

Прудников А.М.

ТЕКСТ ЛЕКЦИЙ ПО КУРСУ

Методы оптимизации

Версия: 0.1

Москва, 2012

Оглавление

1	Понятие математической оптимизации	4
1.1	Исследование операций и место методов оптимизации в данной дисциплине . .	4
1.2	Понятие оптимизации	5
1.2.1	Задача коммивояжера	5
1.2.2	Задача размещения производства	6
1.2.3	Построение математической модели	6
1.3	Общий вид однокритериальной оптимизационной задачи	7
1.4	Особенности задачи математического программирования	8
1.5	Классификация задач математического программирования	8
1.6	Классификация методов решения задач оптимизации	9
1.7	Условия окончания поиска	10
2	Нелинейное программирование	11
2.1	Безусловная оптимизация функций одной переменной	11
2.1.1	Пассивные методы поиска	12
2.1.2	Последовательные методы поиска	14
2.1.3	Сравнение эффективности методов, сокращающих ТИН	19
2.1.4	Методы полиномиальной аппроксимации	20
2.1.5	Методы, использующие производные	22
2.2	Безусловная оптимизация функций многих переменных	24
2.2.1	Основные понятия и определения	24
2.2.2	Простейшие поисковые методы	26
2.2.3	Симплексные методы	28
2.2.4	Градиентные методы	31
2.2.5	Пример использования	34
3	Линейное программирование	38
3.1	Постановка задачи и основные определения	38
3.2	Переход от общей к основной задаче линейного программирования	38
3.3	Существование допустимых решений ОЗЛП	39
3.4	Геометрическая интерпретация ОЗЛП	41
3.5	Пример решения ОЗЛП с помощью геометрической интерпретации	42

3.6	Некоторые свойства решений ОЗЛП. Особые случаи решения ОЗЛП	43
3.7	Признак оптимальности опорного решения	44
3.8	Симплекс-метод решения задач линейного программирования	45
3.8.1	Отыскание опорного решения	45
4	Комбинаторная оптимизация	49
5	Стохастическая оптимизация	50
	Литература	51

Глава 1

Понятие математической оптимизации

1.1 Исследование операций и место методов оптимизации в данной дисциплине

Исследование операций – математическая дисциплина, занимающаяся построением, разработкой и применением математических моделей принятия оптимальных решений во всех областях человеческой деятельности.

Операция – это всякое мероприятие (система действий), объединенное единым замыслом и направленное к достижению какой-то цели.

Пусть необходимо выполнить какое-либо мероприятие для достижения определенной цели (**операцию**). Обычно существует свобода выбора в том, как это мероприятие организовать (например, выбор техники, распределение ресурсов и т.д.). **«Решение»** – это какой-то выбор из ряда допустимых возможностей.

Краеугольным камнем исследования операций является **математическое моделирование**. Данные, полученные при исследовании математических моделей, являются основой для принятия решений. Но общих способов построения математических моделей и методов их решения не существует. В каждом конкретном случае модель выбирается исходя из вида операции, ее целевой направленности, с учетом задачи исследования.

Наиболее известными методами исследования операций (являющимися, зачастую, самостоятельными математическими дисциплинами), являются:

- Математическое программирование – теория и методы решения задач о нахождении экстремумов функций на множествах векторного пространства, определяемых линейными и нелинейными ограничениями (равенствами и неравенствами). (Присутствие в названии термина «программирование» объясняется тем, что первые исследования оптимизационных задач были в сфере экономики, а в английском языке слово «programming» означает планирование, составление планов или программ).
- Сетевые модели – решение оптимизационных задач с использованием графов.
- Марковские процессы – метод решения стохастических задач, где процесс принятия

решений можно представить конечным числом состояний.

- Теория игр – методы изучения оптимальных стратегий в играх. Теория игр помогает выбрать лучшие стратегии с учётом представлений о других участниках, их ресурсах и их возможных поступках.
- Теория массового обслуживания (теория очередей) – раздел теории вероятностей, целью исследований которого является рациональный выбор структуры системы обслуживания и процесса обслуживания на основе изучения потоков требований на обслуживание, длительности ожидания и длины очередей.
- Имитационное моделирование – метод исследования, при котором изучаемая система заменяется компьютерной моделью, с достаточной точностью описывающей реальную систему, и с ней проводятся эксперименты с целью получения информации об этой системе. Экспериментирование с моделью называют имитацией.

В данном курсе будет изучаться математическое программирование, методы которого и являются так называемыми **методами оптимизации**.

1.2 Понятие оптимизации

Оптимизация в самом широком смысле - это выбор наилучшего варианта из множества возможных. Рассмотрим некоторые классические примеры оптимизационных задач.

1.2.1 Задача коммивояжера

Дано некоторое количество городов и расстояния между ними. Коммивояжер должен посетить каждый город и вернуться к месту отправления. Какой маршрут он должен выбрать? Зададим расстояния между городами (допустим, их пять) в виде следующей таблицы:

	1	2	3	4	5
1	—	1	7	2	8
2	2	—	10	3	1
3	7	10	—	2	6
4	2	3	2	—	4
5	8	1	6	4	—

Проложим несколько маршрутов и посчитаем расстояние для них:

- Маршрут $5 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$, расстояние $8 + 1 + 10 + 2 + 4 = 25$
- Маршрут $5 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 3 \rightarrow 5$, расстояние $1 + 1 + 2 + 2 + 6 = 12$

Каким же образом (исключая полный перебор) следует выбрать кратчайший маршрут?

1.2.2 Задача размещения производства

Пусть в некотором регионе имеется ряд потребителей некоторой продукции. Нужно определить, как разместить в этом регионе заводы по производству данной продукции.

Существует два крайних решения:

1. Можно разместить только один завод; в этом случае производственные затраты будут минимальны, но станут максимальными затраты на доставку продукции потребителям (транспортные затраты).
2. Можно разместить большое количество заводов (рядом с каждым потребителем); в этом случае транспортные затраты будут минимальны, но станут максимальными производственные затраты.

Очевидно, что оптимальное решение заключается в минимизации суммарных затрат $C_{\text{общ}} = C_{\text{произв}} + C_{\text{трансп}} \rightarrow \min$.

1.2.3 Построение математической модели

Как было сказано выше, для решения любых задач исследования операций (и, соответственно, задач математического программирования) необходимо формализовать решаемую задачу, построив ее математическую модель. В самых общих чертах процесс построения математической модели можно представить следующим образом:

1. Определение неизвестных параметров (элементов решения).
2. Выражение условий задачи через введенные на первом шаге неизвестные.
3. Выбор критерия оптимальности.

Попробуем выполнить описанные выше шаги для построения математической модели задачи размещения производства (см. раздел 1.2.2).

На шаге **определения неизвестных параметров** введем переменные x_{ij} - объем перевозимой продукции с i -го завода j -му потребителю; здесь $i = \overline{1, m}$ - количество заводов, $j = \overline{1, n}$ - количество потребителей.

Теперь **выразим условия задачи** через эти переменные:

1. Очевидно, что количество перевозимой продукции не может быть отрицательно: $x_{ij} \geq 0$.
2. Если обозначить известный нам объем заказов (потребления) продукции j -м потребителем через b_j , то на объем перевозок можно наложить ограничение $\sum_{i=1}^m x_{ij} = b_j$ (не следует везти к потребителю продукции больше, чем он заказывает).
3. Если обозначить известные нам удельные транспортные затраты на перевозку изделий от i -го завода j -му потребителю через c_{ij} , то можно составить уравнение для транспортных затрат: $C_{\text{трансп}} = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$.

4. Если обозначить известную нам стоимость производства на i -ом заводе (которая, вообще говоря, зависит от объемов производства) через $f_i(x_i)$, то можно составить уравнение для производственных затрат: $C_{\text{произв}} = \sum_{i=1}^m f_i(\sum_{j=1}^n x_{ij})$.

Осталось **выбрать критерий оптимальности**, однако это мы уже сделали в п. 1.2.2: суммарные затраты должны быть минимальны.

Таким образом, в результате формализации задачи мы построили ее математическую оптимизационную модель следующего вида:

$$\min \left\{ \sum_{i=1}^m f_i\left(\sum_{j=1}^n x_{ij}\right) + \sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij} \right\} \quad (1.1)$$

при ограничениях

$$\begin{cases} \sum_{i=1}^m x_{ij} = b_j \\ x_{ij} \geq 0 \end{cases} \quad i = \overline{1, m}, j = \overline{1, n} \quad (1.2)$$

Опираясь на полученный результат, можно попробовать записать постановку оптимизационной задачи в общем виде.

1.3 Общий вид однокритериальной оптимизационной задачи

В общем виде задача математического программирования ставится следующим образом: найти максимум (минимум) функции

$$f(x_1, x_2, \dots, x_n) = f(\bar{x}) \quad (1.3)$$

при ограничениях

$$\begin{cases} g_i(x_1, x_2, \dots, x_n) \leq b_i, i = \overline{1, m} \\ x_j \geq 0, j = \overline{1, n} \end{cases} \quad (1.4)$$

Здесь:

- $f(\bar{x})$ - целевая функция;
- система неравенств и условия неотрицательности переменных (1.4) - система ограничений.

Всякое решение задачи с учетом системы ограничений называется **допустимым решением**. Допустимое решение, максимизирующее (минимизирующее) целевую функцию, называется **оптимальным решением**. Таким образом, задача математического программирования заключается в нахождении оптимального решения, которое по определению обеспечивает максимальное (минимальное) значение целевой функции с учетом заданных ограничений.

1.4 Особенности задачи математического программирования

1. Если требуется найти минимум $f(\bar{x})$, то это эквивалентно поиску максимума $-f(\bar{x})$.
2. В любом случае можно добиться условия неотрицательности переменных, то есть, если задано ограничение $x_j \geq x_{j_min}$, то можно сделать замену переменных $x'_j = x_j - x_{j_min} \geq 0$.
3. Если заданы ограничения вида $g_i(\bar{x}) \geq b_i$, то простой заменой знака приходим к первоначальной форме $-g_i(\bar{x}) \leq -b_i$.
4. Функция $f(\bar{x})$ может иметь несколько экстремумов, а именно локальные экстремумы и глобальный экстремум. Функция $f(\bar{x})$, определенная на области D , достигает на ней глобального максимума $\bar{x}^* \in D$, если неравенство $f(\bar{x}) \leq f(\bar{x}^*)$ справедливо для любой точки $\bar{x} \in D$. Функция $f(\bar{x})$, определенная на области D , достигает на ней локального максимума $\bar{x}^* \in D$, если неравенство $f(\bar{x}) \leq f(\bar{x}^*)$ справедливо для точек из некоторой окрестности \bar{x}^* .
5. В математическом анализе для нахождения экстремумов функций используются производные (это классические методы оптимизации). Такие методы применяют лишь для сравнительно простых задач из-за следующих недостатков:
 - для использования таких методов нужно, чтобы функции $f(\bar{x})$ и $g_i(\bar{x})$ были непрерывны и имели частные производные по крайней мере до 2-го порядка;
 - с помощью классических методов можно найти экстремум только внутри области; если оптимальная точка находится на границе области, то эти методы бессильны;
 - на переменные x_j не должны быть наложены условия целочисленности.

1.5 Классификация задач математического программирования

В зависимости от вида функций, входящих в критерий оптимальности и систему ограничений, а также допустимой области изменения переменных, задачи математического программирования разделяются на следующие классы:

1. Линейное программирование - целевая функция и ограничения являются линейными. Область допустимых значений - многогранник, а оптимальное решение находится в одной из его вершин.
2. Нелинейное программирование - или целевая функция, или какое-либо ограничение содержит нелинейную зависимость.

3. Дискретное программирование - переменные могут принимать только целочисленные значения.

1.6 Классификация методов решения задач оптимизации

Особенность задач оптимизации состоит в том, что вычисление значений целевой функции и значений ограничивающих функций может требовать больших затрат времени. В связи с этим возникает проблема решения задач оптимизации при наименьшем числе испытаний.

Испытанием называется операция однократного вычисления функций $f(\bar{x})$ и $g_i(\bar{x})$ (и, в некоторых случаях, их производных) в некоторой точке \bar{x} . Далее будем говорить, что задача оптимизации решается с помощью **поискового метода оптимизации**, если используется следующая процедура поиска оптимального решения \bar{x}^* :

- по очереди при $r = 0, 1, 2, \dots, N - 1$ производятся испытания в точках

$$\bar{x}^{r+1} = \Psi_{r+1}(\bar{x}^0, f(\bar{x}^0), g_i(\bar{x}^0), \dots, \bar{x}^r, f(\bar{x}^r), g_i(\bar{x}^r)) \quad (1.5)$$

- в качестве решения задачи берется точка \bar{x}^* , которая находится из условия $f(\bar{x}^*) = \min_{r \in [0; N]} f(\bar{x}^r)$.

Здесь:

- r - текущий номер испытания;
- N - число испытаний;
- \bar{x}^0 - начальное приближение;
- Ψ_r - алгоритм поисковой оптимизации на r -ом шаге.

В общем случае **алгоритмом поисковой оптимизации** называется способ выбора начального приближения \bar{x}^0 и конкретная совокупность функций $\{\Psi_r\}$. Таким образом, понятие алгоритма является более частным по сравнению с понятием метода (одному и тому же методу могут соответствовать разные алгоритмы).

Теперь проведем классификацию методов решения с учетом введенным понятием.

1. **Классификация по наличию или отсутствию системы ограничений.** Если в задаче отсутствует система ограничений, то она решается методами **безусловной** оптимизации; в противном случае - методами **условной** оптимизации.
2. **Классификация по размерности вектора \bar{x} .** Если \bar{x} на самом деле скаляр, то применяются **одномерные** методы оптимизации; в противном случае - **многомерные**.

3. **Классификация по характеру искомого решения.** Если метод поиска гарантирует отыскание только локального экстремума, то это метод **локальной** оптимизации. Если делается попытка отыскать глобальный экстремум, то это метод **глобальной** оптимизации. Следует отметить, что удовлетворительных с точки зрения вычислительной эффективности методов глобальной оптимизации не существует.
4. **Классификация по характеру функций Ψ_r .** Если функции Ψ_r являются детерминированными, то метод оптимизации называется **детерминированным**. Если же функции Ψ_r содержат случайные параметры, то метод оптимизации называется **стохастическим**.
5. **Классификация по способу выбора точек \bar{x}^r .** Если все точки \bar{x}^r назначаются заранее (до проведения испытаний), то метод оптимизации называется **пассивным**. Если же очередная точка \bar{x}^{r+1} определяется на основе всей или части информации об испытаниях в точках $\bar{x}^0, \dots, \bar{x}^r$, то метод называется **последовательным**.
6. **Классификация по количеству предыдущих учитываемых шагов.** Если в последовательном методе при определении точки \bar{x}^{r+1} учитывается информация только о предыдущем испытании, то метод называется **одношаговым**. Если же используется информация о $s > 1$ предыдущих испытаниях, то метод называется многошаговым (конкретнее, s -шаговым).
7. **Классификация по виду функций Ψ_r .** Если функция Ψ_r при всех N испытаниях одинакова, то метод называется **итерационным**. Если же функции Ψ_r меняются от испытания к испытанию, то метод является **неитерационным**.
8. **Классификация по порядку используемых производных.** Если при вычислении значений функций Ψ_r производные не используются, то метод называется **прямым** (или **нулевого порядка**). Если же используются производные k -го порядка, то метод называется методом **k -го порядка** (методы 1-го порядка также называются **градиентными**).

1.7 Условия окончания поиска

Выбор условия (критерия) окончания поиска является еще одной важной проблемой при решении оптимизационных задач. Наиболее широко используемыми являются следующие критерии:

- $\|\bar{x}^{r+1} - \bar{x}^r\| \leq \epsilon_x$, где ϵ_x - требуемая точность решения по \bar{x} , $\|\cdot\|$ - некоторая векторная норма (например, евклидова);
- $|f(\bar{x}^{r+1}) - f(\bar{x}^r)| \leq \epsilon_f$, где ϵ_f - требуемая точность решения по f .

Глава 2

Нелинейное программирование

2.1 Безусловная оптимизация функций одной переменной

Оптимизация функций одной переменной является, как правило, необходимым элементом методов оптимизации функций многих переменных. На первый взгляд кажется, что эта задача достаточно проста и решается с помощью дифференцирования (классический метод оптимизации). Однако для широкого класса функций это не так, поскольку задача решения уравнения $f'(x) = 0$ может оказаться весьма сложной (или даже невозможной, если $f(x)$ не дифференцируема).

Как уже было упомянуто выше, существование локальных экстремумов функции почти всегда затрудняет поиск глобального экстремума. Поэтому многие методы оптимизации применимы только тогда, когда любой локальный экстремум является одновременно и глобальным; это дает гарантию сходимости метода. Если же таких сведений о функции нет, то методы применять можно, но без гарантии сходимости.

Одним из классов функций, удовлетворяющих указанному условию, является класс **уни-модальных (одноэкстремальных)** функций. Дадим определение такой функции для задачи поиска минимума (для задачи поиска максимума определение строится аналогичным образом).

Функция $f(x)$ называется **унимодальной** на отрезке $[a, b]$, если она непрерывна¹ на $[a, b]$ и существуют такие α и β ($a \leq \alpha \leq \beta \leq b$), что:

1. на отрезке $[a, \alpha]$ при $a < \alpha$ $f(x)$ монотонно убывает;
2. на отрезке $[\beta, b]$ при $\beta < b$ $f(x)$ монотонно возрастает;
3. существует минимум $f(x)$ при $x \in [\alpha, \beta]$.

Примеры унимодальных функций приведены на рисунке 2.1.

¹В общем случае это не так, но мы под унимодальной функцией будем подразумевать непрерывную унимодальную функцию.

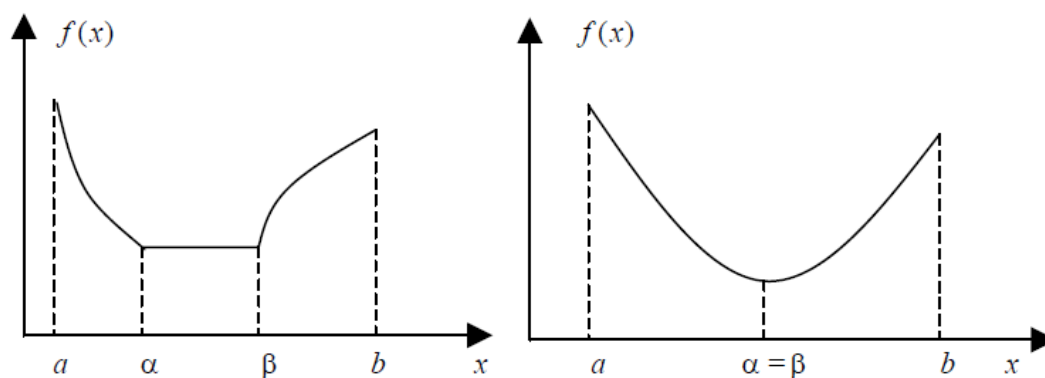


Рис. 2.1. Примеры унимодальных функций

Прежде чем приступить к процедуре оптимизации, следует по возможности установить принадлежность целевой функции классу, для которого гарантирована сходимость процесса.

Заметим, что предположение об унимодальности функции в окрестности точки экстремума весьма естественно. Получение информации о таком промежутке является важным предварительным этапом процедуры оптимизации.

Рассмотрим с общих позиций ряд методов (разделы 2.1.1 и 2.1.2), позволяющих находить экстремумы (далее при рассмотрении алгоритмов будем говорить о задаче поиска минимума) унимодальных функций на отрезке $[a, b]$. Эти методы объединяет идея сокращения **текущего интервала неопределенности (ТИН)**. Она состоит в том, что в процессе поиска исключаются из рассмотрения те подынтервалы, на которых точка оптимума x^* отсутствует в силу унимодальности целевой функции.

2.1.1 Пассивные методы поиска

На практике подобные методы применяются, когда удобно провести независимые эксперименты по измерению значений функции $f(x)$, а последовательное измерение этих значений трудоемко или невозможно по каким-либо причинам.

Метод равномерного поиска

Испытания проводятся в точках, которые определяются путем равномерного деления отрезка $[a, b]$ на N одинаковых интервалов. Из вычисленных в узлах полученной сетки значений целевой функции выбирается наименьшее (пусть это значение достигается в некотором узле x_k). Тогда, в связи с унимодальностью целевой функции, подынтервалы $[a, x_{k-1}]$ и $[x_{k+1}, b]$ можно исключить из рассмотрения, то есть сделать очередным ТИН отрезок $[x_{k-1}, x_{k+1}]$.

Алгоритм метода:

1. Выполняем присваивание $r = 1, a^1 = a, b^1 = b, \text{ТИН}_1 = [a^1, b^1]$.
2. На очередном ТИН строим равномерную сетку с $N + 1$ узлом.
3. Вычисляем значения целевой функции в узлах сетки.
4. Находим минимальное из вычисленных значений $\min(f(x_0^r), f(x_1^r), \dots, f(x_N^r)) = f(x_k^r)$.

5. Выполняем присваивание $a^{r+1} = x_{k-1}^r, b^{r+1} = x_{k+1}^r, \text{ТИН}_{r+1} = [a^{r+1}, b^{r+1}]$.
6. Если $|\text{ТИН}_{r+1}| \leq \epsilon_x$, то заканчиваем вычисления; иначе выполняем присваивание $r = r + 1$ и переходим к шагу 2.

В качестве приближенного значения точки минимума x^* может быть принята любая точка последнего ТИН.

На рисунке 2.2 показан один шаг метода равномерного поиска при $N = 13$.

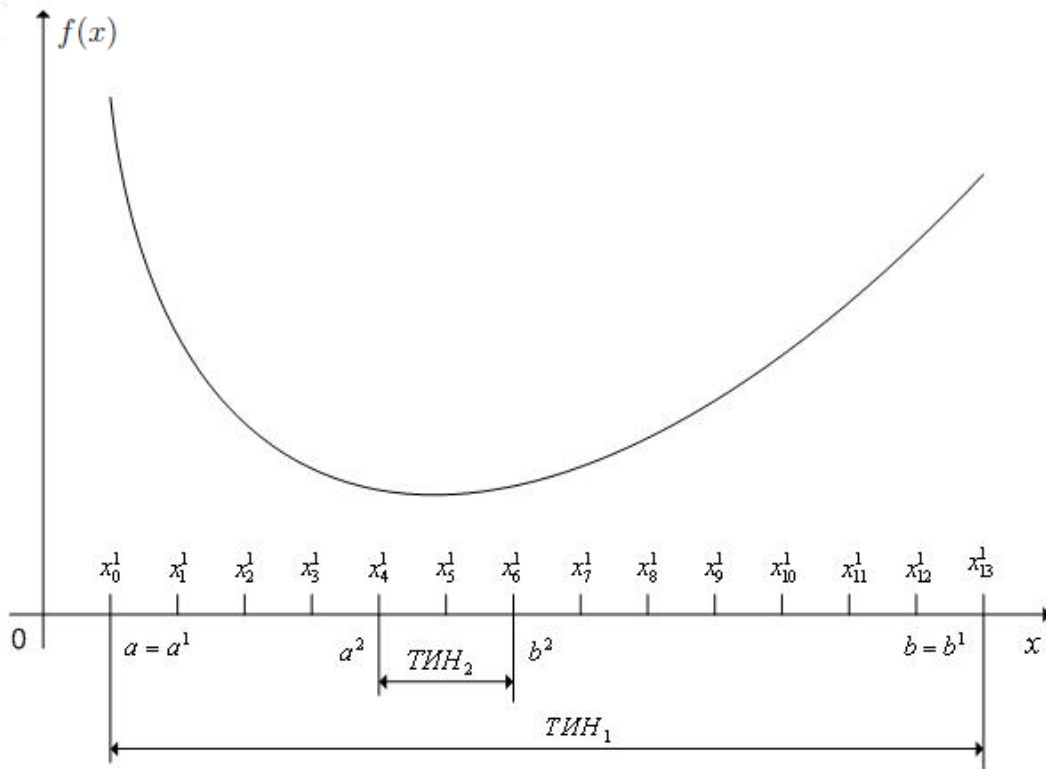


Рис. 2.2. Метод равномерного поиска

Поскольку после каждой итерации длина ТИН уменьшается в фиксированное ($\frac{N}{2}$) количество раз, можно априорно оценить количество итераций по заданной точности решения. Действительно, после первой итерации $|\text{ТИН}_2| = 2^{\frac{b-a}{N}}$, после второй итерации $|\text{ТИН}_3| = 2((2^{\frac{b-a}{N}})/N) = (b-a)(\frac{2}{N})^2$ и т.д. Тогда после r -ой итерации имеем $|\text{ТИН}_{r+1}| = (b-a)(\frac{2}{N})^r$. Но, так как условие окончания поиска $|\text{ТИН}_{r+1}| \leq \epsilon_x$, получаем $(b-a)(\frac{2}{N})^r \leq \epsilon_x$. Заменив знак неравенства на равенство и выразив r , получим оценку количества итераций.

Метод поразрядного поиска

Можно усовершенствовать метод равномерного поиска, уменьшив количество вычислений значений целевой функции на каждой итерации. Во-первых, если $f(x_i^r) < f(x_{i+1}^r)$, то отпадает необходимость вычислять $f(x)$ в точках x_{i+2}^r, x_{i+3}^r и т.д. Во-вторых, разумно было бы сначала грубо определить отрезок, содержащий оптимальную точку; а затем на этом отрезке искать ее с меньшим шагом дискретизации. Иными словами, следует сделать шаг сетки не постоянным, а зависящим от итерации.

В методе поразрядного поиска перебор точек происходит с некоторым шагом h_r до тех пор, пока не выполнится условие $f(x_i^r) < f(x_{i+1}^r)$, или пока очередная из точек не совпадет с концом отрезка. После этого шаг уменьшается (обычно в 4 раза) и перебор точек с новым шагом осуществляется в противоположном направлении до тех пор, пока значения $f(x)$ снова не перестанут уменьшаться или не будет достигнут противоположный край отрезка и т.д. Описанный процесс завершается, когда перебор в каком-либо направлении закончен, а длина шага $|h_r| \leq \epsilon_x$.

Алгоритм метода:

1. Выполняем присваивание $r = 1, i = 0, x_i^r = a, h_r = \frac{b-a}{4}$.
2. Делаем очередной шаг $x_{i+1}^r = x_i^r + h_r$; проверяем, что $x_{i+1}^r \in [a, b]$: если принадлежит, то переходим к шагу 3, в противном случае - к шагу 4.
3. Вычисляем и сравниваем $f(x_i^r)$ и $f(x_{i+1}^r)$. Если $f(x_i^r) \geq f(x_{i+1}^r)$, то положить $i = i + 1$ и перейти к шагу 2, в противном случае - к шагу 4.
4. Полагаем $x^* = x_i^r$ и проверяем условие окончания поиска: если $|h_r| \leq \epsilon_x$, то завершаем вычисления; в противном случае переходим к шагу 5.
5. Переходим к следующей итерации: изменяем направление поиска и уменьшаем шаг. Для этого выполняем присваивание $r = r + 1, i = 0, x_i^r = x^*, h_r = \frac{h_{r-1}}{4}$ и переходим к шагу 2.

2.1.2 Последовательные методы поиска

Последовательные методы поиска на практике используются чаще, чем пассивные. Это обусловлено тем, что использование информации о результатах предыдущих измерений для выбора очередной экспериментальной точки x_i как правило приводит к более эффективному поиску.

Метод дихотомии

В методе дихотомии испытания проводятся парами. Точки каждой последующей пары разнесены между собой на величину $\delta_x < \epsilon_x$. Испытания производятся в середине ТИН. По значениям $f(x)$, полученным в этих точках, одна половина ТИН в силу унимодальности целевой функции исключается из дальнейшего рассмотрения.

Алгоритм метода:

1. Выполняем присваивание $r = 1, a^1 = a, b^1 = b, \text{ТИН}_1 = [a^1, b^1]$.
2. Вычисляем величины $x_0^r = \frac{b^r - a^r}{2}, x_1^r = x_0^r - \frac{\delta_x}{2}, x_2^r = x_0^r + \frac{\delta_x}{2}$.
3. Вычисляем значения $f(x_1^r)$ и $f(x_2^r)$.

4. Если $f(x_1^r) < f(x_2^r)$, то выполняем присваивание $a^{r+1} = a^r$, $b^{r+1} = x_0^r$; в противном случае $a^{r+1} = x_0^r$, $b^{r+1} = b^r$. В обоих случаях $\text{ТИН}_{r+1} = [a^{r+1}, b^{r+1}]$.
5. Если $|\text{ТИН}_{r+1}| \leq \epsilon_x$, то заканчиваем вычисления; иначе выполняем присваивание $r = r + 1$ и переходим к шагу 2.

В качестве приближенного значения точки минимума x^* может быть принята любая точка последнего ТИН.

На рисунке 2.3 показан один шаг метода дихотомии.

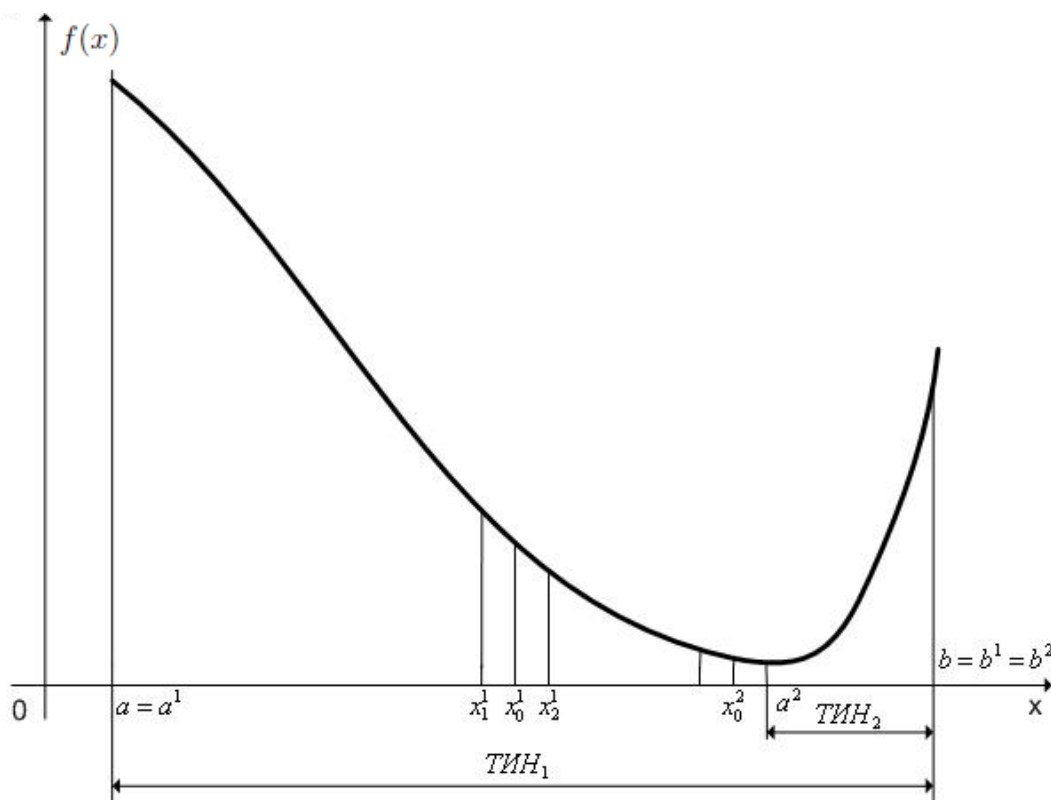


Рис. 2.3. Метод дихотомии

Поскольку после каждой итерации длина ТИН уменьшается в 2 раза, можно априорно оценить количество итераций по заданной точности решения. Действительно, после первой итерации $|\text{ТИН}_2| = \frac{b-a}{2}$, после второй итерации $|\text{ТИН}_3| = ((\frac{b-a}{2})/2) = \frac{b-a}{2^2}$ и т.д. Тогда после r -ой итерации имеем $|\text{ТИН}_{r+1}| = \frac{b-a}{2^r}$. Но, так как условие окончания поиска $|\text{ТИН}_{r+1}| \leq \epsilon_x$, получаем $\frac{b-a}{2^r} \leq \epsilon_x$. Заменяя знак неравенства на равенство и выразив r , получим оценку количества итераций.

Замечание 1. Если производная целевой функции считается достаточно просто, можно видоизменить метод дихотомии. На концах рассматриваемого отрезка вычисляют производные целевой функции, отрезок делят пополам и вычисляют производную в средней точке. Для следующей итерации выбирают тот отрезок из двух получившихся, на концах которого знаки производной различны (по определению унимодальной функции ее производная меняет знак на отрезке поиска только один раз).

Замечание 2. В методе дихотомии на каждой итерации значение функции вычисляется дважды. Однако, как было указано в разделе 1.6, в общем случае стараются уменьшить

количество таких вычислений. Поэтому более эффективными являются методы, где новые «экспериментальные» точки на каждой (по возможности) итерации выбираются таким образом, чтобы значение функции приходилось вычислять только один раз.

Метод Фибоначчи

Идея метода Фибоначчи состоит в том, чтобы определять новые «экспериментальные» точки с помощью чисел Фибоначчи, поэтому сначала следует ввести определение и рассмотреть свойства этих чисел. Числа Фибоначчи задаются рекуррентным соотношением $F_i = F_{i-1} + F_{i-2}$, $i \geq 2$, $F_0 = F_1 = 1$. Вычислять числа Фибоначчи нерекуррентным образом можно с помощью выражения

$$F_i = \frac{\left(\frac{1}{\tau}\right)^{i+1} - (-\tau)^{i+1}}{\sqrt{5}}, \quad (2.1)$$

где $\tau = \frac{\sqrt{5}-1}{2} \approx 0,618$ - решение квадратного уравнения

$$\tau^2 + \tau - 1 = 0 \quad (2.2)$$

При больших значениях i членом $(-\tau)^{i+1}$ можно пренебречь. Тогда

$$F_i \approx \frac{\left(\frac{1}{\tau}\right)^{i+1}}{\sqrt{5}} \quad (2.3)$$

Можно заметить, что из (2.3) следует $F_{i-1} \approx \frac{\left(\frac{1}{\tau}\right)^i}{\sqrt{5}}$ и, следовательно,

$$\frac{F_{i-1}}{F_i} \approx \tau, \quad (2.4)$$

то есть при больших i отношение двух соседних чисел Фибоначчи примерно постоянно.

Алгоритм метода состоит из двух этапов.

Первый этап состоит из $(N-1)$ -ой итерации для $r = 1, 2, \dots, N-1$. Рассмотрим схему r -ой итерации, когда $\text{ТИН}_r = [a^r, b^r]$:

1. Вычисляем величины $x_1^r = a^r + |\text{ТИН}_r| \frac{F_{N-r-1}}{F_{N-r+1}}$, $x_2^r = a^r + |\text{ТИН}_r| \frac{F_{N-r}}{F_{N-r+1}}$.
2. Вычисляем значения $f(x_1^r)$ и $f(x_2^r)$.
3. Если $f(x_1^r) < f(x_2^r)$, то выполняем присваивание $a^{r+1} = a^r$, $b^{r+1} = x_2^r$; в противном случае $a^{r+1} = x_1^r$, $b^{r+1} = b^r$. В обоих случаях $\text{ТИН}_{r+1} = [a^{r+1}, b^{r+1}]$.

Данный этап алгоритма обладает тем свойством, что после выполнения $(N-1)$ -ой итерации имеет место следующая ситуация: $x_1^{N-1} = x_2^{N-1} = x^{N-1}$. Таким образом, на $(N-1)$ -ой итерации сужения ТИН не происходит: $\text{ТИН}_N = [a^{N-1}, b^{N-1}] = \text{ТИН}_{N-1}$, причем точка x^{N-1} оказывается в середине ТИН_{N-1} .

Второй этап позволяет определить, с какой стороны от точки x^{N-1} лежит точка минимума. Для этого:

1. Находим точку $x^N = x^{N-1} + \delta_x$, где $\delta_x \ll |\text{ТИН}_{N-1}|$ - свободный параметр алгоритма.
2. Вычисляем значение $f(x^N)$.
3. Если $f(x^N) > f(x^{N-1})$, то выполняем присваивание $\text{ТИН}_{N+1} = [a^{N-1}, x^{N-1}]$; в противном случае $\text{ТИН}_{N+1} = [x^{N-1}, b^{N-1}]$.

В качестве приближенного значения точки минимума x^* может быть принята любая точка последнего ТИН.

Некоторые **свойства метода Фибоначчи**.

Утверждение 1. Для любого $r = 1, 2, \dots, N-2$ метод Фибоначчи обладает следующим свойством: одна из точек x_1^{r+1} , x_2^{r+1} совпадает с одной из точек x_1^r , x_2^r .

◀ Пусть на r -ой итерации выполняется $f(x_1^r) < f(x_2^r)$, тогда $\text{ТИН}_{r+1} = [a^r, x_2^r]$ и $x_1^r \in \text{ТИН}_{r+1}$. Рассмотрим точку $x_2^{r+1} = a^r + |\text{ТИН}_{r+1}| \frac{F_{N-(r+1)+1}}{F_{N-(r+1)+1}} = a^r + (x_2^r - a^r) \frac{F_{N-r-1}}{F_{N-r}}$. Так как $x_2^r = a^r + |\text{ТИН}_r| \frac{F_{N-r}}{F_{N-r+1}}$, то $x_2^{r+1} = a^r + \left(a^r + |\text{ТИН}_r| \frac{F_{N-r}}{F_{N-r+1}} - a^r\right) \frac{F_{N-r-1}}{F_{N-r}} = a^r + |\text{ТИН}_r| \frac{F_{N-r-1}}{F_{N-r+1}} = x_1^r$ ▶

Доказательство для случая $f(x_1^r) > f(x_2^r)$ проводится аналогично. Указанное свойство позволяет на каждой итерации (кроме первой) производить испытания только в одной точке.

Утверждение 2. Точки x_1^r , x_2^r расположены симметрично относительно концов ТИН.

◀ В соответствии с алгоритмом имеем: $x_1^r - a^r = a^r + (b^r - a^r) \frac{F_{N-r-1}}{F_{N-r+1}} - a^r = (b^r - a^r) \frac{F_{N-r-1}}{F_{N-r+1}}$; $b^r - x_2^r = b^r - a^r - (b^r - a^r) \frac{F_{N-r}}{F_{N-r+1}} = (b^r - a^r) \left(1 - \frac{F_{N-r}}{F_{N-r+1}}\right) = (b^r - a^r) \left(\frac{F_{N-r+1}-F_{N-r}}{F_{N-r+1}}\right)$. Так как из определения чисел Фибоначчи следует, что $F_{N-r} = F_{N-r+1} - F_{N-r-1}$, второе выражение можно записать как $b^r - x_2^r = (b^r - a^r) \frac{F_{N-r-1}}{F_{N-r+1}}$. Получили, что $x_1^r - a^r = b^r - x_2^r$ ▶

Утверждение 3. В результате любой итерации $r = 1, 2, \dots, N-2$ длина ТИН уменьшается в $\frac{F_{N-r}}{F_{N-r+1}}$ раз.

◀ Из утверждения 2 следует, что $x_2^r - a^r = b^r - x_1^r$, поэтому достаточно рассмотреть только один из отрезков $[a^r, x_2^r]$, $[x_1^r, b^r]$. Рассмотрим первый отрезок: $x_2^r - a^r = a^r + (b^r - a^r) \frac{F_{N-r}}{F_{N-r+1}} - a^r = (b^r - a^r) \frac{F_{N-r}}{F_{N-r+1}}$ ▶

Утверждение 4. Из утверждения 3 и (2.4) следует, что при достаточно больших N в результате одной итерации длина ТИН уменьшается в τ раз.

Утверждение 5. В результате N итераций длина ТИН становится равной $\frac{b-a}{F_N}$.

◀ Действительно, после первой итерации $|\text{ТИН}_2| = (b-a) \frac{F_{N-1}}{F_N}$, после второй итерации $|\text{ТИН}_3| = (b-a) \frac{F_{N-1}}{F_N} \frac{F_{N-2}}{F_{N-1}} = (b-a) \frac{F_{N-2}}{F_N}$ и т.д.; после $(N-2)$ -ой итерации имеем $|\text{ТИН}_{N-1}| = (b-a) \frac{F_2}{F_N} = (b-a) \frac{2}{F_N}$, после $(N-1)$ -ой итерации длина ТИН не меняется, после N -ой итерации длина ТИН уменьшается ровно в два раза (так как из утверждения 2 и выражения $x_1^{N-1} = x_2^{N-1} = x^{N-1}$ следует, что x^{N-1} делит ТИН пополам): $|\text{ТИН}_{N+1}| = (b-a) \frac{2}{F_N} / 2 = \frac{b-a}{F_N}$ ▶

Метод Фибоначчи является достаточно эффективным (см. раздел 2.1.3) методом одномерной оптимизации, однако в нем доставляет неудобство тот факт, что после выбора значения параметра N следует обязательно сделать $N-1$ итераций, без возможности остановки даже в том случае, когда длина очередного ТИН стала меньше заданной точности решения. От

этого недостатка можно избавиться следующим образом: из (2.4) следует, что отношение соседних чисел Фибоначчи примерно постоянно, следовательно, на каждой итерации можно выбирать экспериментальные точки так, чтобы они делили ТИН в одном и том же отношении τ , которое не зависит от N . Эти рассуждения ложатся в основу метода золотого сечения.

Метод золотого сечения

Метод золотого сечения является почти таким же эффективным, как метод Фибоначчи, но позволяет остановить вычисления на любой итерации. Определение золотого сечения дается следующим образом: говорят, что точка c выполняет золотое сечение отрезка $[a, b]$, если $\frac{c-a}{b-a} = \tau$.

Из определения золотого сечения следует, что $\frac{b-c}{b-a} = \frac{(b-a)-(c-a)}{b-a} = 1 - \frac{c-a}{b-a} = 1 - \tau$.

Алгоритм метода:

1. Выполняем присваивание $r = 1, a^1 = a, b^1 = b, \text{ТИН}_1 = [a^1, b^1]$.
2. Вычисляем величины $x_1^r = b^r - (b^r - a^r) \tau, x_2^r = a^r + (b^r - a^r) \tau$.
3. Вычисляем значения $f(x_1^r)$ и $f(x_2^r)$.
4. Если $f(x_1^r) < f(x_2^r)$, то выполняем присваивание $a^{r+1} = a^r, b^{r+1} = x_2^r$; в противном случае $a^{r+1} = x_1^r, b^{r+1} = b^r$. В обоих случаях $\text{ТИН}_{r+1} = [a^{r+1}, b^{r+1}]$.
5. Если $|\text{ТИН}_{r+1}| \leq \epsilon_x$, то заканчиваем вычисления; иначе выполняем присваивание $r = r + 1$ и переходим к шагу 2.

В качестве приближенного значения точки минимума x^* может быть принята любая точка последнего ТИН.

Некоторые свойства метода золотого сечения.

Утверждение 1. Для любого $r \geq 1$ метод золотого сечения обладает следующим свойством: одна из точек x_1^{r+1}, x_2^{r+1} совпадает с одной из точек x_1^r, x_2^r .

◀ Пусть на r -ой итерации выполняется $f(x_1^r) < f(x_2^r)$, тогда $\text{ТИН}_{r+1} = [a^r, x_2^r]$ и $x_1^r \in \text{ТИН}_{r+1}$. Чтобы доказать справедливость утверждения, нужно показать, что верно соотношение $\frac{x_1^r - a^r}{x_2^r - a^r} = \tau$. Из 2-го шага алгоритма следует, что $b^r - x_1^r = (b^r - a^r) \tau \Rightarrow b^r - x_1^r - a^r + a^r = (b^r - a^r) \tau \Rightarrow (b^r - a^r) - (x_1^r - a^r) = (b^r - a^r) \tau \Rightarrow x_1^r - a^r = (b^r - a^r)(1 - \tau)$. Аналогично имеем $x_2^r - a^r = (b^r - a^r) \tau$. Разделив первый из полученных результатов на второй, получаем $\frac{x_1^r - a^r}{x_2^r - a^r} = \frac{1 - \tau}{\tau}$. Но из (2.2) следует, что $1 - \tau = \tau^2$, в результате чего имеем $\frac{x_1^r - a^r}{x_2^r - a^r} = \frac{1 - \tau}{\tau} = \frac{\tau^2}{\tau} = \tau$ ►

Доказательство для случая $f(x_1^r) > f(x_2^r)$ проводится аналогично. Указанное свойство позволяет на каждой итерации (кроме первой) производить испытания только в одной точке.

Утверждение 2. Точки x_1^r, x_2^r расположены симметрично относительно концов ТИН.

◀ Из 2-го шага алгоритма следует, что точка x_1^r отстоит от точки b^r на величину $(b^r - a^r) \tau$; точка x_2^r отстоит от точки a^r на ту же величину. ►

Утверждение 3. Поскольку после каждой итерации длина ТИН уменьшается в τ раз, можно априорно оценить количество итераций по заданной точности решения. Действительно, после первой итерации $|\text{ТИН}_2| = (b - a)\tau$, после второй итерации $|\text{ТИН}_3| = (b - a)\tau\tau = (b - a)\tau^2$ и т.д. Тогда после r -ой итерации имеем $|\text{ТИН}_{r+1}| = (b - a)\tau^r$. Но, так как условие окончания поиска $|\text{ТИН}_{r+1}| \leq \epsilon_x$, получаем $(b - a)\tau^r \leq \epsilon_x$. Заменяя знак неравенства на равенство и выразив r , получим оценку количества итераций.

2.1.3 Сравнение эффективности методов, сокращающих ТИН

Обозначим через Φ класс непрерывных унимодальных одномерных функций. Пусть множество рассматриваемых методов поиска есть $\{A_1, A_2, A_3, A_4\}$, где

- A_1 - метод равномерного поиска;
- A_2 - метод дихотомии;
- A_3 - метод Фибоначчи;
- A_4 - метод золотого сечения.

В качестве критерия оптимальности указанных методов на классе функций Φ будем использовать максимальную длину ТИН после m **испытаний**:

$$W(A) = \max_{f \in \Phi} |\text{ТИН}_m(f, A)| \quad (2.5)$$

Использование подобного критерия обусловлено тем, что при решении оптимизационной задачи зачастую требуется минимизировать число испытаний (это уже обсуждалось в разделе 1.6). Но, поскольку в предыдущем разделе длина ТИН связывалась с количеством итераций (а, как было показано, почти во всех методах число итераций не совпадает с числом испытаний), необходимо в выведенных формулах сделать переход от итераций к испытаниям.

При рассмотрении метода равномерного поиска было показано, что после каждой итерации длина ТИН уменьшается в $\frac{N}{2}$ раз, а количество испытаний на каждой итерации $m = N + 1$. Следовательно, $W(A_1) = \frac{2}{N} (b - a) = \frac{2}{m-1} (b - a)$.

При рассмотрении метода дихотомии было показано, что после каждой итерации длина ТИН уменьшается в 2 раза, количество испытаний на каждой итерации равно 2, поэтому $m = 2N$. Следовательно, $W(A_2) = \frac{b-a}{2^N} = \frac{b-a}{2^{m/2}}$.

При рассмотрении метода Фибоначчи было показано, что в результате N итераций длина ТИН становится равной $\frac{b-a}{F_N}$, а количество испытаний $m = N + 1$ (на первой итерации два испытания, дальше по одному испытанию на каждой итерации). Следовательно, $W(A_3) = \frac{b-a}{F_{m-1}}$.

При рассмотрении метода золотого сечения было показано, что в результате N итераций длина ТИН становится равной $(b - a)\tau^N$, а количество испытаний $m = N + 1$ (на первой итерации два испытания, дальше по одному испытанию на каждой итерации). Следовательно, $W(A_4) = (b - a)\tau^{m-1}$.

Пользуясь полученными результатами, сравним метод дихотомии и метод Фибоначчи при числе испытаний $m = 14$: $\frac{W(A_2)}{W(A_3)} = \frac{F_{m-1}}{2^{m/2}} = \frac{F_{13}}{2^7} = \frac{233}{128} \approx 1,82$. Видно, что метод Фибоначчи почти в два раза эффективнее метода дихотомии.

2.1.4 Методы полиномиальной аппроксимации

Если при решении оптимизационной задачи была получена информация о том, что целевая функция является достаточно гладкой в некоторой окрестности экстремума, то в этой окрестности ее можно достаточно точно заменить (аппроксимировать) полиномом некоторого порядка, после чего использовать для нахождения экстремума этот полином.

Качество аппроксимации может быть повышено двумя способами:

- Увеличение степени аппроксимирующего полинома;
- Уменьшение сетки аппроксимации.

Второй способ является более предпочтительным, так как построение полинома порядка выше трех - достаточно сложная задача.

Метод квадратичной аппроксимации

В данном методе целевая функция приближенно заменяется параболой, проходящей через три известные точки (x_i, y_i) , где $y_i = f(x_i)$, $i = 1, 2, 3$.

Построим полином следующего вида: $q(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2)$. Необходимо выбрать a_i , $i = 1, 2, 3$ таким образом, чтобы $q(x_i) = y_i$, $i = 1, 2, 3$.

Подставим в полином x_1 : $q(x_1) = a_0$, следовательно, $a_0 = y_1$. Подставим в полином x_2 : $q(x_2) = a_0 + a_1(x_2 - x_1) = y_2$, следовательно, $a_1 = \frac{y_2 - y_1}{x_2 - x_1}$. Подставим в полином x_3 : $q(x_3) = a_0 + a_1(x_3 - x_1) + a_2(x_3 - x_1)(x_3 - x_2) = y_3$, следовательно, $a_2 = \frac{1}{x_3 - x_2} \left(\frac{y_3 - y_1}{x_3 - x_1} - \frac{y_2 - y_1}{x_2 - x_1} \right)$.

Найдем экстремум классическим методом, взяв производную: $q'(x) = a_1 + a_2(x - x_1) + a_2(x - x_2) = 0$, откуда $x^* = \frac{x_1 + x_2}{2} - \frac{a_1}{2a_2}$.

Пример. Найдем с помощью квадратичной аппроксимации экстремум функции $f(x) = \sin(x) + \cos(x)$ на отрезке $[-1; 1]$. В качестве точек аппроксимации возьмем $x_1 = -1$, $x_2 = 0$, $x_3 = 1$, и, соответственно, $y_1 \approx -0,3$, $y_2 = 1$, $y_3 = 1,38$. Для определения оптимальной точки нужно знать коэффициенты a_1 и a_2 . Найдем их: $a_1 = \frac{y_2 - y_1}{x_2 - x_1} \approx 1,3$; $a_2 = \frac{1}{x_3 - x_2} \left(\frac{y_3 - y_1}{x_3 - x_1} - a_1 \right) \approx -0,46$. Тогда $x^* = \frac{x_1 + x_2}{2} - \frac{a_1}{2a_2} \approx 0,91$. Из рисунка 2.4 видно, что построенный аппроксимационный полином достаточно точно приближает целевую функцию.

Подобные расчеты можно повторять для уточнения точки экстремума, на каждой итерации заново выбирая точки x_i , $i = 1, 2, 3$ каким-либо образом. Например, можно найденную на предыдущем шаге точку оптимума обозначать как x_2 , а в качестве точек x_1 и x_3 брать те использовавшиеся на предыдущем шаге точки, между которыми оказалась точка оптимума. Например, для примера выше $x_1 = -1$, $x_2 = 0$, $x_3 = 1$ и $x^* \approx 0,91$. Тогда на следующем шаге

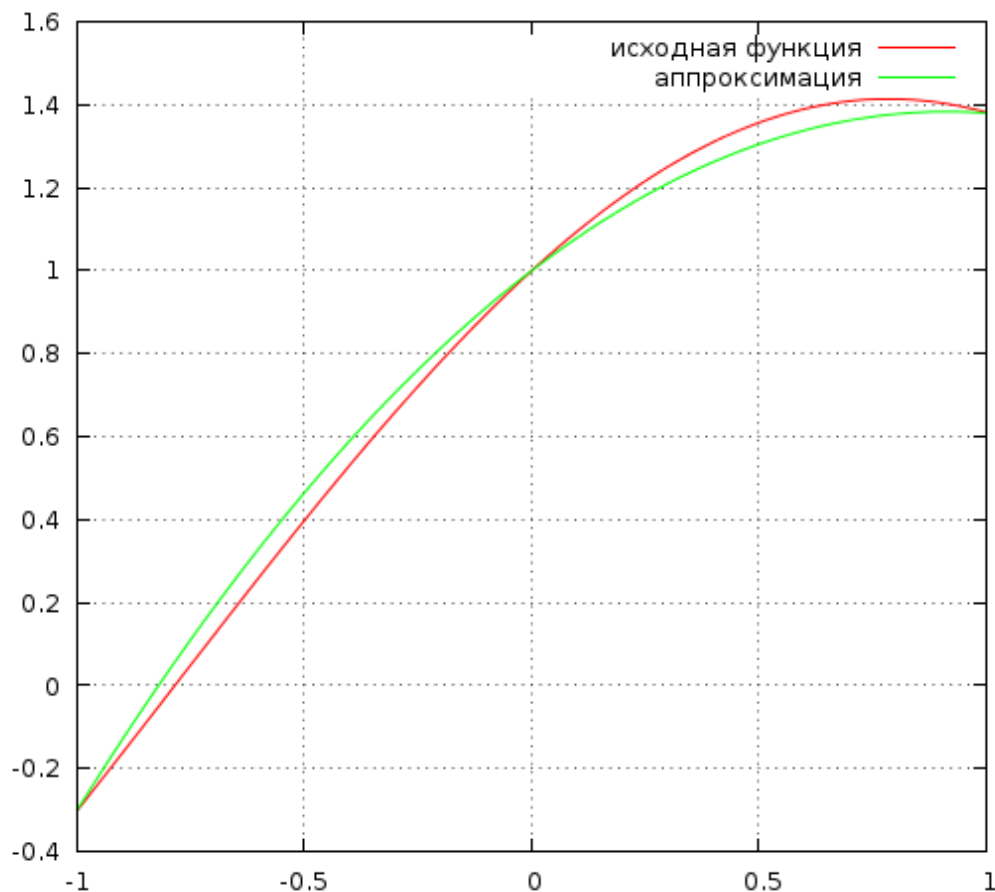


Рис. 2.4. Целевая функция и аппроксимирующий полином

можно положить $x_1 = 0$, $x_2 \approx 0,91$, $x_3 = 1$. Однако это не всегда возможно: скажем, вполне возможна ситуация, когда $x^* \notin [x_1; x_3]$; решить эту проблему помогает метод Пауэлла.

Метод Пауэлла

1. Выбираем начальную точку x_1 , величину шага Δ_x и ϵ_1 , ϵ_2 - малые положительные числа.
2. Выполняем $x_2 = x_1 + \Delta_x$.
3. Вычисляем $f(x_1)$ и $f(x_2)$. Если $f(x_1) > f(x_2)$, то $x_3 = x_1 + 2\Delta_x$; в противном случае $x_3 = x_1 - \Delta_x$.
4. Вычисляем $f(x_3)$ и находим $f_{\min} = \min \{f(x_1), f(x_2), f(x_3)\}$; $x_{\min} = \{x_i | f(x_i) = f_{\min}\}$.
5. По x_1, x_2, x_3 вычисляем x^* и $f(x^*)$, пользуясь формулами квадратичной аппроксимации. Если $a_2 = 0$, то результатом аппроксимации является не парабола, а прямая; в этом случае следует положить $x_1 = x_{\min}$ и перейти к шагу 2.
6. Проверяем условия окончания поиска: $\left| \frac{f_{\min} - f(x^*)}{f(x^*)} \right| \leq \epsilon_1$, $\left| \frac{x_{\min} - x^*}{x^*} \right| \leq \epsilon_2$.
 - Если оба условия выполняются, останавливаем поиск; точка оптимума - x^* ;

- Если хотя бы одно условие не выполняется и $x^* \in [x_1; x_3]$ (или $x^* \in [x_3; x_2]$ - в зависимости от выбранного на шаге 3 направления поиска), то выбираем «наилучшую точку» (либо x_{min} , либо x^*) и две использовавшиеся точки по обе стороны от нее. Обозначаем эти точки в естественном порядке x_1, x_2, x_3 и переходим к шагу 5;
- Если хотя бы одно условие не выполняется и $x^* \notin [x_1; x_3]$ (или $x^* \notin [x_3; x_2]$ - в зависимости от выбранного на шаге 3 направления поиска), то обозначаем $x_1 = x_{min}$ и переходим к шагу 2.

2.1.5 Методы, использующие производные

Если целевая функция достаточно просто дифференцируется и вычисление значения производной не представляет значительных сложностей, то целесообразно воспользоваться этим обстоятельством для поиска экстремума. Общая идея излагаемых ниже методов состоит в численном решении уравнения классического метода оптимизации - $f'(x) = 0$.

Метод средней точки

Данный метод очень схож с методом дихотомии: определяются две точки L и R , в которых производная целевой функции имеет разные знаки (например, $f'(L) < 0$, $f'(R) > 0$). Тогда точка экстремума, в которой $f'(x) = 0$, будет находиться на отрезке $[L, R]$. Полагаем $Z = \frac{L+R}{2}$ и вычисляем $f'(Z)$, после чего исключаем из рассмотрения один из отрезков ($[L, Z]$ или $[Z, R]$), на краях которого знаки производной совпадают. Процесс можно повторять до достижения требуемой точности: $|R - L| \leq \epsilon$ или $|f'(Z)| \leq \epsilon$.

Метод Ньютона (касательных)

Если целевая функция имеет непрерывные и сохраняющие определенные знаки производные вплоть до третьей включительно на некотором отрезке $[a, b]$, причем первая производная имеет разные знаки на концах этого отрезка (то есть $f'(a)f'(b) < 0$), то, исходя из начального приближения $x_0 \in [a, b]$, удовлетворяющего условию $f'(x_0)f'''(x_0) > 0$, можно вычислить с любой точностью единственный корень уравнения $f'(x) = 0$.

Проведем через точку (x_0, y_0) касательную. Уравнение такой касательной записывается как

$$y = f(x_0) + f'(x_0)(x - x_0) \quad (2.6)$$

Поскольку мы ищем решение уравнения $f'(x) = 0$, заменим в (2.6) f на f' и f' на f'' , после чего найдем точку пересечения касательной с осью абсцисс ($y = 0$): $f'(x_0) + f''(x_0)(x - x_0) = 0$. Из полученного соотношения выразим x : $x = x_0 - \frac{f'(x_0)}{f''(x_0)}$. Тогда можно построить итерационный процесс:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} \quad (2.7)$$

Геометрическая интерпретация метода приведена на рисунке 2.5. Итерационный процесс (2.7) заканчивается, когда $|f'(x_{k+1})| \leq \epsilon$ или $|x_{k+1} - x_k| \leq \epsilon$.

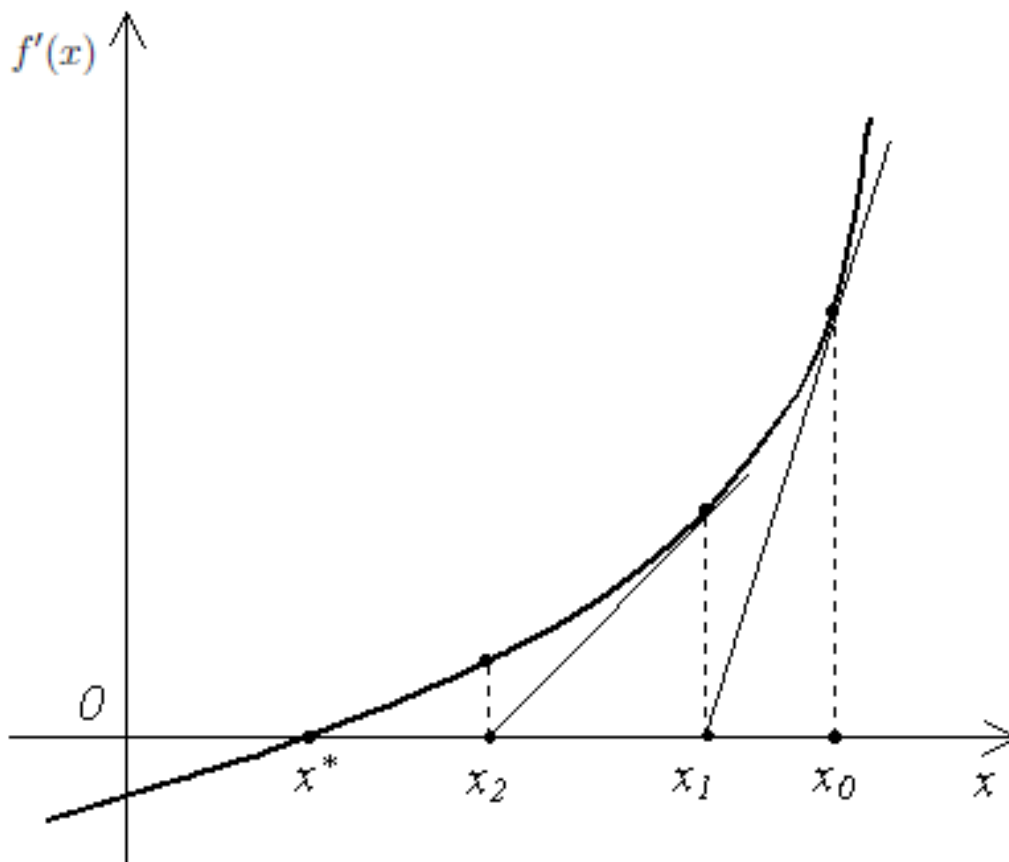


Рис. 2.5. Геометрическая интерпретация метода Ньютона

Метод секущих (хорд)

Метод секущих получается из метода касательных заменой $f''(x_k)$ в (2.7) разностным приближением: $f''(x_k) \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$. В результате получим формулу итерационного процесса:

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})} f'(x_k) \quad (2.8)$$

Методом хорд называют метод секущих в том случае, когда один из концов отрезка $[a, b]$ закреплён, т. е. вычисление приближения корня уравнения $f'(x) = 0$ производят по формулам:

$$x_{k+1} = x_k - \frac{x_k - a}{f'(x_k) - f'(a)} f'(x_k), \quad x_0 = b \quad (2.9)$$

либо

$$x_{k+1} = x_k - \frac{x_k - b}{f'(x_k) - f'(b)} f'(x_k), \quad x_0 = a \quad (2.10)$$

В качестве x_0 в обоих методах выбирают тот конец отрезка, на котором выполняется $f'(x_0)f'''(x_0) > 0$. Для метода секущих дополнительно нужно определить x_1 , так как этот метод является двухшаговым; это делается с помощью одной итерации метода хорд (см. (2.9), (2.10)), а все последующие значения вычисляются с помощью (2.8).

Геометрическая интерпретация метода хорд приведена на рисунке 2.6. Итерационный процесс (2.8) заканчивается, когда $|f'(x_{k+1})| \leq \epsilon$ или $|x_{k+1} - x_k| \leq \epsilon$.

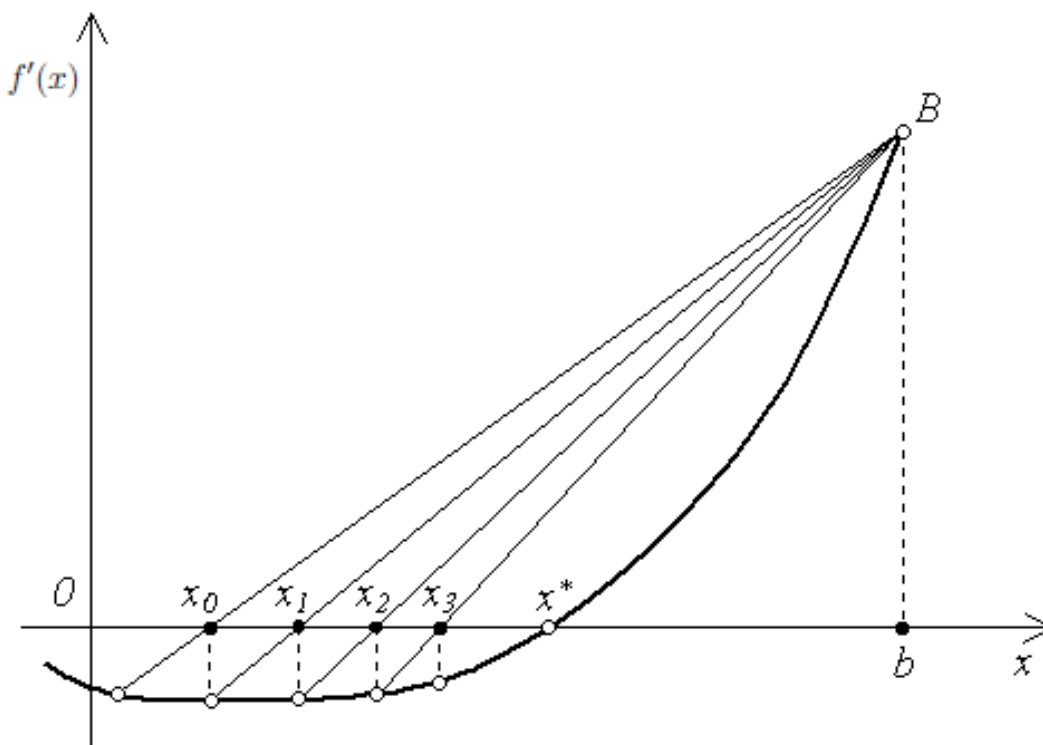


Рис. 2.6. Геометрическая интерпретация метода хорд

2.2 Безусловная оптимизация функций многих переменных

2.2.1 Основные понятия и определения

Постановка задачи

Пусть задана функция n действительных переменных $f(\bar{x}) = f(x_1, x_2, \dots, x_n)$, определенная на множестве $X \in \mathbb{R}^n$. Говорят, что функция $f(\bar{x})$ имеет **локальный минимум** в точке $\bar{x}^* \in X$, если существует окрестность точки \bar{x}^* , такая, что $f(\bar{x}^*) \leq f(\bar{x})$ для любой точки этой окрестности. Аналогично, функция имеет **глобальный минимум** в точке $\bar{x}^* \in X$, если условие $f(\bar{x}^*) \leq f(\bar{x})$ справедливо для любой $\bar{x} \in X$.

Поверхность и линия уровня

Поверхностью уровня называется геометрическое место точек, где $f(\bar{x}) = \text{const}$. Для функции двух переменных поверхность уровня называется **линией уровня**.

Градиент функции многих переменных и его свойства

Градиентом функции многих переменных называется вектор, составленный из первых частных производных функции по всем переменным:

$$\nabla f(\bar{x}) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)^T \quad (2.11)$$

Свойства градиента:

1. Градиент функции перпендикулярен поверхности уровня функции $f(\bar{x})$; в случае функции двух переменных он перпендикулярен касательной к линии уровня.
2. Направление градиента есть направление наиболее быстрого роста функции.

Матрица Гессе

Квадратная матрица размера $n \times n$, состоящая из вторых частных производных функции $f(\bar{x})$, называется **матрицей Гессе**:

$$H(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix} \quad (2.12)$$

Критерий Сильвестра

Матрица является **положительно определенной**, если все ее диагональные миноры положительно определены; и является **отрицательно определенной**, если все ее диагональные миноры чередуют знак, начиная со знака «минус».

Понятие стационарности

Точка \bar{x}^* называется **стационарной**, если $\nabla f(\bar{x}^*) = \bar{0}$ (следует понимать, что $\bar{0}$ - нулевой вектор).

Условия локальной оптимальности

Необходимое условие локальной оптимальности.

Если \bar{x}^* - точка, в которой достигается локальный экстремум, то выполняется условие стационарности $\nabla f(\bar{x}^*) = \bar{0}$ (обратное в общем случае неверно!).

Достаточное условие локальной оптимальности.

Пусть в точке \bar{x}^* $f(\bar{x})$ дважды дифференцируема и выполняется необходимое условие локальной оптимальности. Тогда, если матрица Гессе для указанной точки положительно определена, то \bar{x}^* - точка локального минимума; если матрица Гессе для указанной точки

отрицательно определена, то \bar{x}^* - точка локального максимума; если матрица Гессе для указанной точки не определена ни положительно, ни отрицательно, то \bar{x}^* - седловая точка.

Пример. Попробуем аналитически определить точку экстремума для функции $f(x, y) = 2x^2 + 2y^2 + 2xy + 20x + 10y + 10$. Для этого посчитаем частные производные и составим вектор градиента: $\frac{\partial f}{\partial x} = 4x + 2y + 20$, $\frac{\partial f}{\partial y} = 4y + 2x + 10$, $\nabla f = (4x + 2y + 20, 4y + 2x + 10)^T$. Найдем стационарную точку из условия $\nabla f = (0, 0)^T$, приравняв частные производные к нулю. в результате решения системы уравнений получим $x = -5, y = 0$. Теперь построим матрицу Гессе: $H = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} = \begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix}$. Все диагональные миноры этой матрицы положительно определены, следовательно, найденная точка является точкой минимума. График с изображением линий уровня в окрестности найденной точки это подтверждает (рис. 2.7).

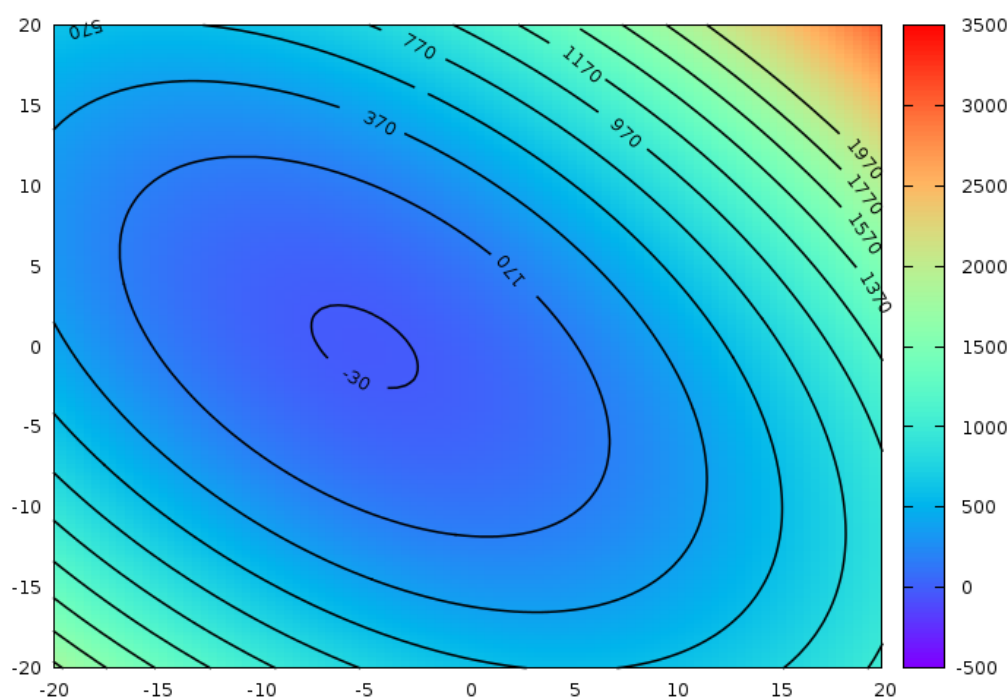


Рис. 2.7. Линии уровня в окрестности точки оптимума

Однако, как уже было сказано, вычислять производные при решении задач оптимизации удастся не всегда. Поэтому сначала рассмотрим ряд методов, позволяющих находить экстремумы функций многих переменных без вычисления частных производных.

2.2.2 Простейшие поисковые методы

Метод Гаусса-Зейделя (покоординатного спуска)

В методе Гаусса-Зейделя на каждом шаге ищут экстремум целевой функции, фиксируя все переменные, кроме какой-либо одной; значения фиксированных переменных полагаются постоянными. Поиск экстремума при одной свободной переменной выполняют любым методом одномерной оптимизации. Таким образом, в методе Гаусса-Зейделя задача многомерной оптимизации сводится к задаче одномерной оптимизации.

Алгоритм метода:

1. Выбираем начальное приближение $\bar{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$; устанавливаем $i = 0$.
2. Фиксируем все координаты, кроме первой, и проводим операцию одномерного поиска $f(x_1) = f(x_1, x_2^{(i)}, x_3^{(i)}, \dots, x_n^{(i)})$; в результате получаем $\bar{x}_1^{(i+1)} = (x_1^*, x_2^{(i)}, x_3^{(i)}, \dots, x_n^{(i)})$.
3. Фиксируем в точке $\bar{x}_1^{(i+1)}$ все координаты, кроме второй, и повторяем одномерный поиск $f(x_2) = f(x_1^*, x_2, x_3^{(i)}, \dots, x_n^{(i)})$; в результате получаем $\bar{x}_2^{(i+1)} = (x_1^*, x_2^*, x_3^{(i)}, \dots, x_n^{(i)})$ и т.д. Продолжаем указанный процесс для всех n координат, пока не получим $\bar{x}_n^{(i+1)} = (x_1^*, x_2^*, x_3^*, \dots, x_n^*)$.
4. Обозначаем $\bar{x}_n^{(i+1)} = \bar{x}^{(i+1)}$ и проверяем условие окончания поиска $\|\bar{x}^{(i+1)} - \bar{x}^{(i)}\| \leq \epsilon$. Если условие окончания поиска не выполняется, устанавливаем $i = i + 1$ и переходим к шагу 2; в противном случае принимаем за точку оптимума $\bar{x}^{(i+1)}$.

Метод обладает тем недостатком, что, в случае так называемой «овражной» функции, оптимум никогда не будет достигнут при попадании точки в «овраг».

Метод Хука-Дживса (конфигураций)

Этот метод можно рассматривать как модификацию метода Гаусса-Зейделя. Идея состоит в том, что из точки начального приближения выполняется одна итерация по методу Гаусса-Зейделя («исследующий поиск»), а дальнейший поиск («поиск по образцу») проводят вдоль прямой, соединяющей точку начального приближения и точку, полученную во время исследующего поиска. Когда на этой прямой будет найдена точка, значение целевой функции в которой в известном смысле оптимально, из нее снова выполняют одну итерацию по методу Гаусса-Зейделя и т.д.

Алгоритм метода:

1. Выбираем начальное приближение $\bar{x}^0 = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$; устанавливаем $i = 0$.
2. Исследующий поиск: выполняем одну итерацию по методу Гаусса-Зейделя, получаем точку $\bar{x}^{(i+1)}$. Проверяем условие окончания поиска $\|\bar{x}^{(i+1)} - \bar{x}^{(i)}\| \leq \epsilon$. Если оно не выполняется, переходим к шагу 3; в противном случае принимаем за точку оптимума $\bar{x}^{(i+1)}$.
3. Поиск по образцу: соединяем точки $\bar{x}^{(i+1)}$ и $\bar{x}^{(i)}$ прямой, после чего двигаемся вдоль полученного направления с некоторым шагом k : $\bar{x}^{(j+1)} = \bar{x}^{(j)} + k(\bar{x}^{(i+1)} - \bar{x}^{(i)})$ (на первой итерации поиска по образцу $j = i$). После каждого определения новой точки проверяем условие окончания поиска $\|\bar{x}^{(j+1)} - \bar{x}^{(j)}\| \leq \epsilon$. Если оно не выполняется, то устанавливаем $j = j + 1$ и повторяем поиск по образцу; в противном случае принимаем за точку оптимума $\bar{x}^{(j+1)}$.

4. Если на шаге исследующего поиска найдена «оптимальная» точка на выбранном направлении, и условие остановки не выполнено, то полагаем $i = j$ и возвращаемся к исследующему поиску (шаг 2).

За счет возможности варьирования параметра k данный метод лучше метода Гаусса-Зейделя в смысле преодоления «оврагов», однако полностью исключить «застывание в овраге» все равно невозможно.

2.2.3 Симплексные методы

Симплексом в пространстве n переменных называется выпуклый многогранник, имеющий $n + 1$ вершину: в пространстве 2-х переменных - это треугольник, в пространстве 3-х переменных - тетраэдр и т.д.

Обычный симплекс-метод

В обычном симплекс-методе используется правильный симплекс, все ребра которого равны. Для наглядности рассмотрим идею симплекс-метода для случая 2-х переменных.

Выбираем начальный симплекс с координатами вершин $\bar{x}^{(1)}$, $\bar{x}^{(2)}$, $\bar{x}^{(3)}$. Размещение этого симплекса в пространстве может быть осуществлено двумя путями.

1. Одна из вершин симплекса помещается в начало координат, а остальные располагаются так, чтобы ребра, выходящее из первой вершины, образовывали одинаковые углы с соответствующими координатными осями (рис. 2.8). Координаты вершин для случая единичного симплекса приведены в таблице 2.1.

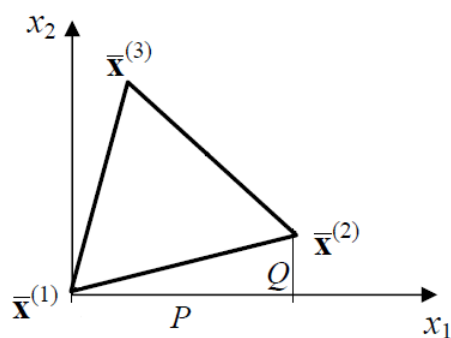


Таблица 2.1.

Вершина	x_1	x_2
$\bar{x}^{(1)}$	0	0
$\bar{x}^{(2)}$	$\frac{\sqrt{3}+1}{2\sqrt{2}}$	$\frac{\sqrt{3}-1}{2\sqrt{2}}$
$\bar{x}^{(3)}$	$\frac{\sqrt{3}-1}{2\sqrt{2}}$	$\frac{\sqrt{3}+1}{2\sqrt{2}}$

Рис. 2.8. Симплекс с вершиной в начале координат

Введя параметры $P = \frac{1}{n\sqrt{2}}(\sqrt{n+1} + n - 1)$ и $Q = \frac{1}{n\sqrt{2}}(\sqrt{n+1} - 1)$, можно расширить таблицу 2.1 для случая единичного симплекса в пространстве n переменных (таблица 2.2).

2. Центр симплекса помещается в начало координат, а $(n+1)$ -я вершина на ось x_n ; остальные располагаются симметрично относительно координатных осей (рис. 2.9). Координаты вершин для случая единичного симплекса приведены в таблице 2.3.

Таблица 2.2.

№ вершины	x_1	x_2	x_3	\dots	x_n
1	0	0	0	\dots	0
2	P	Q	Q	\dots	Q
3	Q	P	Q	\dots	Q
\dots	\dots	\dots	\dots	\dots	\dots
$n + 1$	Q	Q	Q	\dots	P

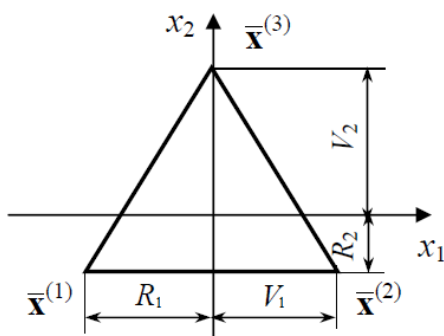


Таблица 2.3.

Вершина	x_1	x_2
$\bar{x}^{(1)}$	$-\frac{1}{2}$	$-\frac{1}{2\sqrt{3}}$
$\bar{x}^{(2)}$	$\frac{1}{2}$	$-\frac{1}{2\sqrt{3}}$
$\bar{x}^{(3)}$	0	$\frac{1}{\sqrt{3}}$

Рис. 2.9. Симплекс с вершиной в начале координат

Введя параметры $R_i = \frac{1}{\sqrt{2i(i+1)}}$ и $V_i = \sqrt{\frac{i}{2(i+1)}}$, можно расширить таблицу 2.3 для случая единичного симплекса в пространстве n переменных (таблица 2.4).

Таблица 2.4.

№ вершины	x_1	x_2	x_3	\dots	x_n
1	$-R_1$	$-R_2$	$-R_3$	\dots	$-R_n$
2	V_1	$-R_2$	$-R_3$	\dots	$-R_n$
3	0	V_2	$-R_3$	\dots	$-R_n$
4	0	0	V_3	\dots	$-R_n$
\dots	\dots	\dots	\dots	\dots	\dots
$n + 1$	0	0	0	\dots	$-R_n$

Как уже упоминалось, все формулы выше даны для случая единичного симплекса. Для произвольной длины ребра каждую формулу нужно дополнительно умножить на желаемую длину ребра.

Если поиск осуществляется не из начала координат, а из начальной точки $\bar{x}^{(0)}$, то к координатам вершин симплекса необходимо прибавить координаты начальной точки.

Алгоритм метода:

1. Выбираем начальное приближение $\bar{x}^{(0)}$ и строим правильный симплекс одним из предложенных способов.
2. В вершинах симплекса вычисляем значения целевой функции, выбираем из них «худшее» и через центр тяжести противоположащей грани строим новую вершину симплекса,

симметричную «худшей» вершине (рис. 2.10). Например, если «худшая» вершина - $\bar{x}^{(1)}$, то $\bar{x}_{ц.т.} = \frac{\bar{x}^{(2)} + \bar{x}^{(3)}}{2}$ и новая вершина $\bar{x}^{(4)} = \bar{x}_{ц.т.} + (\bar{x}_{ц.т.} - \bar{x}^{(1)}) = \bar{x}^{(2)} + \bar{x}^{(3)} - \bar{x}^{(1)}$.

3. В результате отображения «худшей» точки получаем новый симплекс, причем в двух его вершинах значение целевой функции уже известно. Продолжаем описанный процесс до тех пор, пока он не заикнется (отображенная вершина становится «худшей»). В этом случае смотрим на длину ребра симплекса - если она меньше заданной точности, то останавливаем процесс поиска; в противном случае уменьшаем размер симплекса и продолжаем поиск.

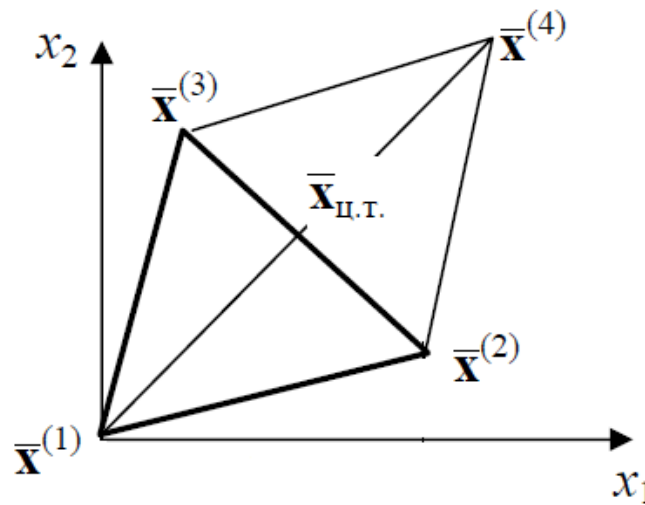


Рис. 2.10. Отображение точки с «худшим» значением функции

В общем, n -мерном случае, если обозначить отображаемую вершину $\bar{x}^{(j)}$, то

$$\begin{cases} \bar{x}_{ц.т.} = \frac{1}{n} \sum_{i \neq j} \bar{x}^{(i)} \\ \bar{x}^{(n+2)} = \bar{x}_{ц.т.} + (\bar{x}_{ц.т.} - \bar{x}^{(j)}) \end{cases} \quad (2.13)$$

Иногда в (2.13) используют не зеркальное отображение, а отображение с неким коэффициентом $\alpha > 0$: $\bar{x}^{(n+2)} = \bar{x}_{ц.т.} + \alpha(\bar{x}_{ц.т.} - \bar{x}^{(j)})$, что позволяет по ходу поиска динамически менять размеры симплекса. Такая идея изменения размеров положена в основу метода Нелдера-Мида.

Метод Нелдера-Мида (деформируемых многогранников)

Данный метод является существенно более эффективным за счет того, что симплекс меняет свою форму от итерации к итерации.

Алгоритм метода:

1. Выбираем начальный (обычно правильный) симплекс с вершинами $\bar{x}^{(1)}, \bar{x}^{(2)}, \dots, \bar{x}^{(n+1)}$ и, как в предыдущих методах, рассчитываем в этих вершинах значения целевой функции.

2. Среди вычисленных значений ищем «наилучшее» и «наихудшее» с точки зрения решаемой оптимизационной задачи. Предполагая, что мы решаем задачу на минимум, обозначим их y_{min} и y_{max} соответственно.
3. Отражаем «наихудшую» вершину относительно центра тяжести противоположащей грани с коэффициентом α : $\bar{x}^{(n+2)} = \bar{x}_{ц.т.} + \alpha(\bar{x}_{ц.т.} - \bar{x}^{(j)})$ (обычно $\alpha = 1$), после чего вычисляем значение целевой функции в полученной точке $f(\bar{x}^{(n+2)}) = y^{(n+2)}$.
4. Анализируем результат отражения, сравнивая значения целевой функции $y^{(n+2)}$, y_{min} , y_{max} :

- $y^{(n+2)} < y_{min}$

В этом случае проводят **растяжение** симплекса, увеличивая в $\gamma > 1$ раз расстояние от отображенной вершины до центра тяжести противоположащей грани: $\bar{x}^{(n+3)} = \gamma\bar{x}^{(n+2)} + (1 - \gamma)\bar{x}_{ц.т.}$ (обычно $\gamma = 2$). Если после операции растяжения значение функции в новой точке улучшается (то есть $y^{(n+3)} \leq y^{(n+2)}$), то эта точка принимается за новую вершину очередного симплекса, в противном случае зановую вершину принимается $y^{(n+2)}$.

- $y_{min} < y^{(n+2)} < y_{max}$

В этом случае проводят **сжатие** симплекса с коэффициентом $\beta < 1$: $\bar{x}^{(n+3)} = \beta\bar{x}^{(n+2)} + (1 - \beta)\bar{x}_{ц.т.}$ (обычно $\beta = 0,5$).

- $y_{max} < y^{(n+2)}$

В этом случае проводят **редукцию** (уменьшение) симплекса (обычно в два раза), то есть координаты всех вершин симплекса сдвигаются на половину расстояния до наилучшей точки: $\bar{x}^{(i)} = \bar{x}_{min} + 0,5(\bar{x}^{(i)} - \bar{x}_{min})$, $i = \overline{1, n+1}$.

В качестве критерия останова авторы метода рекомендуют среднеквадратичную величину разности значений функции в вершинах симплекса и среднего ее значения:

$$\sqrt{\frac{1}{n+1} \sum_{i=1}^{n+1} [f(\bar{x}^{(i)}) - f(\bar{x}_{ц.т.})]^2} \leq \epsilon$$

2.2.4 Градиентные методы

Суть всех градиентных методов заключается в использовании вектора градиента для определения направления движения к экстремуму. Свойства градиента, обуславливающие его эффективное применение при поиске, уже приводились ранее:

- Поиск в направлении вектора градиента (антиградиента) дает быструю сходимость, так как градиент (антиградиент) всегда направлен в сторону наиболее быстрого роста (убывания) функции.
- Условие стационарности (в точке экстремума градиент обращается в ноль) используется для проверки условия окончания поиска, то есть $\|\nabla f(\bar{x})\| \leq \epsilon$.

Общий алгоритм всех градиентных методов заключается в построении из некоторой начальной точки $\bar{x}^{(0)}$ последовательности приближений

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \lambda^{(k)} \bar{s}^{(k)}, \text{ где} \quad (2.14)$$

- Знак « $-$ » в правой части говорит о том, что ищется минимум целевой функции (направление антиградиента); при поиске максимума следует использовать знак « $+$ ».
- $\lambda^{(k)}$ - величина шага.
- $\bar{s}^{(k)}$ - единичный вектор в направлении градиента целевой функции в точке $\bar{x}^{(k)}$: $\bar{s}^{(k)} = \frac{\nabla f(\bar{x}^{(k)})}{\|\nabla f(\bar{x}^{(k)})\|} = \frac{\nabla f(\bar{x}^{(k)})}{\sqrt{\left(\frac{\partial f}{\partial x_1}\right)^2 + \left(\frac{\partial f}{\partial x_2}\right)^2 + \dots + \left(\frac{\partial f}{\partial x_n}\right)^2}}$

Следует заметить, что применение градиентных методов не всегда возможно, так как посчитать частные производные целевой функции не всегда просто (а иногда и невозможно). Поэтому в данном разделе кратко рассмотрим два метода, базирующихся на итерационном процессе (2.14): простой градиентный метод и метод наискорейшего спуска; а также метод Бокса-Уилсона, позволяющий выполнить оценку значения градиента без его непосредственного вычисления.

Простой градиентный метод

Суть простого градиентного метода заключается в использовании итерационного процесса (2.14) с фиксированным значением шага: $\lambda^{(k)} = \lambda = \text{const}$, $\bar{x}^{(k+1)} = \bar{x}^{(k)} - \lambda \bar{s}^{(k)}$.

Метод наискорейшего спуска

В методе наискорейшего спуска шаг варьируется на каждой итерации и определяется решением одномерной задачи оптимизации: $\lambda^{(k)} = \underset{\lambda}{\operatorname{argmin}} f(\bar{x}^{(k)} - \lambda \bar{s}^{(k)})$.

Метод Бокса-Уилсона (крутого восхождения)

Этот метод представляет собой пошаговую процедуру движения по поверхности целевой функции, в которой для оценки составляющих градиента $\nabla f(\bar{x}^{(k)}) = (b_0^{(k)}, b_1^{(k)}, \dots, b_n^{(k)})^T$ используется линейное уравнение регрессии $r(\bar{x}) = b_0^{(k)} + b_1^{(k)} x_1 + \dots + b_n^{(k)} x_n$, полученное в результате планирования эксперимента в окрестности точки $\bar{x}^{(k)}$.

Затем совершается движение по поверхности целевой функции в направлении градиента (антиградиента) с величиной шага, пропорциональной произведению коэффициента $b_j^{(k)}$ на интервал варьирования Δx_j . Движение осуществляется до тех пор, пока значение целевой функции не начнет уменьшаться (увеличиваться). В полученной точке вновь производится планирование эксперимента и оценка нового направления движения.

Рассмотрим для наглядности случай функции двух переменных. Пусть мы имеем функцию $f(x_1, x_2)$. Выбираем начальную точку поиска $\bar{x}^{(0)} = (x_1^{(0)}, x_2^{(0)})^T$. В окрестности этой

точки проводим полный факторный эксперимент (ПФЭ) размерности $N = k^n$, где k - количество уровней варьирования, n - количество факторов; в нашем случае $N = 2^2 = 4$. Графическая интерпретация этого процесса приведена на рисунке 2.11. По сути дела, в данном случае ПФЭ помогает выбрать экспериментальные точки, в которых будут рассчитаны значения функции. Для удобства эти расчеты можно записать в виде матрицы планирования эксперимента (таблица 2.5).

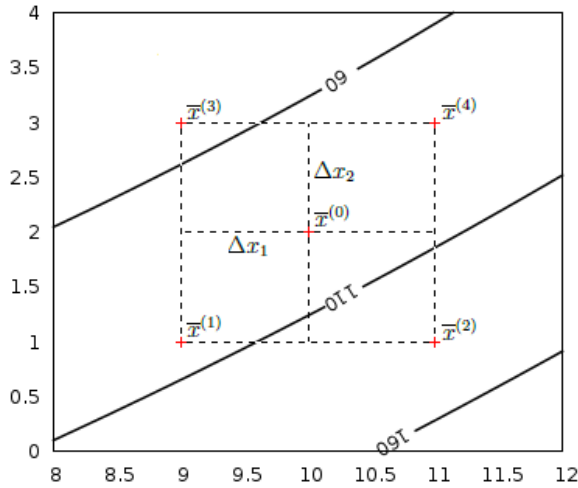


Рис. 2.11. ПФЭ в окрестности начальной точки

Таблица 2.5.

Вершина	X_1	X_2	$y = f(\bar{x})$
$\bar{x}^{(1)}$	-1	-1	y_1
$\bar{x}^{(2)}$	1	-1	y_2
$\bar{x}^{(3)}$	-1	1	y_3
$\bar{x}^{(4)}$	1	1	y_4

Матрица планирования 2.5 представлена в кодированных переменных. Переход от кодированных к натуральным переменным осуществляется по формуле $x_j = x_j^{(0)} + X_j \Delta x_j$, $j = \overline{1, n}$. Здесь $x_j^{(0)}$ - j -ая координата начальной точки, X_j - кодированное значение переменной (-1 или +1), Δx_j - интервал варьирования. Тогда матрица планирования в натуральных переменных будет иметь вид, представленный в таблице 2.6:

Таблица 2.6.

Вершина	x_1	x_2	y
$\bar{x}^{(1)}$	$x_1^{(0)} - \Delta x_1$	$x_2^{(0)} - \Delta x_2$	y_1
$\bar{x}^{(2)}$	$x_1^{(0)} + \Delta x_1$	$x_2^{(0)} - \Delta x_2$	y_2
$\bar{x}^{(3)}$	$x_1^{(0)} - \Delta x_1$	$x_2^{(0)} + \Delta x_2$	y_3
$\bar{x}^{(4)}$	$x_1^{(0)} + \Delta x_1$	$x_2^{(0)} + \Delta x_2$	y_4

Положив для простоты $\Delta x_1 = \Delta x_2 = 1$, получим матрицу 2.7:

Таблица 2.7.

Вершина	x_1	x_2	y
$\bar{x}^{(1)}$	$x_1^{(0)} - 1$	$x_2^{(0)} - 1$	y_1
$\bar{x}^{(2)}$	$x_1^{(0)} + 1$	$x_2^{(0)} - 1$	y_2
$\bar{x}^{(3)}$	$x_1^{(0)} - 1$	$x_2^{(0)} + 1$	y_3
$\bar{x}^{(4)}$	$x_1^{(0)} + 1$	$x_2^{(0)} + 1$	y_4

Метод ПФЭ позволяет получить коэффициенты b_i для уравнения регрессии $r(\bar{x}) = b_0 + b_1 x_1 + \dots + b_n x_n$ по формуле $b_i = \frac{1}{N} \sum_{j=1}^N X_{ij} y_j$, $i = \overline{1, n}$, $j = \overline{1, N}$. Тогда в рассматриваемом двумерном случае имеем $b_1 = \frac{1}{4} (-y_1 + y_2 - y_3 + y_4)$, $b_2 = \frac{1}{4} (-y_1 - y_2 + y_3 + y_4)$.

Коэффициенты b_i дают оценку градиента, которую можно использовать для выбора направления движения. При таком подходе при поиске минимума координаты новых точек вычисляются по формуле $x_j^{(k)} = x_j^{(0)} - kab_j \Delta x_j$ (при поиске максимума знак «-» в формуле меняется на «+»). Здесь $x_j^{(k)}$ - значение j -ой координаты на k -ом шаге, a - коэффициент пропорциональности (выбирается с учетом вида функции).

Движение осуществляется до тех пор, пока значение функции не будет увеличиваться (уменьшаться). В точке, где был достигнут минимум (максимум) вновь реализуется ПФЭ, рассчитываются коэффициенты регрессии и осуществляется движение в новом направлении и т.д. Поиск заканчивается, когда $\|\nabla f(\bar{x})\| = \sqrt{b_1^2 + \dots + b_n^2} \leq \epsilon$.

2.2.5 Пример использования

В заключение данного раздела рассмотрим использование методов многомерной безусловной оптимизации на примере самого интересного, пожалуй, метода - Бокса-Уилсона. Определим минимум функции $f(x_1, x_2) = (x_1 - x_2)^2 + (x_2 - 7)^2$ с точностью $\epsilon = 0,25$, используя в качестве начальной точки $\bar{x}^{(0)} = (10; 2)^T$.

Для начала выберем интервалы варьирования, для простоты расчетов положив $\Delta x_1 = \Delta x_2 = 1$. Теперь можно реализовать ПФЭ размерности $N = 2^2 = 4$, используя в качестве основной точки $x^{(0)}$. Матрица планирования 2.8 будет иметь вид (вместе с натуральными значениями переменных в скобках записаны кодированные):

Таблица 2.8.

Вершина	x_1	x_2	y
$\bar{x}^{(1)}$	9 (-1)	1 (-1)	100
$\bar{x}^{(2)}$	11 (+1)	1 (-1)	136
$\bar{x}^{(3)}$	9 (-1)	3 (+1)	52
$\bar{x}^{(4)}$	11 (+1)	3 (+1)	80

Рассчитаем коэффициенты b_i , дающие оценку градиента:

$$b_1 = \frac{1}{4} (-100 + 136 - 52 + 80) = 16$$

$$b_2 = \frac{1}{4} (-100 - 136 + 52 + 80) = -26$$

Проверим выполнение условия останова: $\|\nabla f(\bar{x})\| = \sqrt{b_1^2 + b_2^2} \approx 30,53 > \epsilon$, то есть условие не выполняется.

Рассчитаем величину шагов по каждой координате, положив коэффициент пропорциональности $a = 0,1$:

$$h_1 = ab_1 \Delta x_1 = 0,1 \cdot 16 \cdot 1 = 1,6$$

$$h_2 = ab_2 \Delta x_2 = 0,1 \cdot (-26) \cdot 1 = -2,6$$

Теперь можно начать двигаться в направлении антиградиента, используя формулы:

$$x_1^{(k)} = x_1^{(0)} - kh_1 = 10 - 1,6k$$

$$x_2^{(k)} = x_2^{(0)} - kh_2 = 2 + 2,6k$$

Данные шаги показаны в таблице 2.9:

Таблица 2.9.

Шаг	x_1	x_2	y
0	10	2	89
1	8,4	4,6	20,2
2	6,8	7,2	0,2
3	5,2	9,8	29

Из таблицы 2.9 видно, что функция перестала убывать после 2-го шага. Таким образом, новой начальной точкой становится $\bar{x}^{(2)} = (6,8; 7,2)^T$.

Реализуем ПФЭ размерности $N = 2^2 = 4$. Матрица планирования 2.10 будет иметь вид:

Таблица 2.10.

Вершина	x_1	x_2	y
$\bar{x}^{(1)}$	5,8 (-1)	6,2 (-1)	0,8
$\bar{x}^{(2)}$	7,8 (+1)	6,2 (-1)	3,2
$\bar{x}^{(3)}$	5,8 (-1)	8,2 (+1)	7,2
$\bar{x}^{(4)}$	7,8 (+1)	8,2 (+1)	1,6

Рассчитаем коэффициенты b_i , дающие оценку градиента:

$$b_1 = \frac{1}{4}(-0,8 + 3,2 - 7,2 + 1,6) = -0,8$$

$$b_2 = \frac{1}{4}(-0,8 - 3,2 + 7,2 + 1,6) = 1,2$$

Проверим выполнение условия останова: $\|\nabla f(\bar{x})\| = \sqrt{b_1^2 + b_2^2} \approx 1,44 > \epsilon$, то есть условие не выполняется.

Рассчитаем величину шагов по каждой координате:

$$h_1 = ab_1\Delta x_1 = 0,1 \cdot (-0,8) \cdot 1 = -0,08$$

$$h_2 = ab_2\Delta x_2 = 0,1 \cdot 1,2 \cdot 1 = 0,12$$

Теперь можно начать двигаться в направлении антиградиента, используя формулы:

$$x_1^{(k)} = 6,8 + 0,08k$$

$$x_2^{(k)} = 7,2 - 0,12k$$

Данные шаги показаны в таблице 2.11:

Таблица 2.11.

Шаг	x_1	x_2	y
0	6,8	7,2	0,2
1	6,88	7,08	0,0464
2	6,96	6,96	0,0016
3	7,04	6,84	0,0656

Из таблицы 2.11 видно, что функция перестала убывать после 2-го шага. Таким образом, новой начальной точкой становится $\bar{x}^{(2)} = (6,96; 6,96)^T$.

Реализуем ПФЭ размерности $N = 2^2 = 4$. Матрица планирования 2.12 будет иметь вид:

Таблица 2.12.

Вершина	x_1	x_2	y
$\bar{x}^{(1)}$	5,96 (-1)	5,96 (-1)	1,0816
$\bar{x}^{(2)}$	7,96 (+1)	5,96 (-1)	5,0816
$\bar{x}^{(3)}$	5,96 (-1)	7,96 (+1)	4,9216
$\bar{x}^{(4)}$	7,96 (+1)	7,96 (+1)	0,9216

Рассчитаем коэффициенты b_i , дающие оценку градиента:

$$b_1 = \frac{1}{4} (-1,0816 + 5,0816 - 4,9216 + 0,9216) = 0$$

$$b_2 = \frac{1}{4} (-1,0816 - 5,0816 + 4,9216 + 0,9216) = -0,08$$

Проверим выполнение условия останова: $\|\nabla f(\bar{x})\| = \sqrt{b_1^2 + b_2^2} = 0,08 < \epsilon$, то есть условие выполняется и поиск нужно остановить. Таким образом, точка $\bar{x} = (6,96; 6,96)^T$ есть точка оптимума.

Траектория поиска изображена на рисунке 2.12.

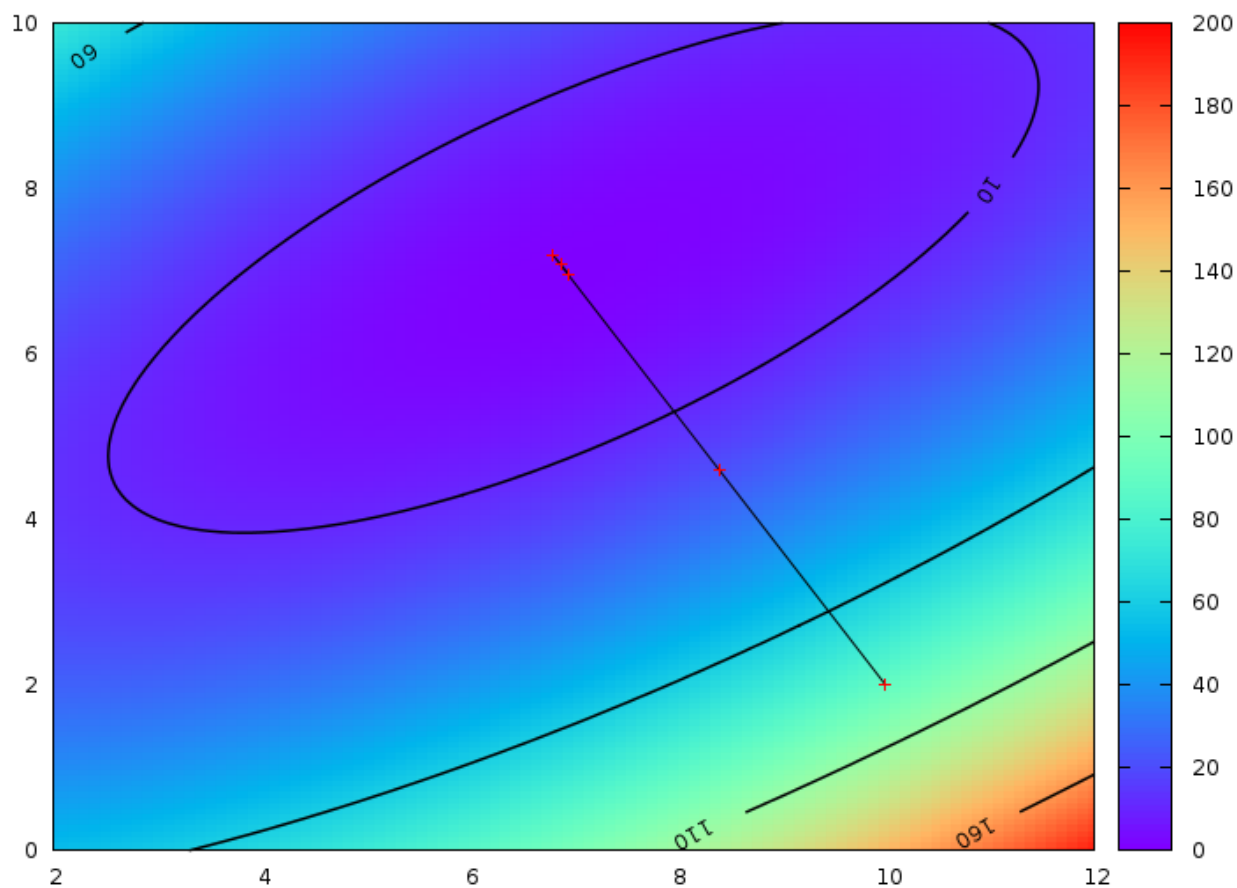


Рис. 2.12. Траектория поиска по методу Бокса-Уилсона

Следует отметить достаточно высокую скорость приближения к оптимальной точке, что обусловлено использованием оценки градиента для выбора направления поиска. На рисунке хорошо видно, что траектория поиска фактически является перпендикуляром к касательным линий уровня.

Глава 3

Линейное программирование

3.1 Постановка задачи и основные определения

Общей задачей линейного программирования называется задача, которая состоит в определении *максимального* (минимального) значения функции

$$F = \sum_{j=1}^n c_j x_j \quad (3.1)$$

при ограничениях

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, k} \quad (3.2)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{k+1, m} \quad (3.3)$$

$$x_j \geq 0, \quad j = \overline{1, l}, \quad l \leq n \quad (3.4)$$

Целевая функция (3.1) называется **линейной формой** задачи (3.1)-(3.4), условия (3.2)-(3.4) называются **ограничениями** данной задачи.

Канонической (основной) задачей линейного программирования (далее - **ОЗЛП**) называется задача, которая состоит в определении максимального значения функции (3.1) при ограничениях (3.3)-(3.4), где $k = 0$ и $l = n$.

Совокупность чисел $(x_1, \dots, x_n) = X$, удовлетворяющих ограничениям (3.2)-(3.4), называется **допустимым решением** или **планом**.

План X^* , при котором функция (3.1) принимает максимальное (минимальное) значение, называется **оптимальным**.

3.2 Переход от общей к основной задаче линейного программирования

Чтобы перейти от общей к основной задаче линейного программирования, нужно в общем случае уметь:

1. Переходить от задачи минимизации к задаче максимизации.

Это делается заменой знака в линейной форме:

$$\min F = c_1x_1 + c_2x_2 + \dots + c_nx_n \Leftrightarrow \max(-F) = -c_1x_1 - c_2x_2 - \dots - c_nx_n$$

2. Переходить от ограничений-неравенств к ограничениям-равенствам.

Неравенство вида « \leq » можно преобразовать в равенство добавлением к его левой части дополнительной неотрицательной переменной:

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \Leftrightarrow a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + x_{n+1} = b_i, \quad x_{n+1} \geq 0$$

Неравенство вида « \geq » можно преобразовать в равенство вычитанием из его левой части дополнительной неотрицательной переменной:

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i \Leftrightarrow a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - x_{n+1} = b_i, \quad x_{n+1} \geq 0$$

3. Делать замену переменных, которые не подчинены условию неотрицательности.

Если некоторая переменная x_j не подчинена условию неотрицательности, то ее заменяют двумя неотрицательными переменными $u_j \geq 0$ и $v_j \geq 0$ (они берутся произвольно):

$$x_j = u_j - v_j.$$

3.3 Существование допустимых решений ОЗЛП

ОЗЛП обязательно должна иметь решение. Может оказаться, что ограничения задачи противоречат друг другу; может оказаться, что решение есть, но не в области отрицательных значений переменных и т.д. Как же узнать, существует или не существует допустимое решение (или решения) ОЗЛП?

Заметим, что ограничения ОЗЛП суть система линейных алгебраических уравнений (СЛАУ):

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{cases} \Leftrightarrow Ax = b,$$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Из курса линейной алгебры известно, что для совместности СЛАУ необходимо и достаточно, чтобы ранг ее матрицы был равен рангу ее расширенной матрицы (рангом матрицы называется наибольший порядок отличного от нуля определителя, который можно получить, вычеркивая из матрицы какие-либо строки и какие-либо столбцы).

Тогда, если система ограничений ОЗЛП совместна, то матрица системы и ее расширенная матрица имеют один и тот же ранг; обозначим его r . Таким образом, r - число линейно-независимых уравнений среди ограничений ОЗЛП.

Очевидно, что ранг системы не может быть больше числа уравнений системы, то есть $r \leq m$. Очевидно, что ранг системы не может быть больше числа переменных задачи, то есть $r \leq n$.

Рассмотрим случай, когда $r = n$. Отбросим линейно-зависимые уравнения-ограничения и получим систему:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

Так как $r = n$, то определитель матрицы этой системы не равен нулю. Из курса линейной алгебры известно, что в этом случае система имеет единственное решение. Если в этом решении хотя бы одна из переменных $x_i, i = \overline{1, n}$ отрицательна, то решение недопустимо и, следовательно, ОЗЛП не имеет решения. Если же все переменные положительны, то найденное решение является допустимым и оптимальным (так как других решений нет).

Очевидно, что такой тривиальный случай малоинтересен, поэтому в дальнейшем будем рассматривать случаи, когда $r < n$, то есть число линейно-независимых ограничений меньше числа переменных в задаче. Тогда, если система ограничений совместна, у нее существует бесчисленное множество решений. При этом $n - r$ переменным можно придавать произвольные значения (будем называть их **свободными** переменными), а остальные r переменных выразятся через них (будем называть эти переменные **базисными**).

В дальнейшем, для простоты, будем полагать все ограничения ОЗЛП линейно-независимыми, то есть $r = m$. Тогда, если $m < n$, система ограничений имеет бесконечно множество решений. Если среди этих решений нет ни одного, для которого все $x_i, i = \overline{1, n}$ неотрицательны, то ОЗЛП не имеет допустимого решения. Если же существуют решения, для которых все переменные положительны, то каждое из них допустимо. Возникает задача отыскания среди них оптимального решения.

Для того, чтобы лучше представить себе ход решения ОЗЛП и различные случаи, которые могут при этом встретиться, удобно воспользоваться геометрической интерпретацией.

3.4 Геометрическая интерпретация ОЗЛП

Рассмотрим случай, когда число переменных на два больше, чем число линейно-независимых ограничений, то есть $n - m = 2$. Тогда можно две переменные выбрать в качестве свободных (пусть это будут x_1 и x_2), а остальные $m = n - 2$ сделать базисными и выразить через свободные.

Получим $m = n - 2$ уравнений вида:

$$\begin{cases} x_3 = \alpha_{31}x_1 + \alpha_{32}x_2 + \beta_3 \geq 0 \\ x_4 = \alpha_{41}x_1 + \alpha_{42}x_2 + \beta_4 \geq 0 \\ \dots \\ x_n = \alpha_{n1}x_1 + \alpha_{n2}x_2 + \beta_n \geq 0 \end{cases}$$

Изобразим на рисунке прямые $x_i = 0, i = \overline{1, n}$ (рис. ??). Свободные переменные в этом случае образуют «координатные оси», а базисные переменные - некоторые прямые в количестве $n - 2$ штук. Так как на все переменные наложено условие неотрицательности, допустимые значения свободных переменных лежат в первой четверти «координатной» плоскости, а допустимые значения базисных переменных - по ту или иную сторону соответствующих прямых (по какую именно - зависит от коэффициентов уравнения). Таким образом, каждая из построенных прямых определяет «допустимую» полуплоскость, на рисунке они обозначены штриховкой. Часть плоскости $x_1 0 x_2$, принадлежащая одновременно всем этим полуплоскостям, образует **область допустимых решений (ОДР)**. ОДР всегда представляет собой выпуклый многоугольник (в общем случае - многогранник).

Теперь возникает вопрос о нахождении из числа допустимых оптимального решения, обращающего в максимум линейную форму. Приведем геометрическую интерпретацию и этой задачи; для этого подставим базисные переменные, выраженные через свободные, в линейную форму, и приведем подобные члены. В результате получится линейная форма, зависящая только от свободных переменных (в первоначальном виде свободного члена нет, но при переходе к свободным переменным он мог появиться):

$$F = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2$$

Очевидно, что данная линейная форма достигает экстремума при тех же значениях переменных, что и линейная форма

$$F^* = \gamma_1 x_1 + \gamma_2 x_2$$

Предположим, что $\gamma_1 > 0, \gamma_2 > 0$. Построим прямую $F^* = 0$ и вектор $\vec{\gamma} = (\gamma_1, \gamma_2)^T$ на уже упомянутой плоскости $x_1 0 x_2$ (рис. ??). Очевидно, что в этом случае линейная форма будет возрастать, если двигать прямую $F^* = 0$ параллельно самой себе в направлении вектора $\vec{\gamma}$. При других знаках коэффициентов γ_1, γ_2 направление возрастания будет меняться, но его всегда указывает вектор $\vec{\gamma}$ (рис. ??).

Итак, для нахождения решения ОЗЛП на основе ее геометрической интерпретации нужно выполнить следующие действия:

1. Построить прямые, уравнения которых представляют собой выражения базисных переменных через свободные переменные.
2. Найти полуплоскости, определяемые ограничениями задачи (те полуплоскости, в которых значения переменных положительны).
3. На пересечении найденных полуплоскостей найти ОДР.
4. Выразить линейную форму через свободные переменные.
5. Построить вектор $\vec{\gamma} = (\gamma_1, \gamma_2)^T$ и прямую $F^* = 0$.
6. Перемещать построенную прямую параллельно самой себе в направлении вектора $\vec{\gamma}$ (для поиска максимума) или в направлении вектора $-\vec{\gamma}$ (для поиска минимума), до тех пор, пока она не пересечет крайнюю точку (точки) ОДР, наиболее удаленную от начала координат в направлении поиска (рис. ??).
7. Определить координаты найденной точки (свободные переменные), а через них найти значения базисных переменных. Найденное таким образом решение ОЗЛП будет оптимальным.

3.5 Пример решения ОЗЛП с помощью геометрической интерпретации

Продemonстрируем описанный в разделе 3.4 алгоритм решения ОЗЛП на конкретном примере. Найдем решение, обращающее в минимум линейную форму

$$F = x_1 - x_2 + 2x_3 - x_4 - 3x_5 + x_6 - 2x_7$$

при ограничениях

$$\begin{cases} x_1 - x_2 + x_3 = 4 \\ 2x_1 - x_2 - x_3 - x_4 = -5 \\ x_1 + x_2 - x_5 = -4 \\ x_2 + x_6 = 5 \\ 2x_1 - 2x_2 - x_6 + 2x_7 = 7 \\ x_i \geq 0, i = 1, \dots, 7 \end{cases}$$

Все уравнения в системе ограничений линейно-независимы, поэтому число свободных переменных равно $n - m = 7 - 5 = 2$. Выберем в качестве свободных переменных x_1 и x_2 , после чего выразим через них остальные (базисные) переменные:

$$\begin{cases} x_3 = -x_1 + x_2 + 4 \\ x_4 = 3x_1 - 2x_2 + 1 \\ x_5 = x_1 + x_2 + 4 \\ x_6 = -x_2 + 5 \\ x_7 = -x_1 + \frac{1}{2}x_2 + 6 \end{cases}$$

Аналогично выразим через свободные переменные линейную форму, после чего отбросим свободный член:

$$F = -5x_1 - 2x_2 - 12$$

$$F^* = -5x_1 - 2x_2$$

Теперь приравняем все полученные уравнения нулю и изобразим полученные прямые на рисунке ???. Определим область допустимых решений и мысленно будем перемещать прямую $F^* = 0$ в направлении, противоположном вектору $\vec{\gamma} = (-5; -2)^T$, так как мы ищем минимум линейной формы. Тогда минимум линейной формы достигается в наиболее удаленной от начала координат точке ОДР. Эта точка (на рисунке она отмечена буквой A), в которой пересекаются прямые $x_6 = 0$ и $x_7 = 0$. Ее координаты можно определить графически, или решив систему из двух уравнений с двумя неизвестными:

$$\begin{cases} x_6 = -x_2 + 5 = 0 \\ x_7 = -x_1 + \frac{1}{2}x_2 + 6 = 0 \end{cases}$$

Отсюда получаем значения свободных переменных $x_1 = 8,5$ и $x_2 = 5$, а потом вычисляем значения базисных переменных: $x_3 = 0,5$, $x_4 = 16,5$, $x_5 = 17,5$, $x_6 = 0$, $x_7 = 0$.

3.6 Некоторые свойства решений ОЗЛП. Особые случаи решения ОЗЛП

Отметим некоторые особенности для случая $n - m = 2$:

1. Решение ОЗЛП, если оно существует, не может лежать внутри ОДР, а только на ее границе.

2. ОЗЛ может иметь бесконечное множество оптимальных решений. Это происходит в том случае, когда прямая $F^* = 0$ параллельна той стороне ОДР, где достигается оптимальное значение линейной формы - это означает, что оптимальное решение достигается не в одной точке, а на всей этой стороне (например, на рисунке ?? оптимальное значение достигается в любой точке отрезка AB).
3. ОЗЛП может не иметь решения, даже когда существует ОДР. Это происходит в том случае, если в направлении вектора $\vec{\gamma}$ ОДР неограничена (рис. ??) или система ограничений несовместна (рис. ??).
4. Оптимальное решение всегда достигается в одной из вершин ОДР (см. п. 1); если оно достигается на целой стороне (см. п. 2), то оно же достигается и в каждой из вершин, через которые проходит эта сторона. Решение, лежащее в одной из вершин ОДР, называется **опорным решением (опорным планом, базисным решением)**.
5. Поскольку оптимальное решение лежит в одной из вершин ОДР (см. п. 4), то обязательно несколько его составляющих будут равны нулю. Если количество таких составляющих равно $n - m$, то решение называют **невырожденным**; если количество таких составляющих больше, чем $n - m$, то решение называют **вырожденным**.

Все приведенные утверждения справедливы и для случая $n - m > 2$; в этом случае изменятся только терминология. Например, в случае $n - m = 3$ ОДР будет не многоугольником, а многогранником; ограничивающие прямые превратятся в плоскости. В случае $n - m > 3$ ОДР станет «гипермногогранником», а ее границы - «гиперплоскостями». Однако принцип нахождения оптимального решения не изменится, так как оно по-прежнему будет лежать в одной из вершин упомянутого «гипермногогранника». Поэтому, исходя из утверждений п. 4, для нахождения оптимального опорного решения необходимо уметь отличать его от прочих опорных решений.

3.7 Признак оптимальности опорного решения

Предположим, что имеется некоторое опорное решение; как известно, часть переменных в нем базисные, часть - свободные. Если выразить линейную форму через свободные переменные:

$$F = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \dots + \gamma_k x_k,$$

то признаком оптимальности опорного решения является $\gamma_i \leq 0$, $i = \overline{1, k}$ (для задачи поиска максимума) или $\gamma_i \geq 0$, $i = \overline{1, k}$ (для задачи поиска минимума).

Указанное свойство следует из условия неотрицательности всех переменных задачи. Предположим, что мы ищем максимум линейной формы; положим значения всех свободных переменных равными нижней допустимой границе - нулю (мы можем это сделать по определению, т.к. свободные переменные могут принимать любые неотрицательные значения).

Тогда значение линейной формы будет равно $F = \gamma_0$. Теперь попробуем изменить значение любой свободной переменной x_i ; делать это можно, только увеличивая ее. Если соответствующий ей коэффициент $\gamma_i \leq 0$, то значение линейной формы в общем случае уменьшится: $F = \gamma_0 + \gamma_i x_i \leq \gamma_0$, что не соответствует поставленной задаче поиска максимума. Если же все $\gamma_i \leq 0$, $i = \overline{1, k}$, то увеличение любой переменной $x_i \leq 0$, $i = \overline{1, k}$ приведет к уменьшению значения линейной формы. Таким образом, если в случае $\gamma_i \leq 0$, $i = \overline{1, k}$ положить $x_i = 0$, $i = \overline{1, k}$, то найденное опорное решение будет оптимальным. Для доказательства в случае поиска минимума проводятся аналогичные рассуждения.

Указанное свойство подсказывает способ решения ОЗЛП, который можно автоматизировать: обменивать местами свободные и базисные переменные до тех пор, пока не будет выполняться условие оптимальности опорного решения; как только условие будет выполнено, положить все свободные переменные равными нулю и вычислить базисные переменные - найденное таким образом решение будет оптимальным. На этой идее основан симплекс-метод решения задач линейного программирования.

3.8 Симплекс-метод решения задач линейного программирования

Геометрическая интерпретация перестает быть пригодной для решения задач линейного программирования при числе свободных переменных $n - m > 3$, а при $n - m = 3$ затруднительна. Для нахождения решения задачи линейного программирования при произвольном числе свободных переменных применяются вычислительные методы, наиболее универсальным из которых является симплекс-метод.

Нахождение решения с помощью симплекс-метода делится на два этапа:

1. Отыскание опорного решения.
2. Перебор опорных решений, начиная с найденного в п. 1, с целью определения оптимального решения.

В процессе первого этапа выясняется, имеет ли вообще данная задача допустимые решения. В процессе второго этапа попутно выясняется, ограничена ли область допустимых решений.

3.8.1 Отыскание опорного решения

Обозначим для удобства свободные переменные через x_i , $i = \overline{1, k}$, а базисные - через y_j , $j = \overline{1, l}$. Выразим базисные переменные через свободные, записав при этом каждое уравнение системы ограничений в специальной форме:

$$\begin{cases} y_1 = b_1 - (a_{11}x_1 + a_{12}x_2 + \dots + a_{1k}x_k) \\ y_2 = b_2 - (a_{21}x_1 + a_{22}x_2 + \dots + a_{2k}x_k) \\ \dots \\ y_l = b_l - (a_{l1}x_1 + a_{l2}x_2 + \dots + a_{lk}x_k) \end{cases} \quad (3.5)$$

В дальнейшем запись базисных переменных в виде 3.5 будем называть **стандартной**. Попробуем получить опорное решение, полагая все $x_i = 0$, $i = \overline{1, k}$. Тогда $y_j = b_j$, $j = \overline{1, l}$. Если все $b_j \geq 0$, то опорное решение получено. Если же есть некоторые $b_j < 0$, то придется искать опорное решение путем обмена свободных и базисных переменных до тех пор, пока опорное решение не будет найдено, или пока не будет доказано, что задача не имеет допустимых решений. Эти обмены удобно выполнять с помощью симплекс-таблиц, схематичный вид которых приведен на рисунке ??.

Алгоритм обмена свободных и базисных переменных с помощью симплекс-таблицы

Предположим, что требуется вывести из состава свободных переменных x_i , а на ее место ввести базисную переменную y_j . С помощью симплекс-таблицы это делается легко алгоритмируемым способом.

1. Выделяем в таблице **разрешающий элемент** a_{ji} . Строка, в которой находится разрешающий элемент, будет **разрешающей строкой**, а столбец - **разрешающим столбцом**.
2. Вычисляем обратную величину разрешающего элемента $\lambda = \frac{1}{a_{ji}}$ и записываем ее в правом нижнем углу той ячейки таблицы, в которой находится разрешающий элемент.
3. Все элементы разрешающей строки (за исключением самого разрешающего элемента) умножаем на λ и результат записываем в правом нижнем углу соответствующих ячеек таблицы.
4. Все элементы разрешающего столбца (за исключением самого разрешающего элемента) умножаем на $-\lambda$ и результат записываем в правом нижнем углу соответствующих ячеек таблицы.
5. Выделяем в разрешающей строке все числа, стоящие в левом верхнем углу («прежние» элементы), а в разрешающем столбце - все числа, стоящие в правом нижнем углу («новые» элементы). Указанные действия не затрагивают сам разрешающий элемент.
6. Для каждого элемента a_{km} , не принадлежащего ни к разрешающей строке, ни к разрешающему столбцу, записываем в правый нижний угол соответствующей ячейки произведение выделенных чисел, находящихся в k -ой строке и m -ом столбце.
7. Переписываем таблицу, заменив:

- x_i на y_j и обратно;

- элементы разрешающей строки и разрешающего столбца - числами, стоящими в правых нижних углах тех же ячеек;
- каждый из остальных элементов - суммой чисел, стоящих в верхнем и нижнем углах соответствующих ячеек.

Пример. В системе уравнений произвести замену x_1 на y_2 и обратно:

$$\begin{cases} y_1 = x_1 - x_2 + 2x_3 - 5 \\ y_2 = 2x_1 - x_2 + 1 \\ y_3 = 2x_2 - x_3 - 1 \\ y_4 = -x_1 - x_3 + 2 \end{cases}$$

Для удобства составления симплекс-таблицы перепишем систему в стандартном виде (3.5):

$$\begin{cases} y_1 = -5 - (-x_1 + x_2 - 2x_3) \\ y_2 = 1 - (-2x_1 + x_2) \\ y_3 = -1 - (-2x_2 + x_3) \\ y_4 = 2 - (x_1 + x_3) \end{cases}$$

Составим по записанной системе симплекс-таблицу, выделим в ней разрешающий элемент (он заключен в круг) и найдем его обратную величину λ (рис. ??).

На рис. ?? показана симплекс-таблица после умножения элементов разрешающей строки и разрешающего столбца на λ и $-\lambda$ соответственно; в квадрат заключены элементы, необходимые для дозаполнения ячеек, не принадлежащих к разрешающей строке и разрешающему столбцу.

Симплекс-таблица, полностью готовая к замене переменных, показана на рисунке ??.

Окончательный вид новой симплекс-таблицы после замены переменных приведен на рисунке ??.

Алгоритм выбора разрешающего элемента

Имея знания об обмене свободных и базисных переменных, вернемся к отысканию опорного решения. Как уже было сказано, необходимо с помощью подобных обменов добиться ситуации, когда все свободные члены неотрицательны: $b_j \geq 0$. Для этого нужно научиться выбирать разрешающие элементы таким образом, чтобы обмен переменных приводил нас к требуемой ситуации как можно быстрее.

1. Выбираем в симплекс-таблице любую строку с отрицательным свободным членом.
2. Ищем в этой строке любой отрицательный элемент a_{ji} . Если таких элементов в строке нет, то это признак того, что система ограничений несовместна с условиями неотрицательности переменных $x_i \geq 0, y_j \geq 0$; то есть задача не имеет допустимых решений.

3. Если отрицательный элемент в строке есть, то столбец, в котором он находится, выбирается в качестве разрешающего.
4. Отбираем строки, в которых свободный член и элемент разрешающего столбца имеют одинаковые знаки; если свободный член или элемент разрешающего столбца равен нулю, то такая строка не отбирается. Для отобранных таким образом строк вычисляется отношение свободного члена к элементу разрешающего столбца.
5. Разрешающей строкой выбирается та, для которой полученное на предыдущем шаге отношение минимально.
6. Пользуясь приведенным выше алгоритмом, совершаем обмен свободной и базисной переменных, соответствующих разрешающему столбцу и строке соответственно.
7. Повторяем шаги 1-6 до тех пор, пока все свободные члены не станут неотрицательными или пока не убедимся, что система ограничений противоречит условию неотрицательности переменных (см. п. 2).

Пример. Найти опорное решение для задачи линейного программирования с ограничениями

$$\begin{cases} y_1 = 1 - (-x_1 - 2x_2 + x_3) \\ y_2 = -5 - (-2x_1 + x_2 - x_3) \\ y_3 = 2 - (x_1 + x_2) \\ y_4 = 1 - (-x_2 + x_3) \end{cases}$$

Составим по записанной системе симплекс-таблицу, выделим в ней разрешающий столбец и найдем отношения для тех строк, где знаки свободного члена и элемента разрешающего столбца совпадают (рис. ??).

Выбираем в качестве разрешающей строки ту, где найденное выше отношение минимально; на пересечении разрешающей строки и разрешающего столбца определяем разрешающий элемент и находим его обратную величину λ (рис. ??).

Делаем замену соответствующих переменных, в результате чего получаем новую симплекс-таблицу (рис. ?? - ??).

Видно, что значение отрицательного свободного члена увеличилось, но пока еще остается недопустимым. Повторное применение алгоритма выбора разрешающего элемента и замены соответствующих переменных дает симплекс-таблицу, в которой все свободные члены неотрицательны (рис. ??).

Теперь, полагая свободные переменные (те переменные, обозначения которых стоят в столбцах последней симплекс-таблицы) равными нулю, а базисные переменные (те переменные, обозначения которых стоят в строках последней симплекс-таблицы) равными соответствующим свободным членам, находим опорное решение.

Глава 4

Комбинаторная оптимизация

Глава 5

Стохастическая оптимизация

Литература