

Метод наискорейшего спуска (подъёма)

При решении задачи на минимизацию речь идёт о спуске, а в случае максимизации – о подъёме. В алгоритмах указанное отличие отражается в знаках при выборе направления, которое совпадает с направлением градиента в задачах максимизации и противоположно ему (знак «минус») в задачах минимизации.

Луи Огюст Коши предложил выбирать значение λ_k путём решения задачи одномерной минимизации из условия

$$\lambda_k^* \Rightarrow \min_{\lambda > 0} f(X_k + \lambda s_k), \quad (2.82)$$

где $s_k = -\nabla f(X_k)$ определяется значением градиента. При этом непосредственно значение λ_k может быть определено из уравнения

$$\frac{\partial f(X_k + \lambda s_k)}{\partial \lambda} = 0 \quad (2.83)$$

аналитически, либо путём численного решения задачи одномерной минимизации. Алгоритм метода тривиальный, изображён на рисунке 2.21.

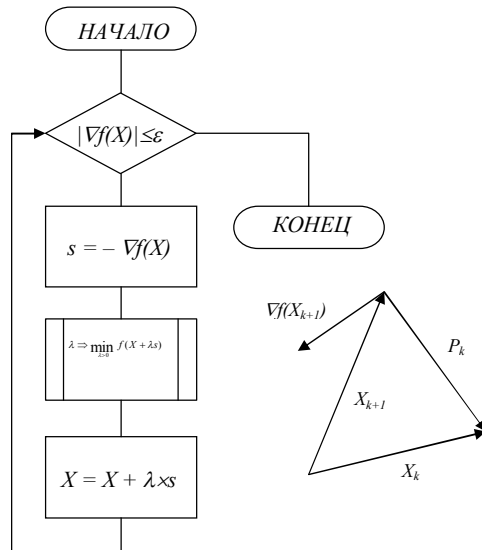


Рисунок 2.21 – Алгоритм метода наискорейшего спуска

Если ввести вектор, определяемый как $\bar{P}_k = \bar{X}_{k+1} - \bar{X}_k$, то вектор-градиент $\nabla \bar{f}(X_{k+1})$ перпендикулярен вектору приращения \bar{P}_k , то есть $\nabla \bar{f}(X_{k+1}) \perp \bar{P}_k$. Поэтому траектория спуска будет иметь, в общем случае, вид зигзага.

Тут уместно заметить, что метод наискорейшего спуска *не обладает* оптимальными свойствами, *не обеспечивая минимум шагов* при минимуме *вычислений*, если только *направление спуска не совпадает* с направлением на минимум, локальный или глобальный.

Последнее справедливо только для ограниченного класса функций, поэтому метод считается *полношаговым*. Наискорейшим спуском будет, например, для нелинейных функций чётных степеней.

Метод Ньютона (вторых производных)

Для итерационных методов, как мы знаем, характерна следующая связь между новыми и старыми координатами в пределах одной итерации

$$X_{k+1} = X_k + \Delta X_k. \quad (2.84)$$

Отыщем значение ΔX_k в ходе решения оптимизационной задачи с использованием ряда Тейлора 2-го порядка. Для функции одной переменной ряд, ограниченный до трёх членов, в окрестностях точки X_0 будет иметь вид

$$f(x) \cong f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2,$$

а для многомерного случая

$$f(X) \cong f(X_0) + \nabla^T f(X_0)(X - X_0) + \frac{1}{2} (X - X_0)^T \nabla^2 f(X_0)(X - X_0),$$

где $\nabla^T f(X_0)$ – матрица Гессе, составленная из частных производных второго порядка. Определяя ΔX_k ка (2.84), придём к формуле

$$f(X_{k+1}) = f(X_k) + \nabla^T f(X_k) \Delta X_k + \frac{1}{2} \Delta X_k^T \nabla^2 f(X_k) \Delta X_k.$$

Для отыскания оптимального значения ΔX_n последнее выражение подвергнем дифференцированию, а результат положим равным нулю. Отсюда

$$\Delta X_k = -\frac{\nabla f(X_k)}{\nabla^2 f(X_k)}.$$

Окончательно рекуррентное выражение для расчётов

$$X_{k+1} = X_k - \frac{\nabla f(X_k)}{\nabla^2 f(X_k)}. \quad (2.85)$$

Если сравнить (2.85) и (2.81), видно, что для данного случая

$$\lambda_k = [\nabla^2 f(X_k)]^{-1},$$

то есть, λ_k на каждой итерации равно обращённой матрице Гёссе в текущей точке.

Выражение (2.85) есть задание алгоритма метода Ньютона в виде формулы, которая применяется циклически применяется циклически, до удовлетворения необходимой точности расчётов, то есть $|\Delta X_k| \leq \varepsilon$.

Замечание. Обращение матрицы Гёссе на каждом шаге – основное неудобство метода, ведущее в временным затратам и росту погрешности вычисления. Последнее, отчасти, препятствует широкому использованию метода в практических приложениях.

Метод сопряжённого градиента (Флетчера - Ривса)

Понятие о сопряжённых направлениях

В общем случае, система линейно-независимых направлений поиска S_1, S_2, \dots, S_{n-1} называется **сопряжённой** по отношению к некоторой положительно определённой матрице Q если

$$S_i^T Q S_j = 0; \quad i \neq j; \quad i = \overline{1, n-1}, \quad j = \overline{1, n-1}.$$

Если $Q = I$ – единичная матрица, то вектора S_i и S_j , $i \neq j$ ортогональны.

Направление спуска S_1 будет сопряжено направлению S_0 , если выполняется тождество:

$$S_0^T \cdot \nabla^2 f(X_0) \cdot S_1 = 0.$$

Как известно из математики [16, 18], для положительно определённой матрицы её собственные вектора образуют систему из $n - 1$ направлений и такая система всегда существует, по крайней мере, одна.

Пусть исходное направление поиска в некоторой точке X_n есть

$$S_n = -\nabla f(x_n). \quad (2.86)$$

Сопряжённое направление будем строить по правилу

$$S_{n+1} = -\nabla f(x_{n+1}) + \omega_{n+1} S_n, \quad (2.87)$$

где ω_{n+1} – весовой коэффициент сопряжения. Этот коэффициент должен обеспечивать условие сопряжённости, то есть

$$S_n^T \nabla^2 f(x_n) S_{n+1} = 0. \quad (2.68)$$

Алгоритм движения точки в ходе поиска подчиняется известному выражению

$$x_{n+1} = x_n + \lambda S_n \Rightarrow \Delta x_n = \lambda S_n. \quad (2.89)$$

Необходимо найти нормирующий множитель ω_{n+1} .

Воспользуемся для начала аппроксимацией матрицы Гёссе по методу Л. Эйлера

$$\frac{\nabla f(x_{n+1}) - \nabla f(x_n)}{x_{n+1} - x_n} \cong H(x_n) = \nabla^2 f(x_n).$$

Из последнего выражения, определив знаменатель дроби через направление S_n по выражению (2.89), последует

$$\nabla f(x_{n+1}) - \nabla f(x_n) = H(x_n) \lambda S_n,$$

откуда получим

$$S_n^T = \frac{1}{\lambda} [\nabla f(x_{n+1}) - \nabla f(x_n)]^T H^{-1}(x_n). \quad (2.90)$$

Подстановка (2.86) в (2.87)

$$S_{n+1} = -\nabla f(x_{n+1}) - \omega_{n+1} \nabla f(x_n),$$

а, в дальнейшем, полученного результата совместно с (2.90) – в условие сопряжённости (2.88), даёт

$$\frac{1}{\lambda} [\nabla f(x_{n+1}) - \nabla f(x_n)]^T H^{-1}(x_n) H(x_n) [-\nabla f(x_{n+1}) - \omega_{n+1} \nabla f(x_n)] = 0.$$

Из последней записи, принимая во внимание, что $H^{-1}(x_n)H(x_n) = I$, выразим весовой коэффициент сопряжения:

$$\omega_{n+1} = \frac{\nabla^T f(x_{n+1}) \cdot \nabla f(x_{n+1})}{\nabla^T f(x_n) \cdot \nabla f(x_n)} = \frac{\|\nabla f(x_{n+1})\|^2}{\|\nabla f(x_n)\|^2}. \quad (2.91)$$

Результат показывает, что коэффициент сопряжения (2.91) равен отношению квадратов норм (длин) векторов текущего и предшествующего градиентов.

Алгоритм расчётов представлен ниже, на рисунке 2.22. Шаг λ^* отыскивается по методу Коши, как и в методе наискорейшего спуска.

2.5.2.3. Методы поиска переменной метрики (квазиНьютоновские)

Обращение матрицы Гёссе на каждом шаге итерации есть основной недостаток метода второго порядка, предложенного Ньютоном и рассмотренного нами выше. Указанная матрица может быть плохо обусловленной, а численное обращение трудоёмким и приводить к погрешности – число операций умножения пропорционально n^3 , где n – размерность матрицы.

Методы переменной метрики или **квазиньютоновские** производят построение аппроксимирующей матрицы $[H(X_k)]^{-1}$ в процессе спуска.

Пусть алгоритм движения точки подчиняется закону

$$x_{n+1} = x_n - \lambda_n D_n \nabla f(x_n),$$

где D_n - аппроксимирующая матрица, λ_n - стабилизирующий множитель.

Аппроксимирующую матрицу D_n строят по правилу:

$$H^{-1}(x_n) \approx D_{n+1} = \omega(D_n + \Delta D_n), \quad (2.92)$$

ω - нормирующий множитель.

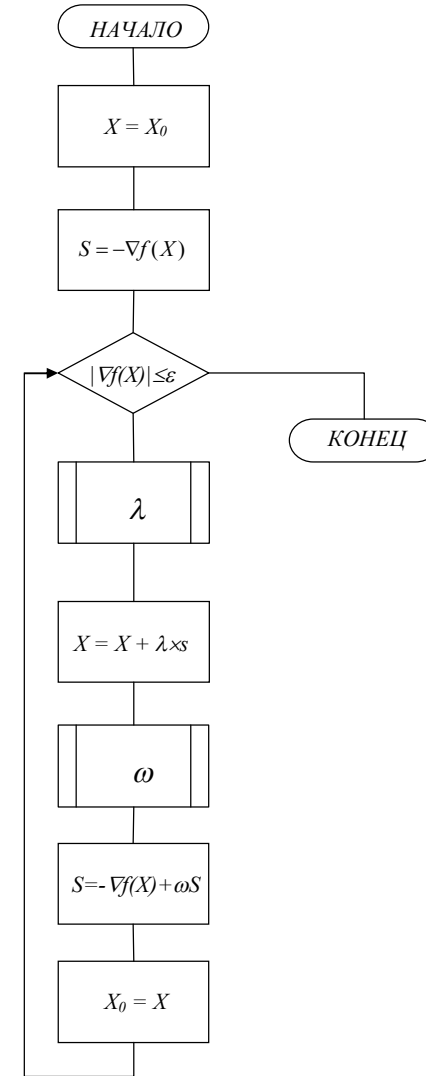


Рисунок 2.22 – Алгоритм метода Флетчера – Ривса

Определим приращение ΔD_n , обеспечивающее корректное построение матрицы $[H(X_k)]^{-1}$.

Воспользуемся аппроксимацией матрицы Гёссе

$$H(x_n) = \frac{\Delta g_n}{\Delta x_n}, \quad (2.93)$$

где $\Delta g_n = \nabla f(x_{n+1}) - \nabla f(x_n)$, а $\Delta x_n = x_{n+1} - x_n$.

Приращение Δx_n определим из (2.93):

$$\Delta x_n = H^{-1}(x_n) \cdot \Delta g_n$$

и подставим сюда определение (2.92) даёт

$$\Delta x_n = \omega(D_n + \Delta D_n)\Delta g_n \Rightarrow \Delta D_n \Delta g_n = \frac{1}{\omega} \Delta x_n - D_n \Delta g_n,$$

откуда

$$\Delta D_n = \frac{1}{\omega} \cdot \frac{\Delta x_n}{\Delta g_n^T} \cdot \frac{Y^T}{Y} - \frac{D_n \Delta g_n}{\Delta g_n^T} \cdot \frac{Z^T}{Z}.$$

Фиктивные добавки – векторы Y и Z , размерности n , решают задачи определенности матричных операций, сходимости D_n к $H^{-1}(x_n)$ и её положительной определённости.

В алгоритме, предложенном **Бройденом**, положено: $\omega = 1$, $Y = Z = \Delta x_n - D_n \Delta g_n$.

В алгоритме **Давидона-Флетчера-Пауэлла** аналогичные добавки определяются как: $\omega = 1$, $Y = \Delta x_n$, $Z = D_n \Delta g_n$.

В обоих случаях, в качестве начального приближения используется единичная матрица I .

Для последнего алгоритма

$$D_{n+1} = D_n + \frac{\Delta x_n}{\Delta g_n^T} \cdot \frac{\Delta x_n^T}{\Delta x} - \frac{D_n \Delta g_n}{\Delta g_n^T} \cdot \frac{(D_n \Delta g_n)^T}{D_n \Delta g_n} = D_n + A_n - B_n$$

или

$$D_{n+1} = I + \sum_{i=1}^n A_i - \sum_{i=1}^n B_i,$$

в котором A_n обеспечивает сходимость D_n к $H^{-1}(x_n)$, а B_n положительную определённую.

Схема работы алгоритма – типовая для градиентных методов.

Параметр $\lambda_n^* \Rightarrow \min_{\lambda > 0} f(X_n + \lambda_n s_n)$, где $s_n = -\nabla f(X_n)$, вычисляется по методу Коши.

Условия окончания работы алгоритма задаются одним из нижеприведенных способов.

1. Каждая из компонент векторов $D_n \nabla f(x_n)$ и $\lambda_n D_n \nabla f(x_n)$ меньше по модулю некоторой заданной величины ε .

2. Длина (норма) каждого из векторов $D_n \nabla f(x_n)$ и $\lambda_n D_n \nabla f(x_n)$ меньше ε .

2.5.3. Поиск экстремумов в задачах нелинейного программирования при ограничениях типа “равенство” (метод Лагранжа)

В основу метода положена идея сведения задачи поиска условного экстремума к поиску безусловного экстремума специальной функции. Пусть математическая модель представлена в виде

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &\rightarrow \max, \\ \begin{cases} h_1(x_1, x_2, \dots, x_n) \geq 0, \\ h_2(x_1, x_2, \dots, x_n) \geq 0, \\ \dots \\ h_m(x_1, x_2, \dots, x_n) \geq 0. \end{cases} \end{aligned} \quad (2.94)$$

Допустим, что функции $f(X)$ и $h_i(X)$, $i = 1, m$, входящие в (2.94), нелинейные и имеют непрерывные частные производные.

Введём набор множителей $\lambda_1, \lambda_2, \dots, \lambda_m$, по числу ограничений в модели (2.94), и составим функцию вида

$$L(x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_m) = f(x_1, x_2, \dots, x_n) + \sum_{i=1}^m \lambda_i h_i(x_1, x_2, \dots, x_n) \quad (2.95)$$

Полученная **функция** называется функцией **Лагранжа**, а множители λ_i , $i = 1, m$, – **коэффициентами** или **множителями Лагранжа**.

Часть выражения, стоящего под знаком суммы в (2.95), можно интерпретировать как результат решения системы уравнений, в ходе которого происходит сложение или вычитание отдельных её уравнений, помноженных на коэффициенты.

Очевидно, что экстремум (2.95) будет при таких значениях $X^* = \{x_1^*, x_2^*, \dots, x_n^*\}$ и $\Lambda^* = \{\lambda_1^*, \lambda_2^*, \dots, \lambda_m^*\}$, которые будут удовлетворять системе уравнений

$$\left. \begin{aligned} \frac{\partial L(X^*, \Lambda^*)}{\partial x_j} &= 0, j = 1, \bar{n}, \\ \frac{\partial L(X^*, \Lambda^*)}{\partial \lambda_i} &= 0, i = 1, \bar{m}. \end{aligned} \right\} \quad (2.96)$$

Эта система компактно записана в форме

$$\nabla f(X^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(X^*) = 0. \quad (2.97)$$

Условия (2.96) суть **необходимые** условия существования экстремума, а запись (2.97) – отражает **достаточность**.

Вектор $\nabla f(X^*)$ как бы лежит в гиперплоскости, натянутой на вектора $\nabla h_i(X^*)$, эдакий зонтик. Известна следующая теорема, носящая имя Лагранжа.

Теорема. Пусть существует точка X^* , в которой достигается экстремум функции $f(X)$ при ограничениях $h_i(X) = 0, i = 1, \bar{m}$. Если ранг матрицы $I = \left[\frac{\partial h_i}{\partial x_j} \right], i = 1, \bar{m}; j = 1, \bar{n}$ в точке X^* равен m , то существует m вещественных чисел $\lambda_1, \lambda_2, \dots, \lambda_m$, **не все из которых равны нулю одновременно**, при которых выполняется условие

$$\nabla f(X^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(X^*) = 0.$$

Алгоритм метода Лагранжа

1. Составить функцию Лагранжа $L(X, \Lambda)$.
2. Найти частные производные функции Лагранжа (2.96).
3. Решить систему уравнений для определения X^* и Λ^* .
4. Исследовать полученные точки на максимум или минимум.

Пример практического применения метода Лагранжа

Содержательная постановка.

Цикл эксплуатации блока датчиков информационного зонда включает следующие фазы: τ_{II} – подготовки к применению; $\tau_{Г}$ – готовности к применению, $\tau_{Р}$ – работы (применения). Требование к надёжности формулируется в виде показателя безотказной работы R . Каковы должны

быть требования по надёжности к каждой фазе эксплуатации, если известны потери от отказа на каждой фазе, задаваемые неотрицательными величинами $C_i > 0, i = II, Г, Р$?

Время эксплуатации изделия в целом есть

$$\tau_{\Sigma} = \tau_{II} + \tau_{Г} + \tau_{Р},$$

а надёжность характеризуется показателем

$$R = P_{II} P_{Г} P_{Р},$$

где $P_i, i = II, Г, Р$ – вероятность безотказной работы на соответствующей фазе.

Если последствия отказов равнозначны, то задача имеет тривиальное решение:

$$P_i = \sqrt[m]{R}, i = II, Г, Р.$$

В противном случае имеем оптимизационную задачу: необходимо рассчитать вероятностные характеристики таким образом, чтобы общие потери были минимальные. Функция цели или функция потерь имеет вид

$$C_{\Sigma} = C_{II}(1-P_{II})P_{Г}P_{Р} + C_{Г}P_{II}(1-P_{Г})P_{Р} + C_{Р}P_{II}P_{Г}(1-P_{Р}),$$

при ограничении

$$R = P_{II} P_{Г} P_{Р}.$$

Таким образом, имеем оптимизационную задачу, которую будем решать изложенным выше методом.

Функция Лагранжа получается такова:

$$L = C_{\Sigma} + \lambda [P_{II} P_{Г} P_{Р} - R].$$

В ходе применения метода, λ из первого, например, уравнения для частных производных $\frac{\partial L}{\partial P_i}$, подставляется в остальные.

$$\begin{cases} C_r \cdot P_{\Pi} - C_{\Pi} \cdot P_r = 0, \\ C_r \cdot P_p - C_p \cdot C_r = 0, \\ R = P_{\Pi} \cdot P_r \cdot P_p. \end{cases} \Rightarrow \begin{cases} P_r = \frac{C_{\Pi} \cdot R}{C_r \cdot P_{\Pi}^2}, \\ P_{\Pi} = \frac{C_p \cdot R}{C_r \cdot P_p^2}. \end{cases}$$

Далее следуют алгебраические преобразования. Окончательно имеем оптимум при переменных

$$P_{\Pi}^* = \sqrt[3]{\frac{C_{\Pi}^2 R}{C_r C_p}}, \quad P_r^* = \sqrt[3]{\frac{C_r^2 R}{C_{\Pi} C_p}}, \quad P_p^* = \sqrt[3]{\frac{C_p^2 R}{C_{\Pi} C_r}}.$$

Иногда стоимость потерь при отказах удобно выражать в относительных единицах:

$$a_1 = \frac{C_r}{C_{\Pi}}, \quad a_2 = \frac{C_{\Pi}}{C_p}, \quad a_3 = \frac{C_p}{C_r}, \text{ причём } a_1 a_2 a_3 = 1.$$

Тогда решение в оптимуме примет вид

$$P_{\Pi}^* = \sqrt[3]{\frac{Ra_2}{a_1}}, \quad P_r^* = \sqrt[3]{\frac{Ra_1}{a_3}}, \quad P_p^* = \sqrt[3]{\frac{Ra_3}{a_2}}.$$

2.5.4. Общий случай задачи нелинейного программирования

Задачи общего вида подразделяются на задачи **выпуклого** и **вогнутого** программирования, но, независимо от их типа, на переменные накладываются условия неотрицательности $x_j \geq 0, j = 1, n$.

Математическая модель для задачи выпуклого программирования имеет вид

$$\begin{cases} f(x_1, x_2, \dots, x_n) \rightarrow \min, \\ g_1(x_1, x_2, \dots, x_n) \leq 0, \\ g_2(x_1, x_2, \dots, x_n) \leq 0, \\ \dots \\ g_m(x_1, x_2, \dots, x_n) \leq 0, \\ x_j \geq 0, \quad j = 1, \bar{n}. \end{cases} \quad (2.98)$$

когда функции $f(X)$ и все $g_i(X), i = 1, m$ выпуклы (в смысле НП-задач).

Для случая вогнутого программирования характерно

$$\begin{cases} f(x_1, x_2, \dots, x_n) \rightarrow \max, \\ g_1(x_1, x_2, \dots, x_n) \geq 0, \\ g_2(x_1, x_2, \dots, x_n) \geq 0, \\ \dots \\ g_m(x_1, x_2, \dots, x_n) \geq 0, \\ x_j \geq 0, \quad j = 1, \bar{n}. \end{cases}$$

при функциях $f(X)$ и всех $g_i(X), i = 1, m$ вогнутых (выпуклых вверх).

Связь между задачами выпуклого и вогнутого программирования в части целевой функции и системы ограничений определяется множителем (-1) . То есть

$$\max U(X) = \min [-U(X)].$$

Поэтому, обозначая

$$\begin{aligned} \hat{f}(X) &= -f(X), \\ \hat{g}_i(X) &= -g_i(X), \quad i = 1, \bar{n}. \end{aligned}$$

перейдём к следующей задаче выпуклого программирования:

$$\begin{cases} \hat{f}(x_1, x_2, \dots, x_n) \rightarrow \min, \\ \hat{g}_1(x_1, x_2, \dots, x_n) \leq 0, \\ \hat{g}_2(x_1, x_2, \dots, x_n) \leq 0, \\ \dots \\ \hat{g}_m(x_1, x_2, \dots, x_n) \leq 0, \\ x_j \geq 0, \quad j = 1, \bar{n}. \end{cases}$$

Рассмотрим общий случай задачи НП-программирования (2.98):

Если в точке минимума X^* неравенство $g_i(X)$ выполняется как **равенство**, то оно называется **активным**. В этом случае, согласно теореме Лагранжа, формула (2.97), для активных ограничений имеем

$$\nabla f(X^*) + \sum_I \lambda_i \nabla g_i(X^*) = 0, \quad I = \{i : g_i(X^*) = 0\}. \quad (2.99)$$

Введём в рассмотрение вектор Λ с неотрицательными компонентами, обеспечивающими **условие линейной независимости**

$$\sum_{i=1}^m \lambda_i g_i(X^*) = 0. \quad (2.100)$$

Условия (2.99) и (2.100) – суть условия **регулярности** ограничений.

Существует теорема Куна – Таккера.

Теорема. Пусть функции $f(X)$ и $g_i(X)$, $i = 1, m$ обладают частными производными на некоторой области \mathcal{R} , содержащей X^* . Точка X^* будет являться точкой минимума функции $f(X)$ при ограничениях $g_i(X) \leq 0$, $i = 1, m$, удовлетворяющих условиям регулярности в виде линейной независимости, если существуют такие неотрицательные множители Лагранжа $\lambda_1, \lambda_2, \dots, \lambda_m$, что

$$\begin{cases} \nabla f(X^*) + \sum_{i=1}^m \lambda_i \nabla g_i(X^*) = 0; \\ \sum_{i=1}^m \lambda_i g_i(X^*) = 0; \\ \lambda_i \geq 0, \quad i = 1, m. \end{cases} \quad (2.101)$$

Ограничения (2.100) называются **условиями дополняющей нежёсткости** или условиями Слейтера, а точка X^* – точкой Куна-Таккера.

Смысл множителей Лагранжа в данной ситуации состоит в том, что они позволяют “нейтрализовать” (аннулировать) неактивные ограничения, которые и без того выполняются. Тем самым, задача нелинейного программирования общего вида сводится к задаче Лагранжа.

2.5.4.1. Седловая точка в НП-задачах

Определение. Пара векторов X^* и Λ^* называется **седловой точкой** функции $L(X, \Lambda)$ на области \mathcal{R} , если для всех $\lambda \geq 0$ и $X \in \mathcal{R}$ выполняется условие

$$L(X^*, \Lambda) \leq L(X^*, \Lambda^*) \leq L(X, \Lambda^*), \quad (2.102)$$

называемое **неравенством седловой точки**.

В самой седловой точке выполняется равенство, называемое ещё неравенством(условием) максимина или минимакса.

$$\max_{\lambda_i \geq 0} \min_{X \in \mathcal{R}} L(X, \Lambda) = \min_{X \in \mathcal{R}} \max_{\lambda_i \geq 0} L(X, \Lambda), \quad i = 1, \bar{n}.$$

Пример вида простейшей функции Лагранжа с седловой точкой показан на рисунке 2.23.

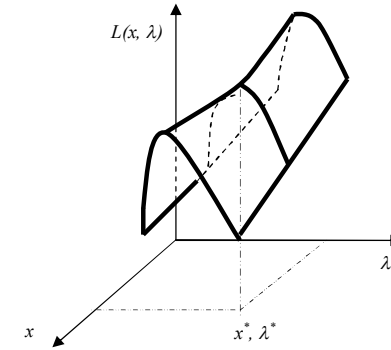


Рисунок 2.23 – Седло нелинейной функции

Существует нижеследующая теорема о седловой точке.

Теорема. Пусть $f(X)$ и все $g_i(X)$, $i = 1, m$ выпуклы, а функции $g_i(X)$ удовлетворяют условиям **регулярности Слейтера** (вида $\exists_X \forall_i g_i(X) < 0$). Вектор X^* является решением задачи нелинейного программирования

$$\begin{aligned} \min f(X), \\ g_i(X) \leq 0, \quad i = 1, \bar{m}. \end{aligned}$$

тогда и только тогда, когда существует такой вектор Λ с неотрицательными компонентами, что выполняются неравенство **седловой точки** (2.102) и **условие линейной независимости** (2.100).

Условие регулярности Слейтера означает наличие в области ограничений точек, в которых ограничения задачи выполняются строго.

Таким образом, НП-задача может быть, в принципе, сведена к задаче поиска седловой точки функции Лагранжа. При этом предпочтения будут отданы тому решению, ход решения которого более прост.

2.5.4.2. Применение теоремы Куна-Таккера к НП-задачам

Имеем выпуклую задачу НП-программирования

$$\begin{aligned} \min f(X); \\ \begin{cases} g_i(X) \leq 0, & i = 1, \bar{m}; \\ x_j \geq 0, & j = 1, \bar{n}. \end{cases} \end{aligned}$$

Введём дополнительные ограничения, учитывающие неотрицательность переменных x_j в явном виде: $x_j = -h_j(x_j)$, $j = 1, n$, откуда непосредственно следует, что $h_j(x_j) \leq 0$, $j = 1, n$.

Для такой расширенной системы построим функцию Лагранжа

$$L(X, \Lambda, U) = f(X) + \sum_{i=1}^m \lambda_i g_i(X) + \sum_{j=1}^n u_j h_j(x_j),$$

к которой применим теорему Куна-Таккера, найдём $\nabla L(X, \Lambda, U)$. Это эквивалентно системе уравнений

$$\begin{cases} \frac{\partial L(X, \Lambda, U)}{\partial x_j} = \frac{\partial f(X)}{\partial x_j} + \sum_{i=1}^m \lambda_i \frac{\partial g_i(X)}{\partial x_j} - u_j = 0, & j = 1, \bar{n}; \\ u_j x_j = 0, & j = 1, \bar{n}; \\ \lambda_i g_i(X) = 0, & i = 1, \bar{m}; \\ \lambda_i \geq 0, & i = 1, \bar{m}. \end{cases}$$

В точке Куна-Таккера будем иметь следующее:

$$\begin{cases} \frac{\partial L(X^\circ, \Lambda^\circ)}{\partial x_j} \geq 0, & j = 1, \bar{n}; \\ \frac{\partial L(X^\circ, \Lambda^\circ)}{\partial x_j} \cdot x_j^\circ = 0, & j = 1, \bar{n}; \\ \frac{\partial L(X^\circ, \Lambda^\circ)}{\partial \lambda_i} = g_i(X^\circ) \leq 0, & i = 1, \bar{m}; \\ \lambda_i g_i(X^\circ) = 0, & i = 1, \bar{m}. \end{cases}$$

Знак “ \geq ” в первом неравенстве записан, потому, что не все значения u_j равны нулю, а первых два ограничения, выделенные “ $\}$ ”, есть условия дополняющей нежёсткости по u_j .

Для вогнутого программирования, благодаря симметрии задач получается

$$\begin{cases} \frac{\partial L(X^\circ, \Lambda^\circ)}{\partial x_j} \leq 0, & j = 1, \bar{n}; \\ \frac{\partial L(X^\circ, \Lambda^\circ)}{\partial x_j} \cdot x_j^\circ = 0, & j = 1, \bar{n}; \\ \frac{\partial L(X^\circ, \Lambda^\circ)}{\partial \lambda_i} = g_i(X^\circ) \geq 0, & i = 1, \bar{m}; \\ \frac{\partial L(X^\circ, \Lambda^\circ)}{\partial \lambda_i} \cdot \lambda_i^\circ = 0, & i = 1, \bar{m}. \end{cases}$$

Таким образом, теорема Куна-Таккера предоставляет исследователю удобный инструмент, позволяющий по условию задачи нелинейного программирования построить систему уравнений, решение которой даёт систему векторов X и Λ . Эти векторы дополнительно проверяются на выполнение условий линейной независимости и неотрицательности.