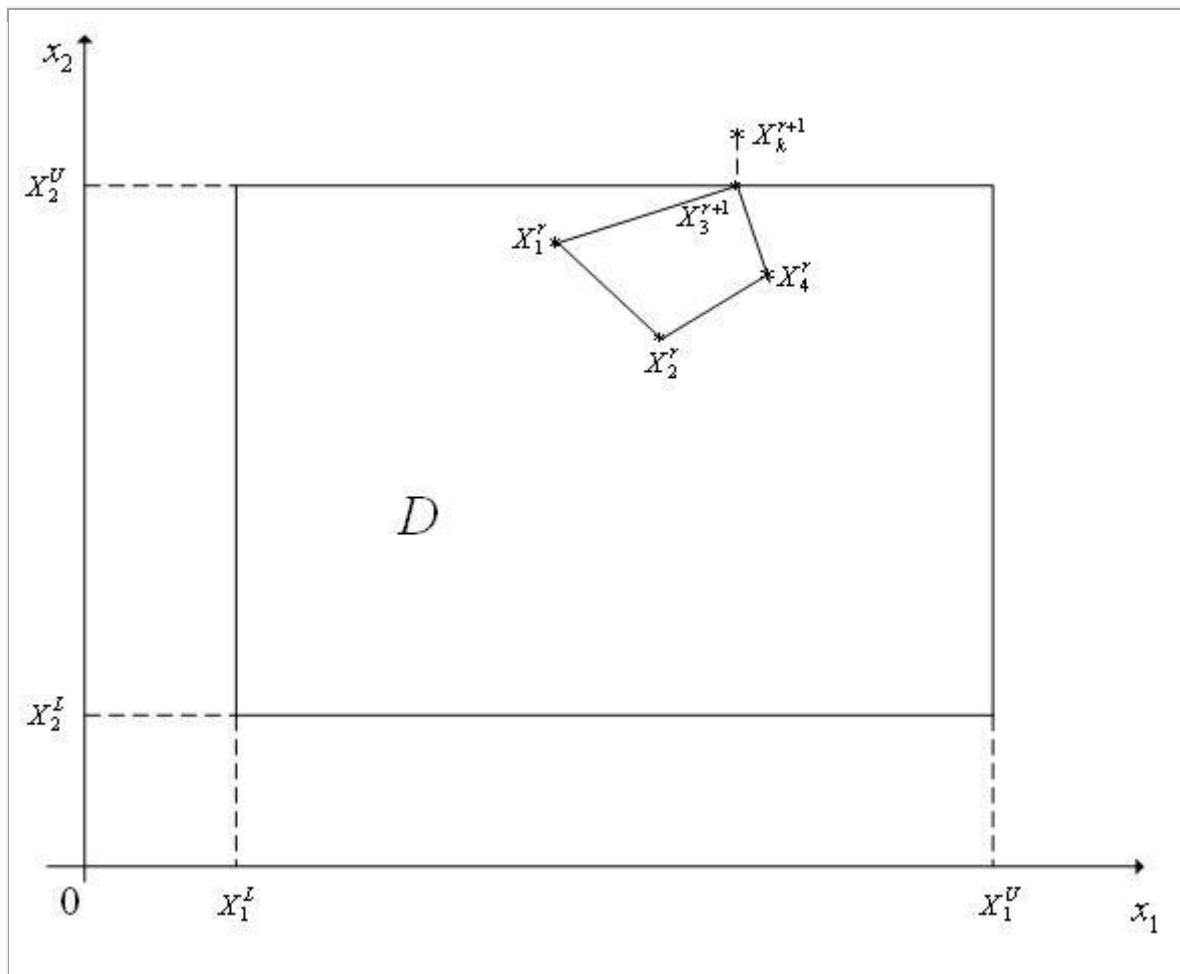


Рис. 4.3.6. Построение комплекса C^{r+1}

На рис. 4.3.4 точка X_0^P оказалась за границей области D . После операции сжатия комплекса вершины X_0^P комплекса вдоль направления

$\left(X_0^p, (X_c^o)_p \right)$ получаем вершину X_3^0 . Здесь $(X_c^o)_p$ — центр тяжести комплекса.

На рис. 4.3.5 полагается, что $\Phi(X_k^{r+1}) > \Phi(X_2^r)$. Точка X_k^{r+1} оказалась границей области D . После операции сжатия комплекса вершины X_k^{r+1} комплекса вдоль направления (X_k^{r+1}, X_c^{r+1}) получаем вершину X_2^{r+1} .

На рис. 4.3.6 полагается, что $\Phi(X_k^{r+1}) < \Phi(X_2^r)$. Точка X_k^{r+1} оказалась за границей области D - нарушено ограничение $X_k^{r+1} < X_2^U$. Точка X_3^{r+1} получена проектированием точки X_k^{r+1} на прямую $X_2 = X_2^U$.

В качестве критерия окончания поиска может использоваться следующее условие: максимальная длина ребра комплекса C^r не превышает ε_X — требуемую точность решения по X . Может использоваться также следующее аналогичное условие: максимальная разность значений функции $\Phi(X)$ в двух вершинах комплекса C^r не превышает ε_Φ — требуемую точность решения по Φ .

Могут использоваться также более сложные условия окончания поиска, учитывающие текущий размер комплекса или в некотором смысле среднее значение функции $\Phi(X)$ в его вершинах (см. параграф 3.10).

Изложенная схема метода комплексов приводит к "уплощению" комплекса вблизи границы области допустимых значений D , что может значительно уменьшить эффективность метода. С 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.

4.4. Метод линейной аппроксимации

Сделаем ряд дополнительных допущений. Пусть множество допустимых значений D определяется только ограничениями типа неравенств и ограничивающие функции $g_j(X), j \in [1, \dots, l]$ являются непрерывными, дифференцируемыми и выпуклыми:

$$\min_{X \in D \subset \mathbb{R}^n} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (4.4.1)$$

$$D = \left\{ X / g_j(X) \geq 0, j \in [1, \dots, l] \right\}. \quad (4.4.2)$$

Пусть функция $\Phi(X)$ также непрерывна, дифференцируема и выпукла во множестве D .

Суть метода линейной аппроксимации.

Метод линейной аппроксимации использует на каждой итерации линейную аппроксимацию целевой функции $\Phi(X)$ и ограничивающих функций $g_j(X), j \in [1, \dots, l]$ в окрестности текущей точки X^r

$$\tilde{\Phi}^r(X) = \Phi(X^r) + (\nabla \Phi(X^r), (X - X^r)), \quad (4.4.3)$$

$$\tilde{g}_j^r(X) = g_j(X^r) + (\nabla g_j(X^r), (X - X^r)), \quad j \in [1, \dots, l]. \quad (4.4.4)$$

Вместо задачи (4.4.1) на каждой итерации решается вспомогательная задача линейного программирования

$$\min_{X \in \tilde{D}^r} \tilde{\Phi}^r(X) = \tilde{\Phi}^r(X^{r+1}), \quad (4.4.5)$$

где $\tilde{D}^r = \left\{ X / \tilde{g}_j^r(X) \geq 0, j \in [1, \dots, l] \right\}$.

В изложенном виде метод может привести к выходу точки X^{r+1} за пределы допустимой области (см. прим. 4.4.1).

Пример 4.4.1. Рассмотрим следующую двумерную задачу условной оптимизации с тремя ограничениями типа неравенств (первое ограничение — нелинейное, второе и третье ограничения — линейные), где

$$\Phi(X) = x_1^2 + (x_2 - 6)^2 - 12, \quad g_1(X) = -x_1^2 - x_2^2 + 25, \quad g_2(X) = x_1, \quad g_3(X) = x_2.$$

Положим, что текущая точка есть $X^r = (2, 4)$. Линеаризуем целевую функцию $\Phi(X)$ и ограничивающую функцию $g_1(X)$ в окрестности этой

точки. Поскольку $\Phi(X^r) = 2^2 + (4 - 6)^2 - 12 = -4$, $\nabla\Phi(X) = \begin{pmatrix} 2x_1 \\ 2(x_2 - 6) \end{pmatrix}$,

$\nabla\Phi(X^r) = \begin{pmatrix} 4 \\ -4 \end{pmatrix}$ по формуле (4.4.3) имеем

$$\tilde{\Phi}^r(X) = -4 + \left(\begin{pmatrix} -4 \\ -8 \end{pmatrix}, \begin{pmatrix} x_1 - 2 \\ x_2 - 4 \end{pmatrix} \right) = -4x_1 - 4x_2 + 4.$$

Аналогично для ограничивающих функции по формуле (4.4.4) имеем:

$$\tilde{g}_1^r(X) = 5 + \left(\begin{pmatrix} -4 \\ -8 \end{pmatrix}, \begin{pmatrix} x_1 - 2 \\ x_2 - 4 \end{pmatrix} \right) = -4x_1 - 8x_2 + 45; \quad \tilde{g}_2^r(X) = g_2(X) = x_1;$$

$$\tilde{g}_3^r(X) = g_3(X) = x_2.$$

Пример 4.4.1 иллюстрирует рис. 4.4.1.

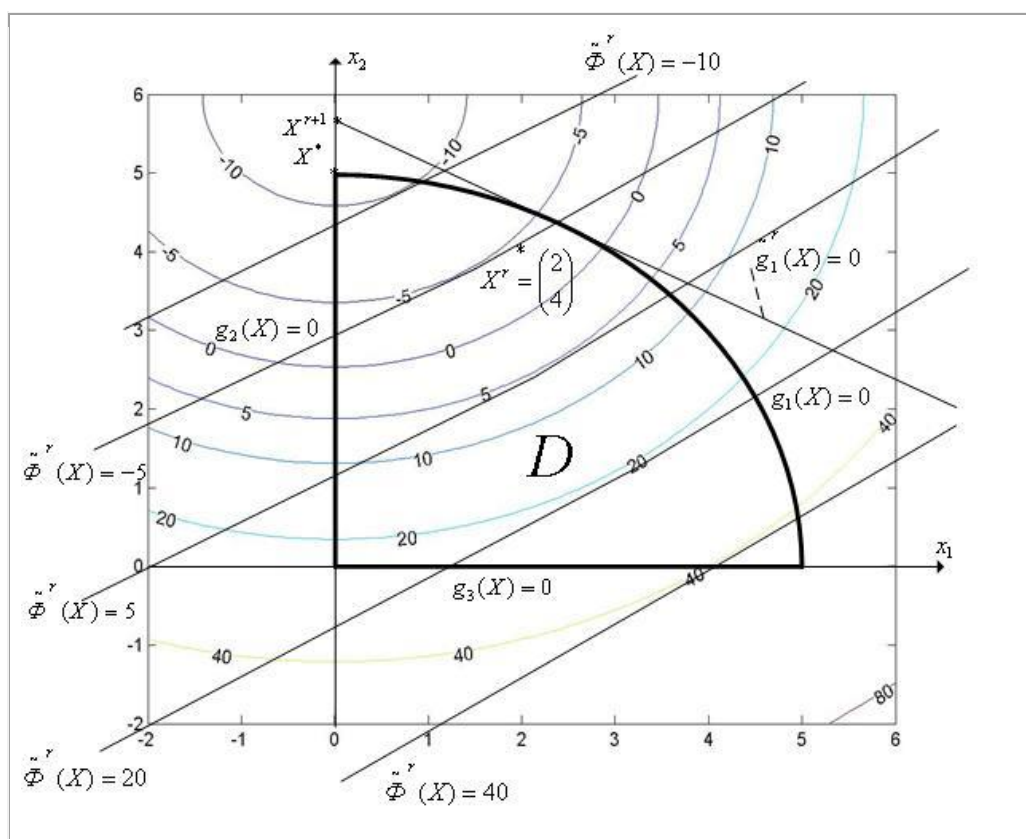


Рис. 4.4.1. Точка X^{r+1} лежит вне области допустимых значений D

Примечание 4.4.1. Прямая \tilde{g}_1^r представляет собой след от пересечения плоскости, которая является касательной к поверхности $z = g_1(X)$ в точке X^r , с плоскостью Ox_1x_2 . Эта прямая не обязательно является касательной к линии $g_1(X) = 0$ - прямая $\tilde{g}_1^r(X) = 0$ может пересекать кривую $g_1(X) = 0$, быть касательной к ней или не иметь с ней общих точек. Аналогично, линия уровня $\Phi^r(X) = c$ функции $\Phi(X)$ представляет собой след от пересечения плоскости, которая является касательной к поверхности $z = \Phi(X)$ в точке X^r , с плоскостью $z = c$.

Чтобы избежать выхода текущей точки за границы области допустимых значений, следующее приближение X^{r+1} к точке минимума X^* функции $\Phi(X)$ из множества D находится по формуле

$$X^{r+1} = X^r + \lambda^r (\tilde{X}^r - X^r), \quad (4.4.6)$$

где \tilde{X}^r — решение вспомогательной задачи линейного программирования (4.4.5).

Величина шага $\lambda^r \in [0, 1]$ в формуле (4.4.6) в разных вариантах метода линейной аппроксимации может определяться разными способами.

Приведем два из множества возможных способов.

1-й способ выбора величины шага λ^r . Величина λ^r находится как решение задачи одномерной оптимизации функции $\Phi(X^r + \lambda(\tilde{X}^r - X^r))$ на отрезке $[0, 1]$:

$$\Phi(X^{r+1}) = \Phi(X^r + \lambda^r(\tilde{X}^r - X^r)) = \min_{\lambda^r \in [0, 1]} \Phi(X^r + \lambda(\tilde{X}^r - X^r)). \quad (4.4.7)$$

2-й способ выбора величины шага λ^r . Полагаем $\lambda^r = 1$ и по формуле (4.4.6) находим вектор X^{r+1} . Вычисляем значение $\Phi(X^{r+1})$ целевой функции в полученной точке. Если условие

$$\Phi(X^{r+1}) < \Phi(X^r) \quad (4.4.8)$$

не выполнено, то уменьшаем величину шага λ^r (например, в два раза) и повторно проверяем выполнение условия (4.4.8). Дробление шага λ^r и вычисление X^{r+1} производим до выполнения условия (4.4.8).

Схема метода линейной аппроксимации.

Рассмотрим вариант метода, в котором используется 1-й способ выбора величины шага λ^r .

1. Задаем начальную точку $X^0 \in D$ и полагаем счетчик числа итераций $r=0$.
2. Вычисляем градиенты функций $\Phi(X)$, $g_j(X)$, $j \in [1, \dots, l]$ в точке X^r .
3. Решаем задачу линейного программирования (4.4.5) – находим точку \tilde{X}^r .
4. Решаем одномерную задачу минимизации (4.4.7) – находим величину шага λ^r и вектор X^{r+1} .
5. Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем $X^* \approx X^{r+1}$, $\Phi^* \approx \Phi(X^{r+1})$ и завершаем итерации. Иначе – полагаем $r = r + 1$ и переходим к п.2.

В качестве критерия окончания поиска можно использовать стандартные условия окончания итераций

$$\begin{aligned} \|X^{r+1} - X^r\| &= \lambda^r \leq \varepsilon_X, \\ |\Phi(X^{r+1}) - \Phi(X^r)| &\leq \varepsilon_\Phi \end{aligned}$$

или условие

$$\|\nabla \Phi(X^{r+1})\| \leq \varepsilon_\nabla,$$

где ε_{∇} — константа, определяющая требуемую точность решения по градиенту функции $\Phi(X)$.

Отметим следующие трудности, возникающие при использовании метода линейной аппроксимации:

1. Если функция $\Phi(X)$ имеет высокую степень нелинейности, то на основе решения вспомогательной задачи минимизации (4.4.5) направление поиска может быть выбрано слишком неточно (см. рис. 4.4.2), что приводит к медленной сходимости метода.
2. Метод требует, чтобы точка X^0 принадлежала множеству допустимых значений D . Если это требование не выполнено, то прежде придется использовать какой-либо метод поиска точки, принадлежащей множеству допустимых значений.

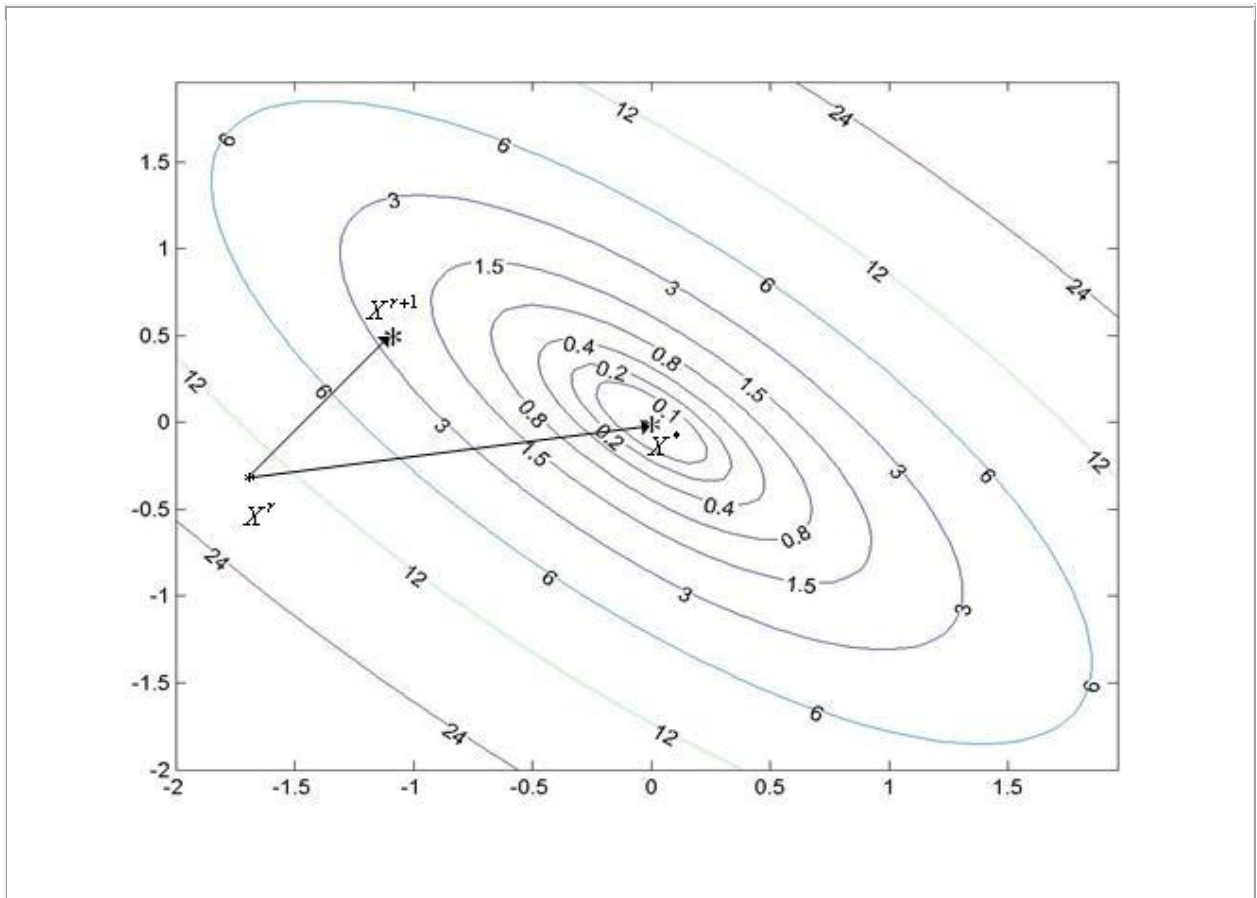


Рис. 4.4.2. Направление поиска ($X^{r+1} - X^r$), которое обеспечивает метод на основе линейной аппроксимации, далеко от оптимального направления ($X^* - X^r$)

Возможны модификации метода линейной аппроксимации, при которых необходимые производные вычисляются с помощью конечных разностей.

4.5. Метод проекции градиента

Рассмотрим многомерную задачу локальной условной оптимизации

$$\min_{X \in D \subset R^n} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (4.5.1)$$

где множество допустимых значений определяется только ограничениями типа неравенств

$$D = \left\{ X / g_j(X) \geq 0, j \in [1, \dots, l] \right\}. \quad (4.5.2)$$

и целевая функция $\Phi(X)$ и ограничивающие функции $g_j(X), j \in [1, \dots, l]$ 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

$$X^{r+1} = X^r + \lambda^r S^r, \quad (4.5.3)$$

полученная с помощью градиентного метода наискорейшего спуска (см. параграф 3.7), 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 (4.5.3) λ^r - длина шага на r -ой итерации в направлении S^r ;

$$S^r = - \frac{\nabla \Phi^r}{\|\nabla \Phi^r\|}$$

- единичный вектор направления антиградиента функции $\Phi(X)$ в точке X^r , $\|\ast\|$ - некоторая векторная норма, например, евклидова.

Определение 4.5.1. Проекцией точки $X \in R^n$ на замкнутое множество $D \subset R^n$ называется ближайшая к X точка \hat{X} множества D . Т.е. точка $\hat{X} \in D$ называется проекцией точки $X \in R^n$ на замкнутое множество $D \subset R^n$, если

$$\rho(X, \hat{X}) = \min_{Y \in D} \rho(X, Y), \quad (4.5.4)$$

где $\rho(A, B)$ - расстояние между точками $A \in R^n$, $B \in R^n$ в некоторой метрике, например, $\rho(A, B) = \|A - B\|$.

Проекцию \hat{X} точки X на замкнутое множество D будем обозначать $P_D(X) = \hat{X}$ (см. рис. 4.5.1). Очевидно, что $P_D(X) = X$, если $X \in D$.

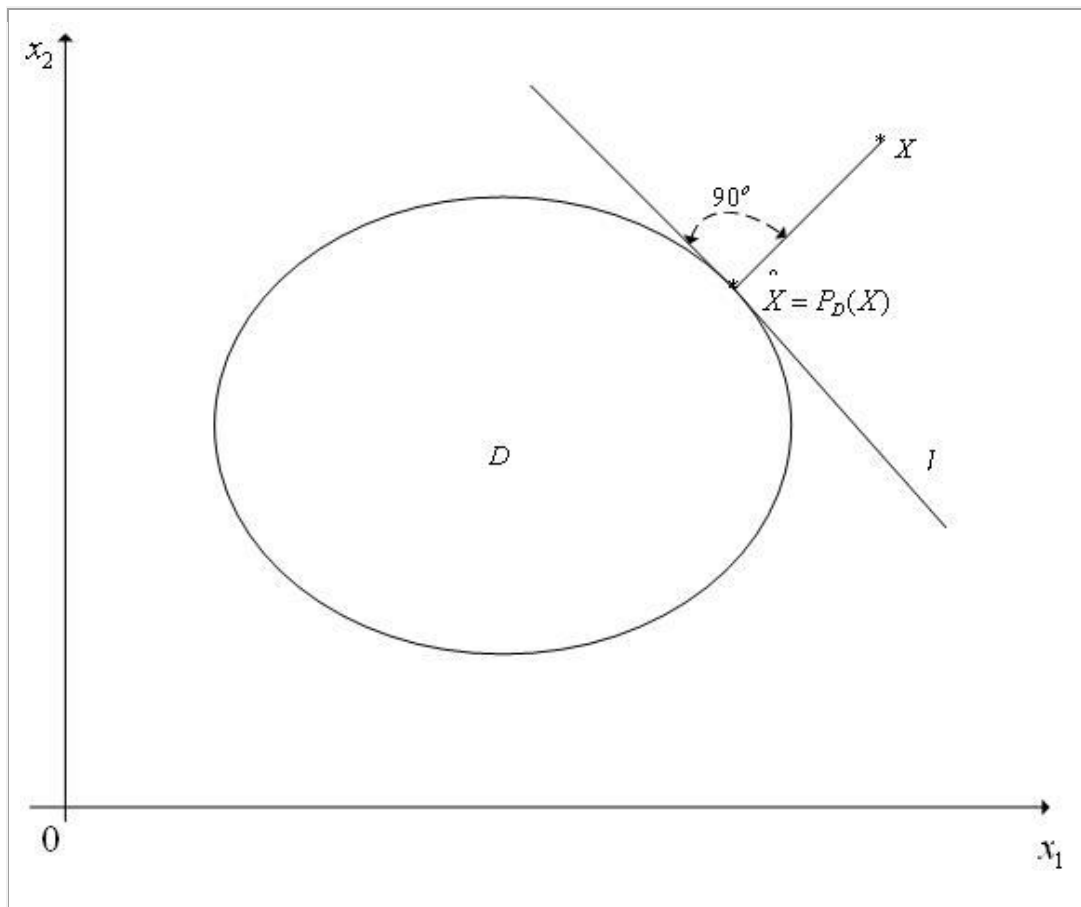


Рис. 4.5.1. К определению проекции точки на множество. Прямая l является касательной к границе области D в точке $P_D(X) = \hat{X}$

Можно показать, что если D — замкнутое выпуклое множество пространства R^n , то для любой точки $X \in R^n$ существует единственная ее проекция на это множество.

Задача (4.5.4) поиска проекции точки на множество также является многомерной задачей условной оптимизации и ее решение может вызвать в общем случае значительные затруднения.

Задача (4.5.4) становится задачей квадратичного программирования, если множество D задается лишь линейными ограничениями типа неравенств и если функция $\rho(X, Y)$ является квадратичной функцией Y , например, если $\rho(X, Y) = \|X - Y\|$.

Наибольший практический интерес представляет ситуация, когда множество D таково, что задача (4.5.4) 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. параграф 3.7). Рассмотрим комбинацию метода проекции градиента с градиентным методом дробления шага.

Напомним, что в градиентном методе с дроблением шага величина шага λ^r находится из условия

$$\Phi(X^r) - \Phi(X^{r+1}) < 0.5\lambda^r \|\nabla\Phi^r\|. \quad (4.5.5)$$

Схема метода.

1. Задаем начальную точку X^0 , начальную величину шага λ^0 и коэффициент дробления шага $\nu \in (0, 1)$. Полагаем счетчик числа итераций $r = 0$.
2. По формуле (4.5.3) вычисляем координаты точки \tilde{X}^{r+1} и проекцию $P_D(\tilde{X}^{r+1}) = X^{r+1}$ этой точки на множество D .
3. Вычисляем величину $\Phi(X^{r+1})$ - значение функции $\Phi(X)$ в точке X^{r+1} .

4. Если условие дробления шага выполнено (см. параграф 3.7), то переходим к следующему пункту. Иначе – переходим к п.6.
5. Полагаем $\lambda^r = \nu\lambda^r$ и переходим к п.2.
6. Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем $X^* \approx X^{r+1}$, $\Phi^* \approx \Phi(X^{r+1})$ и завершаем итерации. Иначе – полагаем $r = r + 1$ и переходим к п.2.

В качестве критерия окончания поиска можно использоваться одно из стандартных условий окончания итераций

$$\|X^{r+1} - X^r\| = \lambda^r \leq \varepsilon_X,$$

$$|\Phi(X^{r+1}) - \Phi(X^r)| \leq \varepsilon_\Phi$$

или условие

$$\|\nabla \Phi^r\| \leq \varepsilon_{\nabla},$$

где ε_{∇} - константа, определяющая требуемую точность решения по градиенту функции $\Phi(X)$.

Комбинацию метода проекции градиента и градиентного метода с дроблением шага иллюстрирует рис. 4.5.2, на котором показан фрагмент линий уровня функции Химмельблау.

Известны модификации метода проекции градиента, ориентированные на решение задач условной оптимизации с ограничениями типа равенств.

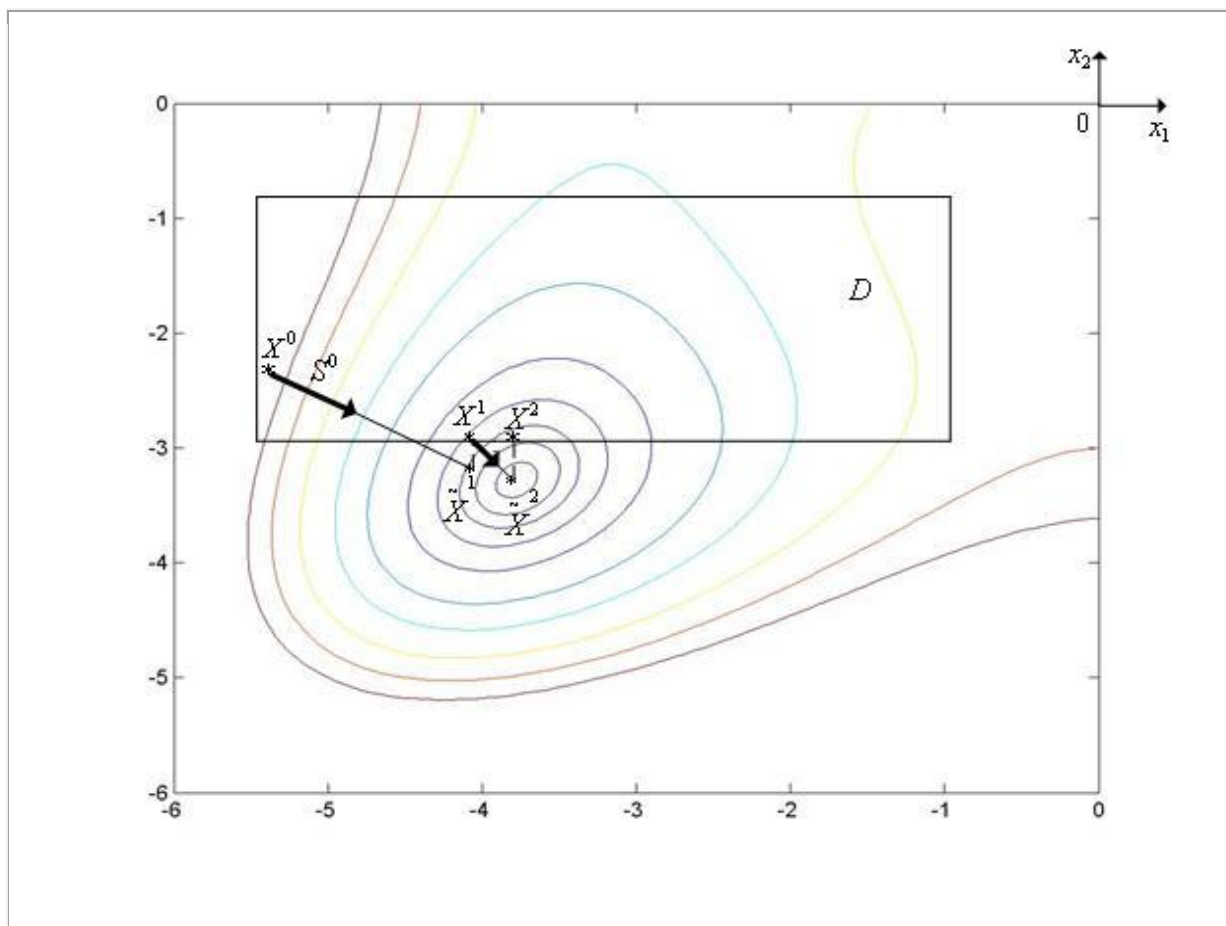


Рис. 4.5.2. Траектория поиска минимума функции Химмельблау комбинацией метода проекции градиента и градиентного метода с дроблением шага

4.6. Метод сведения к совокупности вложенных задач глобальной одномерной минимизации

Рассматривается следующая многомерная задача глобальной условной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного во множестве D пространства R^n ,

$$\min_{X \in D \subset R^n} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (4.6.1)$$

где множество допустимых значений

$$D = \left\{ X / h_i(X) = 0, g_j(X) \geq 0, i \in [1, \dots, m], j \in [1, \dots, l] \right\}. \quad (4.6.2)$$

Положим, что множество допустимых значений D задается только с помощью ограничений типа неравенств и представляет собой гиперпараллелепипед

$$D = \left\{ X / x_i^U \geq x_i \geq x_i^L, i \in [1, \dots, n] \right\}. \quad (4.6.3)$$

Метод сведения к совокупности вложенных одномерных задач глобальной оптимизации состоит в решении вместо задачи (4.6.1), (4.6.3) следующей совокупности вложенных одномерных задач условной оптимизации:

$$\min_{X \in D} \Phi(X) = \min_{X \in [x_1^L, x_1^U]} \min_{X \in D(x_1)} \dots \min_{X \in D(x_1, \dots, x_{n-1})} \Phi(x_1, x_2, \dots, x_n), \quad (4.6.4)$$

где множества $D(x_1), D(x_1, x_2), \dots, D(x_1, x_2, \dots, x_{n-1})$, представляют собой соответствующие сечения множества D (см. ниже).

Поясним смысл метода с помощью примера.

Пример 4.6.1. Положим, что $\Phi(X) = \Phi(x_1, x_2)$ и $D = \left\{ X / x_1^L \leq x_1 \leq x_1^U, x_2^L \leq x_2 \leq x_2^U \right\}$, т.е. $n = 2$. Вложенные одномерные задачи глобальной оптимизации (4.6.4) в этом случае можно представить в виде (см. рис. 4.6.1)

$$\min_{X \in D} \Phi(X) = \min_{x_1 \in [x_1^L, x_1^U]} \Phi(x_1), \quad (4.6.5)$$

$$\Phi(x_1) = \min_{x_2 \in D(x_1)} \Phi(x_1, x_2), \quad (4.6.6)$$

где $D(x_1)$ — сечение области D прямой, параллельной оси Ox . Задача (4.6.5) 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 $x_2 \in D(x_1)$

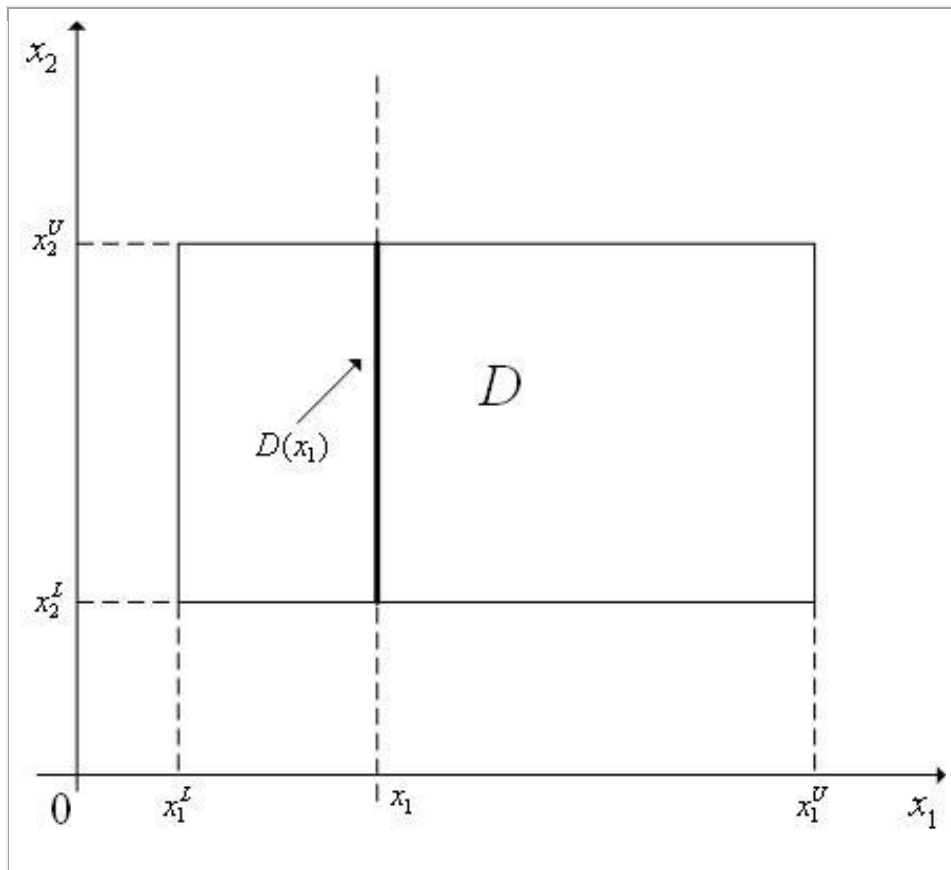


Рис. 4.6.1. К примеру 4.6.1. При решении задачи (4.6.5) вычисление значения критерия оптимальности $\Phi(X)$ при некотором $x = x_1$ требует решения задачи минимизации (4.6.6) на множестве $D(x_1)$

Положим, что для решения всех вложенных одномерных задач глобальной оптимизации (4.6.4) 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 (4.6.1), (4.6.3) 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 параграфах 2.7-2.10 методами решения одномерных задач глобальной оптимизации. Рассмотрим комбинацию этого метода с методом случайного поиска для двумерной задачи (4.6.1), (4.6.3).

Схема комбинации метода с методом случайного поиска ($n = 2$).

1. Задаем величины N_1, N_2 – количества испытаний при решении задач (4.6.5), (4.6.6), соответственно. Полагаем $r = 1$.
2. Генерируем с помощью какого-либо программного генератора случайных чисел, равномерно распределенных в интервале $[x_1^L, x_1^U]$, случайное число $x_1^0 = x_1^*$.
3. Методом случайного поиска решаем задачу (4.6.6) при $x_1 = x_1^*$ – находим точку $x_2^0 = x_2^*$ и вычисляем значение критерия оптимальности $\Phi^* = \Phi(x_1^*, x_2^*)$.

4. Аналогично п.2 генерируем случайное число x_1^r .
5. Методом случайного поиска решаем задачу (4.6.6) при $x_1 = x_1^r$ – находим точку \tilde{x}_2^r и вычисляем значение критерия оптимальности $\Phi^r = \Phi(x_1^r, \tilde{x}_2^r)$.
6. Если $\Phi^r < \Phi^*$, то выполняем присваивания $x_1^* = x_1^r$, $x_2^* = \tilde{x}_2^r$, $\Phi^* = \Phi^r$.
7. 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 $\Phi(X)$ в области D или каким-либо из рассмотренных ранее методов организуем в окрестности указанной точки поиск локального минимума функции $\Phi(X)$ 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. 4.6.2, 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 $x_1 = x_1^r$ случайным образом сгенерированы на r -ой итерации.

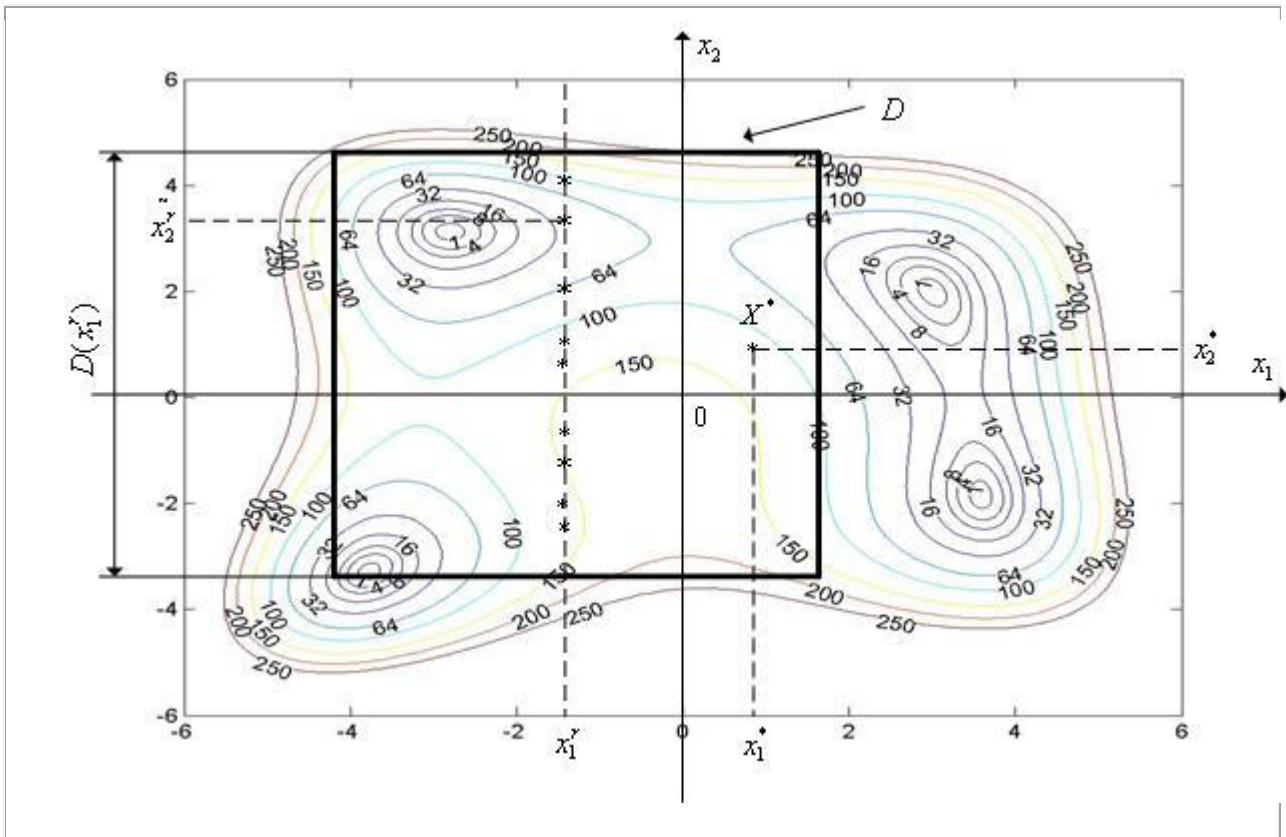


Рис. 4.6.2. Итерация номер r комбинации метода сведения с методом случайного поиска для двумерной задачи

4.7. Метод сведения к задаче одномерной глобальной оптимизации с помощью развертки Пеано

Рассмотрим многомерную задачу глобальной условной оптимизации

$$\min_{X \in D \subset \mathbb{R}^n} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (4.7.1)$$

где множество допустимых значений D задается только с помощью ограничений типа неравенств и представляет собой гиперкуб с длиной ребра, равной 1,

$$D = \{X / -0.5 \leq x_i \leq 0.5, i \in [1, \dots, n]\}. \quad (4.7.2)$$

Отметим, что произвольный гиперпараллелепипед с помощью линейного преобразования может быть сведен к гиперкубу (4.7.2), так что рассмотрение в качестве множества D гиперкуба (4.7.2), 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 первого разбиения с длиной ребра, равной $\frac{1}{2} = \left(\frac{1}{2}\right)^{s=2}$

(см. рис. 4.7.1(a)). 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 Π_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, на 2^n гиперкубов второго разбиения с длиной ребра, равной $\frac{1}{4} = \left(\frac{1}{2}\right)^{s=2}$ (см. рис. 4.7.1(б)). 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 Π_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 разбиения с длиной ребра, равной $\left(\frac{1}{2}\right)^s$, нумеруем их с помощью переменной $z_s \in [0, 2^n - 1]$, соединяем центры гиперкубов ломаной Π_s в порядке введенной нумерации и обозначаем $D(z_1, z_2, \dots, z_s)$.

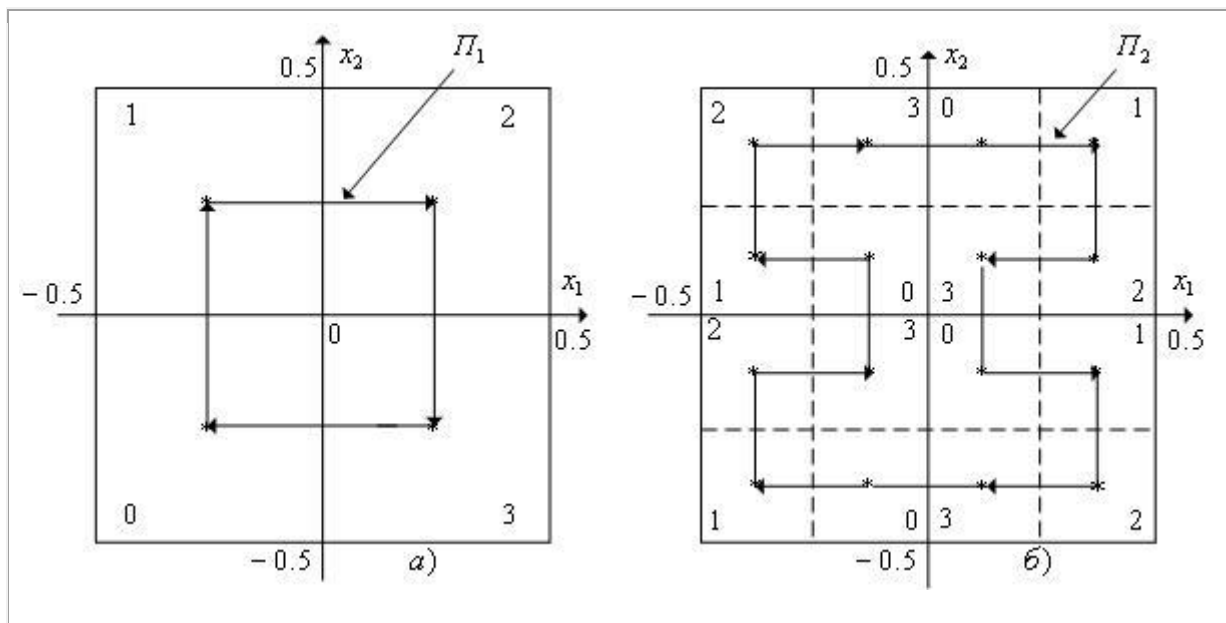


Рис. 4.7.1. К разбиению гиперкуба ($n=2$). а) Первое разбиение. б) Второе разбиение. Стрелками показано направление нумерации гиперкубов

Ломаная Π_s называется **разверткой Пеано**. В пределе при $s \rightarrow \infty$ ломаная Π_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. 4.7.2(а)), пронумеруем их слева направо с помощью переменной $z_1 \in [0, 2^n - 1]$ и обозначим Δz_1 .

Шаг 2 ($s=2$). Каждый из отрезков Δz_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. 4.7.2(б)), 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 $\Delta(z_1, z_2, \dots, z_n)$.

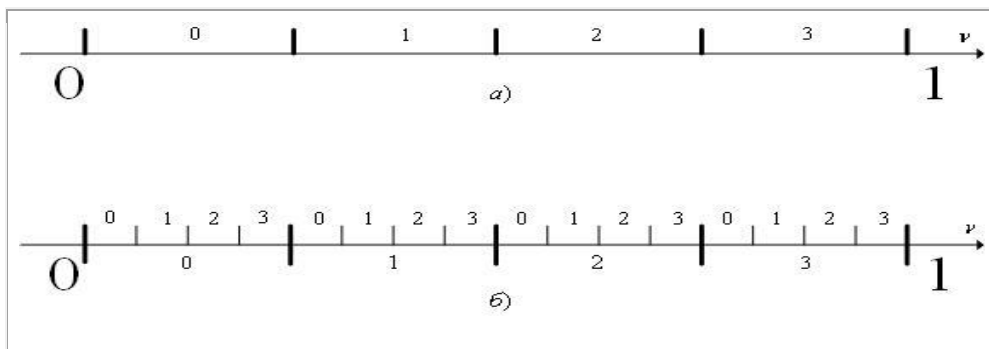


Рис. 4.7.2. К разбиению отрезка $[0, 1]$. а) Первое разбиение. б) Второе разбиение

Отображение отрезка $[0,1]$ на гиперкуб.

Определим отображение точки ν отрезка $[0,1]$ на гиперкуб D следующим образом: если точка $\nu \in \Delta(z_1, z_2, \dots, z_s)$, 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 $P(\nu)$. Таким образом, если $\nu \in \Delta(z_1, z_2, \dots, z_s)$, то $P(\nu) = X_{D(z_1, z_2, \dots, z_s)} \in D(z_1, z_2, \dots, z_s)$ (см. рис. 4.7.3).

На рис. 4.7.3 любая точка $\nu_1 \in \Delta(0,2)$ отображается в центр гиперкуба $D(0,2)$ - точку $X_{D(0,2)}$. Аналогично, любая точка $\nu_2 \in \Delta(1,3)$ отображается в точку $X_{D(1,3)}$ и любая точка $\nu_3 \in \Delta(2,1)$ 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.

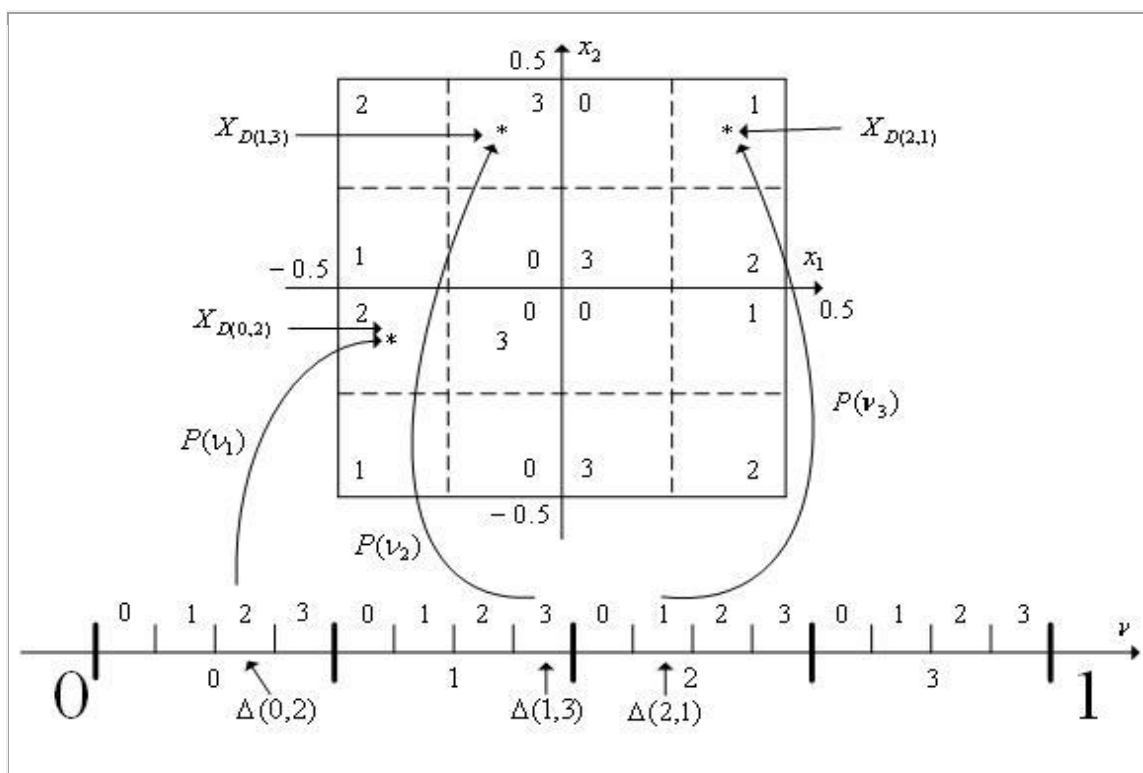


Рис. 4.7.3. К отображению отрезка $[0,1]$ на гиперкуб

Пусть $0, \alpha_1 \alpha_2 \dots \alpha_k \dots$ - двоичное представление числа ν , т.е.
 $\nu_b = 0, \alpha_1 \alpha_2 \dots \alpha_k \dots$

Утверждение 4.7.1. Если $\nu \in \Delta(z_1, z_2, \dots, z_s)$, то первые sn двоичных цифр числа $\nu_b = 0, \alpha_1 \alpha_2 \dots \alpha_k \dots$ 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 $\langle * \rangle$ - операция преобразования двоичного числа в десятичное.

Пример 4.7.1. Пусть область D представляет собой квадрат ($n = 2$).

На отрезке $[0, 1]$ рассмотрим точки $\nu_1 = 0.26$, $\nu_2 = 0.49$, $\nu_3 = 0.74$ - см. рис. 4.7.4.

реобразуем $\nu_1 = 0.26$ в двоичную систему счисления:

$$0.26 \times 2 = 0.52 \text{ — запоминаем целую часть } 0;$$

$$0.52 \times 2 = 1.04 \text{ — запоминаем целую часть } 1;$$

$$0.04 \times 2 = 0.08 \text{ — запоминаем целую часть } 0;$$

$$0.08 \times 2 = 0.16 \text{ — запоминаем целую часть } 0;$$

$$0.16 \times 2 = 0.32 \text{ — запоминаем целую часть } 0;$$

$$0.32 \times 2 = 0.64 \text{ — запоминаем целую часть } 0;$$

$$0.64 \times 2 = 1.28 \text{ — запоминаем целую часть } 1;$$

$$\text{Итого, } \nu_1 = 0.26 = 0.0100001 \dots_b,$$

$$z_1 = \langle \alpha_1, \alpha_2 \rangle = \langle 01 \rangle = 1_d,$$

$$z_2 = \langle \alpha_3, \alpha_4 \rangle = \langle 00 \rangle = 0_d.$$

Таким образом, $\nu_1 \in \Delta(z_1, z_2) = \Delta(1, 0)$.

Аналогично $v_2 = 0.49 = 0.0111111\dots_b$, $z_1 = \langle \alpha_1, \alpha_2 \rangle = \langle 01 \rangle = 1_d$,
 $z_2 = \langle \alpha_3, \alpha_4 \rangle = \langle 00 \rangle = 3_d$. 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, $v_3 \in \Delta(z_1, z_2) = \Delta(2, 3)$.

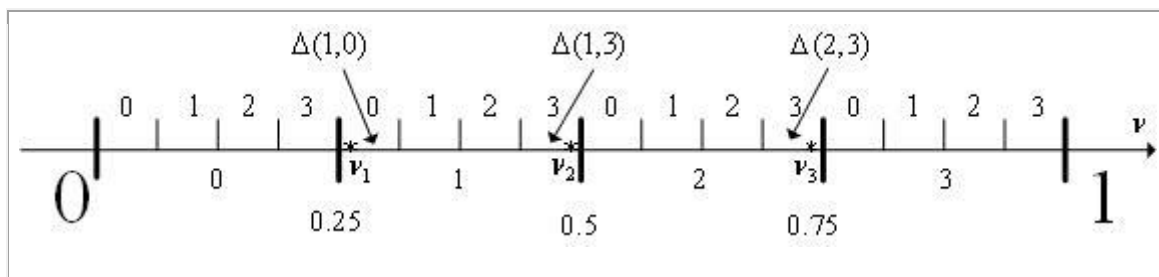


Рис. 4.7.4. К примеру 4.7.1

Определим на отрезке $[0, 1]$ функцию $\Phi(v) = \Phi(X) = P(v)$. Отметим, что если функция $\Phi(X)$ является непрерывной функцией, то функция $\Phi(v)$ также непрерывна. Однако эта функция является негладкой и многоэкстремальной, даже если исходная функция $\Phi(X)$ гладкая и унимодальная.

Таким образом, с помощью развертки Пеано многомерная задача глобальной условной оптимизации (4.7.1), (4.7.2) 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 параграфах 2.7-2.10 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:

1. Должна быть задана требуемая точность ε_x решения исходной задачи (4.7.1), (4.7.2) по X . Исходя из этой точности, предварительно должно быть определено S — количество разбиений области D (см. ниже).

2. Вычисления значений критерия оптимальности $\Phi(v)$, $v \in [0,1]$ должны производиться по следующей схеме:

- для заданного $v \in [0,1]$ находим S_n цифр его двоичного представления

$$v_b = 0, \alpha_1 \alpha_2 \dots \alpha_{S_n} \dots;$$

- определяем числа $z_1 = \langle \alpha_1 \alpha_2 \dots \alpha_n \rangle$, $z_2 = \langle \alpha_{n+1} \alpha_{n+2} \dots \alpha_{2n} \rangle, \dots,$

$$z_s = \langle \alpha_{(s-1)n+1} \alpha_{(s-1)n+2} \dots \alpha_{sn} \rangle;$$

- в гиперкубе $D(z_1, z_2, \dots, z_s)$ выбираем его центр $X_{D(z_1, z_2, \dots, z_s)}$;

- вычисляем значение критерия оптимальности $\Phi(X)$ в этой точке, которое и принимаем за значение $\Phi(v)$.

При заданной точности ε_x решения задачи (4.7.1), (4.7.2) по 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 и 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.

4.8. Метод Монте-Карло

Рассмотрим многомерную задачу глобальной условной оптимизации

$$\min_{X \in D \subset R^n} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (4.8.1)$$

где множество допустимых значений

$$D = \left\{ X / h_i(X) = 0, g_j(X) \geq 0, i \in [1, \dots, m], j \in [1, \dots, l] \right\}$$

определяется как ограничениями типа неравенств, так и ограничениями типа равенств.

Метод Монте-Карло относится к классу прямых методов случайного поиска.

Схема метода Монте-Карло.

1. Задаем общее количество испытаний N и полагаем счетчик числа итераций $r = 1$.
2. С помощью какого-либо программного генератора случайных чисел генерируем n компонент вектора $X^1 \in D$.
3. Вычисляем $\Phi(X^1)$ и полагаем $X^* = X^1, \Phi^* = \Phi(X^1), r = 1$.
4. Аналогично п.2 генерируем случайную точку $X^r \in D$. Вычисляем соответствующее значение критерия оптимальности $\Phi(X^r) = \Phi^*$.
5. Выполняем следующие присваивания: $X^* = \begin{cases} X^r, & \text{если } \Phi^r < \Phi^*, \\ X^*, & \text{иначе.} \end{cases}$
6. Если $r < N$ полагаем $r = r + 1$ и переходим на п.4, иначе принимаем X^*, Φ^* в качестве приближенного решения задачи и заканчиваем вычисления.

Отметим, что в простейшем случае точки $X^r \in D$ 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.

1. Задаем общее количество исходных случайных точек N .
2. С помощью какого-либо программного генератора случайных чисел генерируем N случайных точек X^1, X^2, \dots, X^N , принадлежащие множеству D .
3. Полагаем $r = 1$.
4. Исходя из точки X^r , каким-либо многомерным методом условной оптимизации (см. параграфы 4.1-4.5) находим локальный минимум $(X^*)^r$ функции $\Phi(X)$ в окрестности этой точки и вычисляем $\Phi\left((X^*)^r\right) = (\Phi^*)^r$.
5. Если $r < N$ полагаем $r = r + 1$ и переходим на п.4, иначе – переходим к следующему пункту.
6. Находим минимальное из чисел $(\Phi^*)^r, r \in [1, \dots, N]$. Пусть

$$\min_{r \in [1, \dots, N]} (\Phi^*)^r = (\Phi^*)^k.$$

Принимаем в качестве приближенного решения задачи $(X^*)^k, (\Phi^*)^k$ и заканчиваем вычисления.

Список литературы

1. *Аттетков А.В.* Методы оптимизации / А.В. Аттетков, С.В. Галкин, В.С. Зарубин. – М.: Из-во МГТУ им. Н.Э.Баумана, 2001.
2. *Батищев Д.И.* Поисквые методы оптимального проектирования / Д.И. Батищев. - М.: Советское Радио,1975.
3. *Болтянский В.Г.* Математические методы оптимального управления / В.Г. Болтянский. - М.: Наука, 1969.
4. *Зайченко Ю.П.* Курс лекций «Исследование операций».- <http://iasa.org.ua>.
5. *Карпенко А.П.* Методы оптимизации. - <http://bigor.bmstu.ru/?cnt/?doc=МО/base.cou> . Дата обращения 01.08.2016.
6. *Конюховский П.В.* Математические методы исследования операций в экономике / П.В. Конюховский. – СПб.: Изд-во С.-Петербургского Университета, 2008.
7. *Лесин В.В.* Основы методов оптимизации / В.В. Лесин, Ю.П. Лисовец. - М.: Изд-во МАИ, 1995.
8. *Ляшенко И.Н.* Линейное и нелинейное программирование / И.Н. Ляшенко, Е.А. Карагодова, Н.В. Черникова, Н.З. Шор. – Киев: Издательское объединение “Вища школа”, 1975.
9. *Таха Х.А.* Введение в исследование операций. Том 1 и Том 2 / Х.А. Таха. - М.: Мир, 1985.
10. *Федоренко Р.П.* Приближенное решение задач оптимального управления / Р.П. Федоренко. - М.: Наука, 1978.
11. *Хедли Дж.* Нелинейное и динамическое программирование / Дж. Хедли.- М.: Мир, 1967.