

## 11. МЕТОДЫ ОПТИМИЗАЦИИ

# АЛГОРИТМЫ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

# 11.1. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ БЕЗ ОГРАНИЧЕНИЙ

В этом разделе представлен *градиентный* метод решения задач нелинейного программирования без ограничений на переменные. В этом реализуется для нахождения экстремальной точки используется градиент целевой функции.

## Градиентный метод

В этом разделе рассматривается метод решения задач нелинейного программирования с дважды непрерывно дифференцируемыми функциями. Основная идея метода заключается в построении последовательности точек, с учетом направления градиента исследуемой функции.

Одним из градиентных методов является изложенный ранее метод Ньютона – Рафсона, который ориентирован на решение систем уравнений. Здесь рассматривается еще один метод, который называется методом *наискорейшего подъема*. В русской математической литературе описываемый метод называется *методом наискорейшего спуска* и ориентирован на решение задач безусловной минимизации.

Согласно градиентному методу, вычисления завершаются при нахождении точки, в которой градиент функции равен нулю. Но это лишь необходимое условие для того, чтобы в данной точке находился оптимум. Чтобы удостовериться, что найдена именно точка оптимума, обычно привлекаются свойства вогнутости или выпуклости функции  $f(x)$ .

Рассмотрим задачу максимизации функции  $f(x)$ . Пусть  $X^0$  – начальная точка, с которой начинается реализация метода,  $\nabla f(X^k)$  – градиент функции  $f$  в  $k$ -й точке  $X^k$ . Идея метода сводится к определению в данной точке направления  $p$ , вдоль которого производная по направлению  $df/dp$  достигает своего максимума. Это достигается в том случае, когда последовательные точки  $X^k$  и  $X^{k+1}$  связаны соотношением

$$X^{k+1} = X^k + r^k \nabla f(X^k),$$

где  $r^k$  – параметр, называемый **длиной шага**.



Обычно значение параметра  $r^k$  выбирается из условия, чтобы в точке  $X^{k+1}$  наблюдалось максимальное увеличение значения целевой функции  $f$ . Другими словами, если функция  $h(r)$  определяется соотношением

$$h(r) = f(X^k + r^k \nabla f(X^k)),$$

то  $r^k$  равен значению  $r$ , на котором функция  $h(r)$  достигает максимума. Поскольку  $h(r)$  является функцией одной переменной, для нахождения ее максимума можно применить метод прямого поиска экстремума, если, конечно, функция  $h(r)$  строго одновершинная.

Описанная процедура заканчивается, когда две последовательные точки  $X^k$  и  $X^{k+1}$  близки. Это эквивалентно тому, что  $r^k \nabla f(X^k) \approx 0$ . Поскольку  $r^k \neq 0$ , последнее соотношение означает, что в точке  $X^k$  выполняется необходимое условие экстремума  $\nabla f(X^k) = 0$ .

### Пример 9.1

Рассмотрим задачу максимизации функции

$$f(x_1, x_2) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2.$$

Функция  $f(x_1, x_2)$  является квадратичной и нам известно, что ее абсолютный максимум достигается в точке  $(x_1^*, x_2^*) = (1/3; 4/3)$ . Решим эту задачу методом наискорейшего подъема. На рис. 9.1 изображена последовательность получаемых точек. Градиенты в двух последовательных итерационных точках с необходимостью являются ортогональными (перпендикулярными) друг к другу. Для данной функции градиент вычисляется по формуле

$$\nabla f(X) = (4 - 4x_1 - 2x_2, 6 - 2x_1 - 4x_2).$$

Пусть начальной точкой является  $X^0 = (1, 1)$ .

*Первая итерация*

$$\nabla f(X^0) = (-2, 0).$$

Следующая точка  $X^1$  находится на основании формулы

$$X = (1, 1) + r(-2, 0) = (1 - 2r, 1).$$

Отсюда получаем выражение для функции  $h(r)$ :

$$h(r) = f(1 - 2r, 1) = -2(1 - 2r)^2 + 2(1 - 2r) + 4.$$



Оптимальная длина шага  $r$ , при которой функция  $h(r)$  достигает максимума, равна  $1/4$ . Таким образом, мы нашли, что  $X^1 = (1/2, 1)$ .

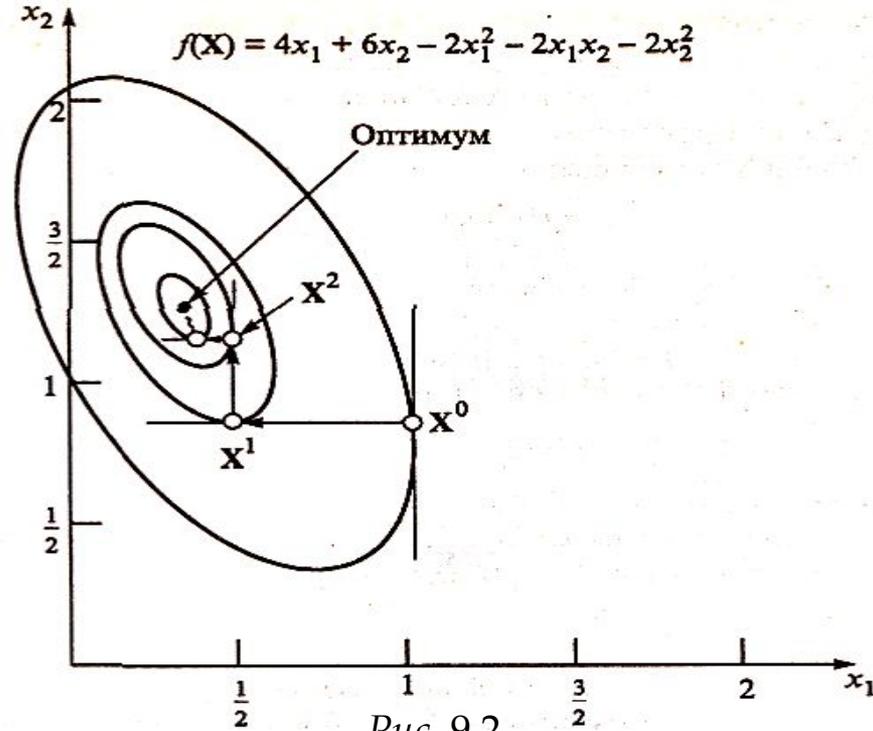


Рис. 9.2.

### Вторая итерация

$$\nabla f(X^1) = (0, 1).$$

Теперь следующая точка  $X^2$  определяется выражением

$$X = \left(\frac{1}{2}, \frac{5}{4}\right) + r \left(-\frac{1}{2}, 0\right) = \left(\frac{1-r}{2}, \frac{5}{4}\right).$$

Следовательно,

$$h(r) = -2(1+r)^2 + 5(1+r) + 3/2$$

Отсюда получаем  $r = 1/4$  и  $X^2 = (1/2, 5/4)$ .

### Третья итерация

$$\nabla f(X^2) = (-1/2, 0).$$



Отсюда получаем  $r = 1/4$  и  $X^2 = (1/2, 5/4)$ .

*Третья итерация*

$$\nabla f(X^2) = (-1/2, 0).$$

Точка  $X^3$  определяется выражением

$$X = \left(\frac{1}{2}, 1\right) + r(0, 1) = \left(\frac{1}{2}, 1+r\right).$$

Следовательно,

$$h(r) = -2(1+r)^2 + 5(1+r) + 3/2$$

Отсюда получаем  $r = 1/4$  и  $X^3 = (3/8, 5/4)$ .

*Четвертая итерация*

$$\nabla f(X^3) = (0, 1/4).$$

Точка  $X^4$  определяется выражением

$$X = \left(\frac{1}{2}, \frac{5}{4}\right) + r\left(-\frac{1}{2}, 0\right) = \left(\frac{1-r}{2}, \frac{5}{4}\right).$$

Следовательно,

$$h(r) = -\frac{1}{2}(1-r)^2 + \frac{3}{4}(1-r) + \frac{35}{8}.$$

Отсюда имеем  $r = 1/4$  и  $X^4 = (3/8, 21/16)$ .

*Пятая итерация*

Точка  $X^5$  определяется выражением

$$X = \left(\frac{3}{8}, \frac{5}{4}\right) + r\left(0, \frac{1}{4}\right) = \left(\frac{3}{8}, \frac{5+r}{4}\right).$$

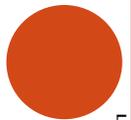
Поэтому

$$h(r) = -\left(\frac{1}{8}\right)(5+r)^2 + \left(\frac{21}{16}\right)(5+r) + \frac{39}{32}.$$

Отсюда получаем  $r = 1/4$  и  $X^5 = (11/32, 21/16)$ .

*Шестая итерация*

Так как  $\nabla f(X^5) \approx 0$ , вычисления можно закончить. Получена *приближенная* точка максимума  $X^5 = (0.3437, 1.3125)$ . Точный максимум достигается в точке  $X^* = (0.3333, 1.3333)$ .



# 11.2. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ С ОГРАНИЧЕНИЯМИ

Общая задача нелинейного программирования с ограничениями записывается в виде:

максимизировать (или минимизировать)  $z = f(\mathbf{X})$

при ограничениях

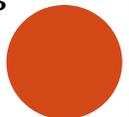
$$g(\mathbf{X}) \leq 0.$$

Условия неотрицательности переменных  $X \geq 0$  составляют часть заданных ограничений общего вида. Предполагается также, что, по крайней мере, одна из функций  $f(x)$  или  $g(X)$  является нелинейной и все функции непрерывно дифференцируемы.

Универсальных алгоритмов решения задач нелинейного программирования не существует, и связано это, главным образом, с разнообразием нелинейных функций. Возможно, наиболее общим результатом, имеющим отношение к рассматриваемым задачам, являются условия Куна – Таккера. Как показано в ранее, эти условия являются лишь необходимыми для существования экстремума, за исключением, когда функции  $f(x)$  и  $g(X)$  являются выпуклыми или вогнутыми.

Существует ряд алгоритмов решения задач нелинейного программирования, которые условно можно разделить на *непрямые и прямые*. В непрямых методах решение задачи нелинейного программирования сводится к решению одной или нескольких *линейных* задач, порожденных исходной. Прямые методы непосредственно имеют дело с исходной нелинейной задачей и позволяют построить последовательность точек, сходящуюся к точке экстремума.

Непрямые методы представлены алгоритмами сепарабельного, квадратичного, геометрического и стохастического программирования. К числу прямых методов относится метод линейных комбинаций и метод последовательной безусловной максимизации, изложенный далее вкратце.



# Сепарабельное программирование

Функция  $f(x_1, x_2, \dots, x_n)$  называется **сепарабельной** (разделимой), если она представляется в виде суммы и функций одной переменной  $f_1(x_1), f_2(x_2), \dots, f_n(x_n)$ , т.е.

$$f(x_1, x_2, \dots, x_n) = f(x_1) + f(x_2) + \dots + f(x_n).$$

К примеру, линейная функция

$$h(x_1, x_2, \dots, x_n) = a_1x_1 + a_2x_2 + \dots + a_nx_n$$

(здесь  $a_i, i = 1, 2, \dots, n$  – константы) является сепарабельной. Функция же

$$h(x_1, x_2, x_3) = x_1 \sin(x_2 + x_3) + x_2$$

таковой не является.

Некоторые нелинейные функции сепарабельными непосредственно не являются, однако могут быть приведены к такому виду путем соответствующих подстановок. Рассмотрим, к примеру, задачу максимизации функции  $z = x_1x_2$ . Если ввести обозначение  $y = x_1x_2$ , то  $\ln y = \ln x_1 + \ln x_2$  и задача принимает следующий вид.

Максимизировать  $z = y$

при ограничении

$$\ln y = \ln x_1 + \ln x_2$$

т.е. она является сепарабельной. При такой замене предполагается, что переменные  $x_1$  и  $x_2$  принимают *положительные* значения, иначе логарифмическая функция не определена.

В случае, когда переменные  $x_1$  и  $x_2$  принимают и нулевые значения (т.е.  $x_1, x_2 \geq 0$ ), можно поступить следующим образом. Пусть  $\delta_1$  и  $\delta_2$  – положительные константы, введем новые переменные  $w_1 = x_1 + \delta_1$  и  $w_2 = x_2 + \delta_2$ .

Эти переменные принимают только положительные значения. Теперь имеем

$$x_1x_2 = w_1w_2 - \delta_2w_1 - \delta_1w_2 + \delta_1\delta_2.$$

Пусть  $y = w_1w_2$ , тогда исходная задача эквивалентна следующей.

Максимизировать  $z = y - \delta_2w_1 - \delta_1w_2 + \delta_1\delta_2$

при ограничениях

$$\ln y = \ln w_1 + \ln w_2, w_1 \geq \delta_1, w_2 \geq \delta_2.$$



В этой задаче все функции сепарабельные.

Примерами других функций, которые в результате замены переменных приводятся к сепарабельным, могут служить и . В этих случаях для реализации свойства сепарабельности следует применить соответствующий вариант описанной выше процедуры.

В сепарабельном программировании рассматриваются задачи нелинейного программирования, в которых как целевая функция, так и функции ограничений являются сепарабельными. В этом разделе рассматриваются методы приближенного решения задачи сепарабельного программирования, основанные на линейной аппроксимации функций и на симплекс-методе линейного программирования.

Функцию одной переменной  $f(x)$  можно аппроксимировать кусочно-линейной функцией с помощью методов частично-целочисленного программирования. Пусть требуется аппроксимировать функцию  $f(x)$  на интервале  $[a, b]$ . Обозначим через  $a_k, k = 1, 2, \dots, K, k$ -ю точку разбиения интервала  $[a, b]$ , причем  $a = a_1 < a_2 < \dots < a_K = b$ . Тогда функция  $f(x)$  аппроксимируется следующей кусочно-линейной функцией:

$$f(x) \approx \sum_{k=1}^K f(a_k) t_k, \quad x = \sum_{k=1}^K a_k t_k,$$

где  $t_k$  — неотрицательный весовой коэффициент, связанный с  $k$ -й точкой разбиения интервала. Весовые коэффициенты удовлетворяют условию

$$\sum_{k=1}^K t_k = 1.$$

Методы частично-целочисленного программирования гарантируют корректность такой аппроксимации. В частности, аппроксимация является обоснованной в следующих случаях.

Если не больше двух весовых коэффициентов  $t_k$  имеют положительные значения.

Если значение  $t_k$  больше нуля, то положительным может оказаться лишь один из смежных весовых коэффициентов  $t_{k+1}$  или  $t_{k-1}$ .

Чтобы показать, как выполняются перечисленные условия, рассмотрим общую задачу сепарабельного программирования.

Максимизировать (минимизировать)  $z = \sum_{i=1}^n f_i(x_i)$

при ограничениях

$$\sum_{i=1}^n g_i^j(x_i) \leq b_j, \quad j = 1, 2, \dots, m.$$



Эту задачу можно привести к задаче частично-целочисленного программирования следующим образом. Обозначим через  $K_i$  число точек разбиения для  $i$ -й переменной  $x_i$ , а через  $a_i^k$  —  $k$ -ю точку разбиения. Пусть  $t_i^k$  — весовой коэффициент, ассоциируемый с  $k$ -й точкой разбиения для  $i$ -й переменной. Тогда эквивалентная задача частично-целочисленного программирования имеет вид

$$\text{максимизировать (или минимизировать)} \quad z = \sum_{i=1}^n \sum_{k=1}^{K_i} f_i(a_i^k) t_i^k$$

при ограничениях

$$\sum_{i=1}^n \sum_{k=1}^{K_i} g_i^j(a_i^k) t_i^k \leq b_j, \quad j = 1, 2, \dots, m.$$

$$0 \leq t_i^1 \leq y_i^1, \quad 0 \leq t_i^k \leq y_i^{k-1}, \quad k = 2, 3, \dots, K_i - 1,$$

$$0 \leq t_i^{K_i} \leq y_i^{K_i-1},$$

$$\sum_{k=1}^{K_i-1} y_i^k = 1,$$

$$\sum_{k=1}^{K_i} t_i^k = 1,$$

$$y_i^k = 0 \text{ или } 1, \quad k = 1, 2, \dots, K_i, \quad i = 1, 2, \dots, n.$$

В аппроксимирующей задаче переменными являются  $t_i^k$  и  $y_i^k$ .

Данное представление свидетельствует о том, что в принципе любая задача сепарабельного программирования может быть решена методами частично-целочисленного программирования. Трудность такого подхода к решению задачи связана с быстрым ростом числа ограничений при увеличении количества точек разбиения. В частности, проблематичной является вычислительная реализация процедуры, поскольку нет эффективных компьютерных программ для решения задач частично-целочисленного программирования большой размерности.

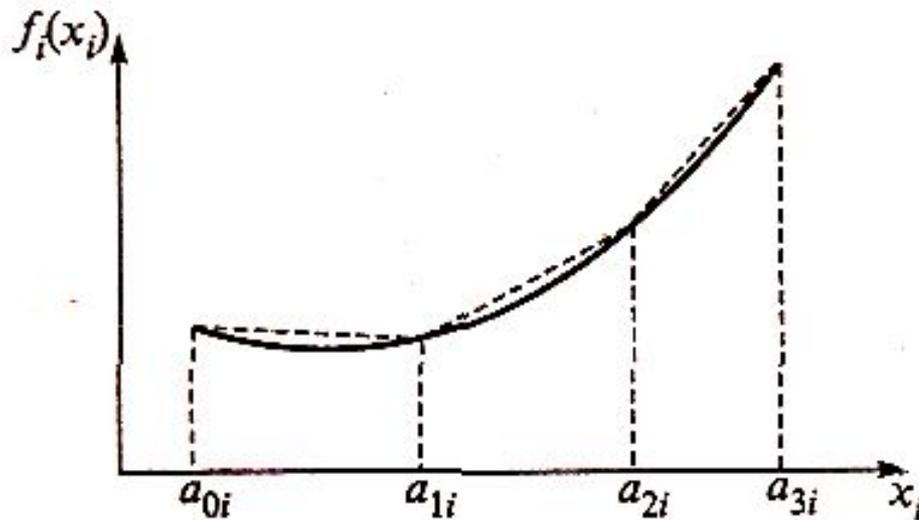
Для решения аппроксимирующей задачи можно использовать также обычный симплекс-метод, дополненный правилом ограниченного ввода в базис. В этом случае игнорируются вспомогательные ограничения, содержащие  $y_i^k$ . Согласно данному правилу, в базисе может находиться *не более двух положительных* весовых коэффициентов  $t_i^k$ . Более того, два коэффициента  $t_i^k$  могут быть положительными лишь тогда, когда они являются смежными. Таким образом, строгое условие оптимальности симплекс-метода *только тогда* используется для выбора переменной  $t_i^k$  подлежащей введению в число базисных, когда она удовлетворяет указанным выше требованиям

Метод частично-целочисленного программирования позволяет найти глобальный экстремум аппроксимирующей задачи, тогда как симплекс-метод с учетом правила ограниченного ввода в базис может гарантировать нахождение лишь локального оптимума. Кроме того, приближенное решение, полученное любым из двух упомянутых методов, может быть недопустимым для исходной задачи, поскольку при решении аппроксимирующей задачи могут быть обнаружены дополнительные экстремальные точки, которые для исходной задачи экстремальными не являются. Это зависит от точности линейной аппроксимации исходной задачи.

*Сепарабельное выпуклое программирование.* Задача выпуклого сепарабельного программирования имеет места в том случае, когда функции  $g_i^j(x_i)$  являются выпуклыми, обеспечивая, таким образом, выпуклость области допустимых решений. Кроме того, считаем, что функции  $f_i(x_i)$  являются выпуклыми в задаче минимизации и вогнутыми в задаче максимизации. При этих условиях можно использовать следующий упрощенный метод аппроксимации.

Рассмотрим задачу минимизации. Пусть функция  $f_i(x_i)$  имеет такой вид, как показано на рис. 9.3. Обозначим точки разбиения для функции  $f_i(x_i)$  через  $x_i = a_{ki}$ ,  $k = 0, 1, \dots, K_i$ .

Пусть  $x_{ki}$  - величина изменения переменной  $x_i$  на интервале  $(a_{k-1}, a_{ki})$ ,  $k = 1, 2, \dots, K_i$ , а  $\rho_{ki}$  - тангенс угла наклона линейного сегмента на том же интервале.



Тогда

$$f_i(x_i) \approx \sum_{k=1}^{K_i} \rho_{ki} x_{ki} + f_i(a_{0i}), \quad x_i = \sum_{k=1}^{K_i} x_{ki}.$$



Здесь предполагается, что  $0 \leq x_{ki} \leq a_{ki} - a_{k-1,i}$ ,  $k = 1, 2, \dots, K_i$ .

Так как функция  $f_i(x_i)$  выпуклая, то  $\rho_{1i} < \rho_{2i} < \rho_{k,i}$ . Это означает, что в задаче минимизации при  $p < q$  большее влияние на значение целевой функции оказывает переменная  $x_{pi}$  чем  $x_{qi}$ . Следовательно, переменная  $x_{pi}$  всегда будет входить в решение до того, как в него войдет  $x_{qi}$ . При этом значения каждой переменной  $x_{ki}$  должны быть ограничены сверху величиной  $(a_{ki} - a_{k-1,i})$ .

Выпуклые функции ограничений  $g_i^j(x_i)$  аппроксимируются аналогичным образом.

Пусть  $\rho_{ki}^j$  — тангенс угла наклона  $k$ -го линейного сегмента, соответствующего функции  $g_i^j(x_i)$ . Тогда функция  $g_i^j(x_i)$  аппроксимируется формулой

$$g_i^j(x_i) \approx \sum_{k=1}^{K_i} \rho_{ki}^j x_{ki} + g_i^j(a_{0i}).$$

Следовательно, рассматриваемая задача принимает вид

$$\text{Минимизировать } z = \sum_{i=1}^n \left( \sum_{k=1}^{K_i} \rho_{ki}^j x_{ki} + g_i^j(a_{0i}) \right),$$

при ограничениях

$$\sum_{i=1}^n \left( \sum_{k=1}^{K_i} \rho_{ki}^j x_{ki} + g_i^j(a_{0i}) \right) \leq b_j, j = 1, 2, \dots, m,$$

$$0 \leq x_{ki} \leq a_{ki} - a_{k-1,i}, k = 1, 2, \dots, K_i, i = 1, 2, \dots, n,$$

где

$$\rho_{ki} = \frac{f_i(a_{ki}) - f_i(a_{k-1,i})}{a_{ki} - a_{k-1,i}},$$

$$\rho_{ki}^j = \frac{g_i^j(a_{ki}) - g_i^j(a_{k-1,i})}{a_{ki} - a_{k-1,i}}.$$

Задача максимизации преобразуется аналогично. В этом случае,  $\rho_{1i} > \rho_{2i} > \rho_{k,i}$  откуда следует, что при  $p < q$  переменная всегда  $x_{pi}$  будет входить в решение раньше  $x_{qi}$ .

Полученную задачу можно решить симплекс-методом, предназначенным для решения задач с ограниченными сверху переменными. При этом исчезает необходимость в соблюдении правила ограниченного ввода в базис, поскольку выпуклость (вогнутость) функций гарантирует надлежащий выбор переменных.