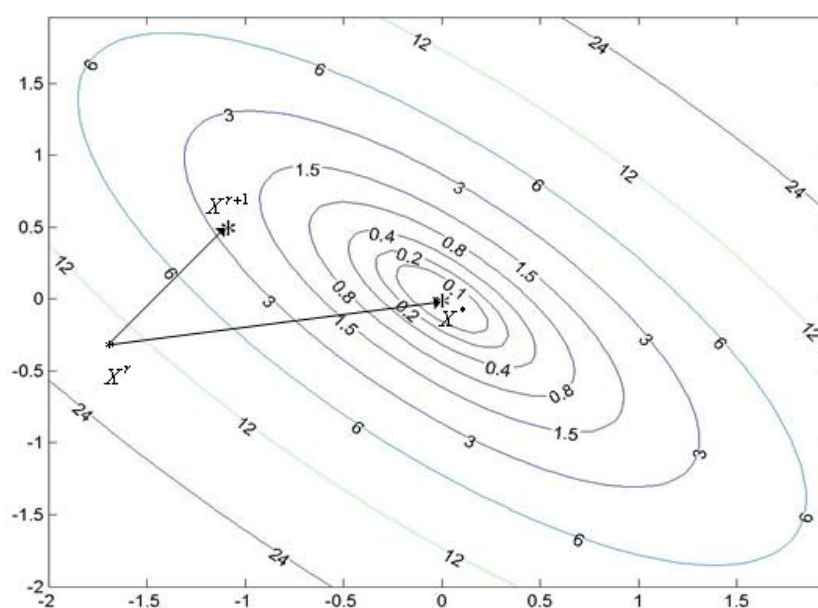


Е.Л. Богданова,

К.А. Соловейчик, К.Г. Аркина

ОПТИМИЗАЦИЯ В ПРОЕКТНОМ МЕНЕДЖМЕНТЕ: НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

Е.Л. Богданова

К.А. Соловейчик

К.Г. Аркина

**ОПТИМИЗАЦИЯ В ПРОЕКТНОМ
МЕНЕДЖМЕНТЕ: НЕЛИНЕЙНОЕ
ПРОГРАММИРОВАНИЕ**

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО

по направлению подготовки (специальности)

27.04.08 «Управление интеллектуальной собственностью»

в качестве учебно-методического пособия для реализации основных
профессиональных образовательных программ высшего образования
магистратуры



Санкт-Петербург

2017

Богданова Е.Л. Оптимизация в проектном менеджменте: нелинейное программирование: учебное пособие / Е.Л. Богданова, К.А. Соловейчик, К.Г. Аркина. – СПб.: Университет ИТМО, 2017. – 190 с.

Рецензенты: Аркин Павел Александрович, д.э.н, профессор, заместитель генерального директора по инновациям ООО «ХОЛДИНГ ЛЕНПОЛИГРАФМАШ», профессор базовой кафедры «Процессы управления наукоемкими производствами» ФГАОУВО «Санкт-Петербургский политехнический университет Петра Великого».

Якубсон Михаил Яковлевич, к.ф.-м.н., доцент, доцент кафедры математического анализа ФГБОУВО «Российский государственный педагогический университет имени А.И. Герцена».

Учебное пособие соответствует содержанию блока 1 структуры программы магистратуры федерального государственного образовательного стандарта высшего образования по направлению подготовки 27.04.08 - Управление интеллектуальной собственностью (уровень магистратуры). Пособие имеет учебно-методическое назначение и предназначено для изучения студентами материала раздела нелинейное программирование дисциплины учебного плана «Проектный менеджмент». В пособии рассмотрены основные методы вычислений раздела исследования операций: нелинейное программирование на основании примеров в области экономики предприятия.

Рекомендовано к печати УМС ИМБИП, протокол № 5 от «10» мая 2017 г.



Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2017

©Богданова Е.Л., Соловейчик К.А., Аркина К.Г., 2017

Рецензенты:

Аркин Павел Александрович – доктор экономических наук, профессор, профессор базовой кафедры «Процессы управления наукоемкими производствами» ФГАОУВО «Санкт-Петербургский политехнический университет Петра Великого», заместитель генерального директора по инновациям ООО «ХОЛДИНГ ЛЕНПОЛИГРАФМАШ»

Якубсон Михаил Яковлевич – кандидат физико-математических наук, доцент, доцент кафедры математического анализа ФГБОУВО «Российский государственный педагогический университет имени А.И. Герцена»

Содержание

Введение.....	6
---------------	---

Раздел 1 Нелинейное программирование. Задачи оптимизации

1.1 Постановка задачи оптимизации.....	10
1.2 Классификация критериев оптимальности.....	16
1.3 Классификация задач оптимизации.....	22
1.4 Одномерная задача оптимизации.....	24
1.5 Многомерная задача безусловной оптимизации.....	26
1.6. Задача выпуклого программирования.....	27
1.7 Задача нелинейного программирования с ограничениями типа равенств.....	30
1.8 Теорема Куна-Таккера для задачи нелинейного программирования с ограничениями типа неравенств.....	33
1.9 Теорема Куна-Таккера для общей задачи нелинейного программирования.....	37
1.10 Аналитическое решение многомерных задач нелинейного программирования.....	38
1.11 Классификация методов решения задач оптимизации.....	43

Раздел 2 Нелинейное программирование. Безусловная одномерная оптимизация

2.1 Алгоритм равномерного поиска.....	46
2.2 Алгоритм деления пополам.....	48
2.3 Алгоритм Фибоначчи.....	49
2.4 Алгоритм золотого сечения.....	55
2.5 Метод хорд (метод секущих).....	57
2.6 Метод касательных (метод Ньютона).....	59
2.7 Метод перебора.....	61
2.8 Одномерный метод Монте-Карло.....	62
2.9 Метод выделения интервалов унимодальности.....	63
2.10 Метод аппроксимирующих моделей.....	66

Раздел 3 Нелинейное программирование. Безусловная многомерная оптимизация

3.1 Метод Гаусса-Зейделя.....	70
3.2 Метод Хука-Дживса.....	75
3.3 Метод Розенброка.....	79
3.4 Метод сопряженных направлений.....	82
3.5 Симплекс-метод.....	86

3.6	Метод деформируемого многогранника (Нелдера-Мида).....	94
3.7	Метод наискорейшего спуска. Метод дробления шага.....	100
3.8	Метод оптимизации Ньютона.....	111
3.9	Метод с возвратом при неудачном шаге. Метод наилучшей пробы.....	116
3.10	Метод комплексов.....	119
3.11	Метод повторяющегося случайного поиска.....	124
3.12	Метод случайного поиска с постоянным радиусом поиска и случайными направлениями.....	128

Раздел 4 Нелинейное программирование. Многомерная условная оптимизация

4.1	Методы последовательной безусловной оптимизации.....	132
4.2	Метод скользящего допуска.....	138
4.3	Модифицированный метод комплексов.....	146
4.4	Метод линейной аппроксимации.....	154
4.5	Метод проекции градиента.....	160
4.6	Метод сведения к совокупности вложенных задач глобальной одномерной минимизации.....	164
4.7	Метод сведения к задаче одномерной глобальной оптимизации с помощью развертки Пеано.....	169
4.8	Метод Монте-Карло.....	176

	Список литературы.....	178
--	------------------------	-----

ВВЕДЕНИЕ

При решении конкретной задачи оптимизации при изучении дисциплины «Проектный менеджмент» студент прежде всего должен выбрать математический метод, который приводил бы к конечным результатам с наименьшими затратами на вычисления или же давал возможность получить наибольший объем информации об искомом решении. Выбор того или иного метода в значительной степени определяется постановкой задачи, а также используемой математической моделью. Нельзя рекомендовать какой-либо один метод, который можно использовать для решения всех без исключения задач, возникающих на практике. Одни методы в этом отношении являются более, а другие – менее общими. Целую группу методов на определенных этапах решения задачи можно применять в сочетании с другими методами. Некоторые методы специально разработаны или наилучшим образом подходят для решения оптимальных задач с математическими моделями определенного вида.

Так, математический аппарат линейного программирования, специально создан для решения задач с линейными критериями оптимальности и линейными ограничениями на переменные и позволяет решать большинство задач, сформулированных в такой постановке.

Пожалуй, наилучшим путем при выборе способа решения, наиболее пригодного для соответствующей задачи, следует признать исследование возможностей и опыта применения различных методов оптимизации. Дадим краткую характеристику рассматриваемых в учебном пособии методов и областей их применения, что до некоторой степени может облегчить выбор того или иного метода для решения конкретной задачи.

Нелинейное программирование применяют для решения оптимальных задач с нелинейными функциями цели. На независимые переменные могут быть наложены ограничения также в виде нелинейных соотношений, имеющих вид равенств или неравенств. Общих способов решения задач нелинейного программирования не существует, способ решения выбирается в каждом конкретном случае в зависимости от вида целевой функции и накладываемых на элементы решения ограничений. В отличие от задач линейного программирования, методы решения которых хорошо отлажены, устоялись и не представляют существенных сложностей, задачи нелинейного программирования принадлежат к трудным вычислительным задачам. Названием **методы нелинейного программирования** объединяется большая группа численных методов, многие из которых приспособлены для решения оптимальных задач соответствующего класса. Выбор того или иного метода обусловлен сложностью вычисления критерия оптимальности и сложностью ограничивающих условий, необходимой точностью решения, мощностью имеющейся вычислительной машины и т.д.

Все известные численные методы отыскания экстремума нелинейных функций в зависимости от используемой информации для получения следующей точки можно разделить на три основные группы:

- прямые методы, использующие только текущие значения оптимизируемой функции;
- методы первого порядка, использующие дополнительно и значения первой производной функции;
- методы второго порядка, требующие знания и второй производной оптимизируемой функции.

Задачи нелинейного программирования можно разделить на две группы: задачи, в которых на независимые переменные накладываются некоторые ограничения, решаются с помощью методов **условной оптимизации**; задачи, в которых поиск максимума или минимума нелинейной целевой функции не содержит дополнительных ограничений, решаются с помощью методов **безусловной оптимизации**. В области нелинейного программирования с ограничениями (условная оптимизация) методы решения задач менее разработаны по сравнению с областью нелинейного программирования без ограничений (**безусловная оптимизация**). Здесь встречаются большие трудности по той причине, что искомое решение должно подчиняться дополнительным требованиям, выраженным в виде ограничений. Многие алгоритмы решения задачи с ограничениями предполагают сведение ее к последовательности задач безусловной оптимизации. Другой класс методов основан на поиске подходящего направления и последующей минимизации вдоль этого направления. Обоснование методов безусловной оптимизации может быть естественным образом распространено на обоснование процедур решения задач с ограничениями.

В свою очередь, методы безусловной оптимизации делятся на методы **одномерной** и **многомерной** оптимизации. Несмотря на то, что безусловная оптимизация функции одной переменной – наиболее простой тип задач, она занимает центральное место, как с теоретической, так и с практической точек зрения. Это связано с тем, что задачи однопараметрической оптимизации достаточно часто встречаются в инженерной практике и, кроме того, находят свое применение при реализации более сложных итеративных процедур многопараметрической оптимизации.

Теория игр – раздел математики для изучения конфликтных ситуаций. Это значит, что с помощью теории игр можно выработать оптимальные правила поведения каждой стороны, участвующей в решении конфликтной ситуации.

На практике часто появляется необходимость согласования действий фирм, объединении, министерств и других участников проектов в случаях, когда их интересы не совпадают. В таких ситуациях **теория игр** позволяет найти лучшее решение для поведения участников, обязанных согласовывать действия при столкновении интересов. Теория игр все шире проникает в практику экономических решений и исследований. Ее можно рассматривать как инструмент, помогающий повысить эффективность плановых и управленческих решений. Это имеет большое значение при решении задач в промышленности, сельском хозяйстве, на транспорте, в торговле, особенно при заключении договоров с иностранными государствами на любых иерархических уровнях. Так, можно определить научно обоснованные уровни снижения розничных цен и оптимальный

уровень товарных запасов, решать задачи экскурсионного обслуживания и выбора новых линий городского транспорта, задачу планирования порядка организации эксплуатации месторождений полезных ископаемых в стране и др.

В экономике, например, оказался недостаточным аппарат математического анализа, занимающийся определением экстремумов функций. Появилась необходимость изучения так называемых оптимальных минимаксных и максиминных решений. Следовательно, **теорию игр** можно рассматривать как новый раздел оптимизационного подхода, позволяющего решать новые задачи при принятии решений.

Любая сфера человеческой деятельности связана с принятием решений в условиях неполноты информации. Источники неопределенности могут быть самые разнообразные: нестабильность экономической и/или политической ситуации, неопределенность действий партнеров по бизнесу, случайные факторы, т.е. большое число обстоятельств, учесть которые не представляется возможным (например, погодные условия, неопределенность спроса на товары, неабсолютная надежность процессов производства, неточность информации и др.). Решения с учетом перечисленных и множества других неопределенных факторов возможно принимать в рамках так называемой **теории игр** – аналитического подхода к выбору наилучшего действия (альтернативы) или последовательности действий. В зависимости от степени определенности возможных исходов или последствий различных действий, с которыми сталкивается лицо, принимающее решение (ЛПР), в теории принятия решений рассматриваются три типа моделей:

- выбор решений в условиях определенности, если относительно каждого действия известно, что оно неизменно приводит к некоторому конкретному исходу;
- выбор решения при риске, если каждое действие приводит к одному из множества возможных частных исходов, причем каждый исход имеет вычисляемую или экспертно оцениваемую вероятность появления. Предполагается, что ЛПР эти вероятности известны или их можно определить путем экспертных оценок;
- выбор решений при неопределенности, когда то или иное действие или несколько действий имеют своим следствием множество частных исходов, но их вероятности совершенно не известны или не имеют смысла.

Разделы 1-4 данного пособия посвящены нелинейному программированию. В разделах 2 и 3 рассмотрены методы безусловной оптимизации нелинейного программирования. В разделе 2 приведены поисковые методы одномерной оптимизации без ограничений. Рассмотренные в параграфах 2.7-2.10 методы относятся к поиску глобального минимума одномерных многоэкстремальных функций. В разделе 3 изложены методы многомерной безусловной оптимизации: в параграфах 3.1-3.6 – прямые методы; 3.7-3.8 – методы первого и второго порядка; и, наконец, в параграфах 3.9-3.12 – методы случайного поиска. Далее, в разделе 4, параграфы 4.1-4.5 посвящены методам локальной, а 4.6-4.8 – глобальной многомерной условной оптимизации.

Дисциплина Проектный менеджмент при подготовке магистров по направлению 27.04.08 – Управление интеллектуальной собственностью охватывает базовые основы науки исследования операций, в том числе нелинейное програм-

мирование. В ходе дальнейшего обучения отдельные разделы теории исследования операций будут изучены студентами в других дисциплинах.

РАЗДЕЛ 1 НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ. ЗАДАЧИ ОПТИМИЗАЦИИ

1.1 Математическая формулировка задачи оптимизации

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

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

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

$$Y = F(X, Q),$$

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

Если внешние параметры Q известны и фиксированы, то $Y = F(X)$.

Такая модель называется **детерминированной моделью объекта проектирования** (в том смысле, что значения Y однозначно определяются значениями X).

Модель, в которой внешние параметры Q являются случайными величинами, называется **стохастической моделью объекта проектирования** или вероятностной моделью объекта проектирования.

Варьируемые при оптимизации параметры называются **управляемыми параметрами** (варьируемыми параметрами) или переменными.

Будем их также обозначать вектором X и называть **вектором управляемых параметров** (вектором варьируемых параметров) или переменных. Важно понимать, что в этот вектор не обязательно включаются все внутренние параметры объекта проектирования.

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

$$x_i^- \leq x_i \leq x_i^+, \quad i \in [1, \dots, n] \quad (1.1.1)$$

$$y_j^- \leq y_j \leq y_j^+, \quad j \in [1, \dots, m] \quad (1.1.2)$$

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

Поскольку существует функциональная связь, ограничения (1.1.2) эквивалентны системе:

$$\{g_j(X), j \in [1, \dots, m]\} = g(X) \geq 0, \quad (1.1.3)$$

где $g(X)$ – m -мерная вектор-функция. Заметим, что в виде (1.1.3) можно записать и ограничения типа равенств $g(X) = 0$ путем замены их парой неравенств:

$$\begin{cases} g(x) \geq 0 \\ -g(x) \geq 0. \end{cases}$$

Одной из особенностей задач проектирования является то, что в систему ограничений (1.1.3) могут входить функции, зависящие от одной из компонент вектора Q – некоторого параметра q , заданного на интервале $[q_{\min}, q_{\max}]$. Таким параметром может быть время, частота, температура и т.п. В этом случае:

$$g(X, q) \geq 0, \quad [q_{\min}, q_{\max}] \quad (1.1.4)$$

Для перехода от (1.1.4) к (1.1.3) можно использовать: 1) сеточный метод, 2) принцип гарантированного результата.

Идея сеточного метода основана на дискретизации интервала $[q_{\min}, q_{\max}]$ и рассмотрении функции $g(X, q)$ на дискретной совокупности точек (см. рис. 1.1.1).



Рис. 1.1.1. Дискретизация интервала $[q_{\min}, q_{\max}]$

При этом выполнение ограничений (1.1.4) сводится к требованию выполнения системы из M неравенств:

$$g(X, q_i) \geq 0, \quad i = 1, 2, \dots, M. \quad (1.1.5)$$

Недостатки подхода: трудно обоснованно выбрать число M ; вместо одного неравенства (1.1.4) приходится рассматривать систему из M неравенств (1.1.5).

Принцип гарантированного результата состоит в том, что ограничения (1.1.4) проверяются для наиболее неблагоприятного значения параметра $q^* \in [q_{\min}, q_{\max}]$: $g(X, q^*) = \min_{q \in [q_{\min}, q_{\max}]} g(X, q)$.

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

Далее будем полагать, что ограничения на параметр q , заданные на интервале $[q_{\min}, q_{\max}]$, с помощью сеточного метода или принципа гарантированного результата сведены к системе неравенств (1.1.3). Тогда условие (1.1.3) определяет множество допустимых значений вектора $X : D_g = \{X / g(X) \geq 0\}$.

Ограничения (1.1.1) определяют следующее множество допустимых значений вектора $X : D_X = \{x / x_i^- \leq x_i \leq x_i^+, i \in [1, \dots, n]\}$.

Множество, полученное пересечением множеств D_g и D_X , будем называть **множеством допустимых значений вектора управляемых параметров X** , т.е. $D = D_g \cap D_X$.

Любой вектор управляемых переменных $X \in D$ называется **допустимым вектором управляемых параметров**.

Если вектор $X \in D$, то будем такой вектор называть также точкой множества D .

Если не оговорено противное, множество допустимых значений вектора варьируемых параметров D является ограниченным и замкнутым (компактным).

Чаще всего ограничения (1.1.1) – (1.1.2) записывают единообразно – в виде ограничений вида $g(X) \geq 0$. Будем называть такие ограничения ограничениями типа неравенств, а функции $g(X)$ – **ограничивающими функциями**.

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

$$D = \{X / g(X) \geq 0\} = \{X / g_i(X) \geq 0, i \in [1, \dots, m]\}. \quad (1.1.6)$$

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

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

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

Широкий класс методов оптимизации ориентирован на решение задач оптимизации, у которых множество допустимых значений вектора варьируемых параметров является выпуклым множеством.

Множество допустимых значений вектора варьируемых параметров D называется **выпуклым множеством допустимых значений вектора ва-**

варьируемых параметров, если для любых точек $X_1, X_2 \in D$ и для любого $\lambda \in [0, 1]$ выполняется соотношение $\lambda X_1 + (1 - \lambda) X_2 \in D$.

Например, на рис. 1.1.2, который иллюстрирует двумерный случай ($n = 2$), все точки отрезка $[X_1, X_2]$ принадлежат множеству D и поэтому это множество выпукло.

В противном случае множество допустимых значений D называется не выпуклым множеством допустимых значений вектора варьируемых параметров. Например, на рис. 1.1.3 часть $[A, B]$ отрезка $[X_1, X_2]$ не принадлежат множеству D , которое поэтому не является выпуклым.

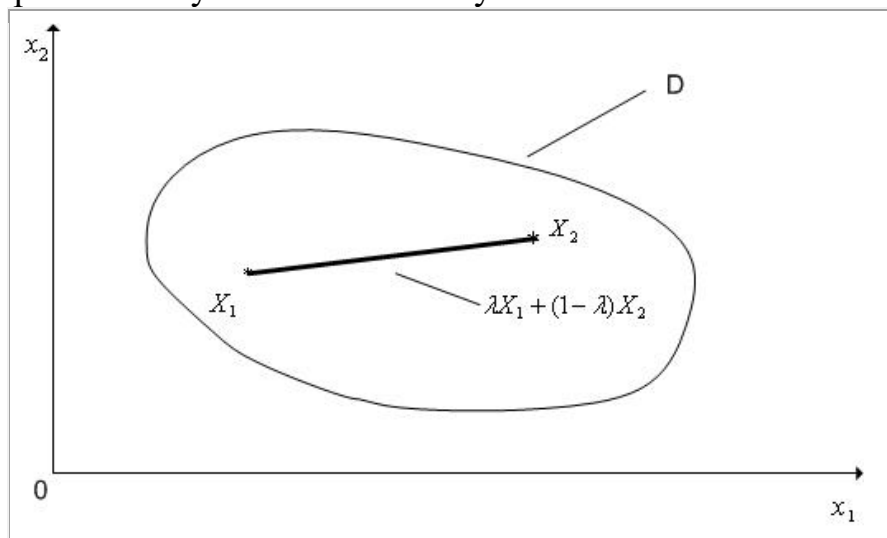


Рис. 1.1.2. К определению выпуклого множества ($n = 2$)

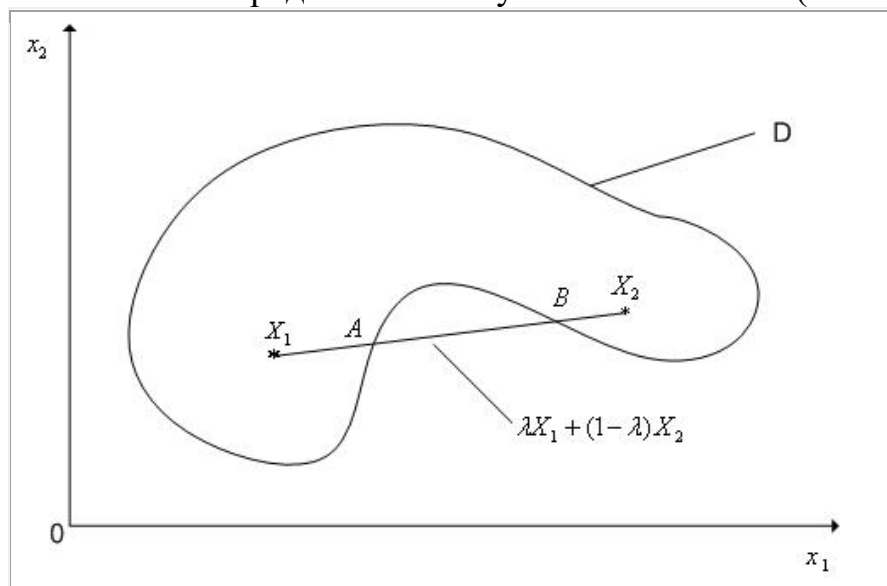


Рис. 1.1.3. Пример не выпуклого множества ($n = 2$)

Множество $L(X_1, X_2) = \{X / X = \lambda X_1 + (1 - \lambda) X_2 \in D, \lambda \in [0, 1]\}$ будем называть отрезком с концами X_1, X_2 и обозначать $[X_1, X_2]$.

На основе введенного понятия, можно дать другое определение выпуклого множества D . Множество допустимых значений вектора варьируемых парамет-

ров D называется выпуклым множеством, если оно наряду с любыми точками $X_1, X_2 \in D$ содержит в себе также отрезок $[X_1, X_2]$.

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

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

Детерминированная задача оптимизации формулируется следующим образом:

$$\min_{X \in D} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (1.1.7)$$

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

Задача оптимизации, в которой критерий оптимальности $\Phi(X)$ и/или ограничивающие функции $g(X)$ зависят от случайного вектора внешних параметров Q , называется **стохастической задачей оптимизации**.

Заметим, что задача максимизации критерия оптимальности $\Phi(X)$ сводится к задаче минимизации критерия $(-\Phi(X))$: $\max_{X \in D} \Phi(X) = \min_{X \in D} (-\Phi(X))$.

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

Из курса математического анализа известна следующая теорема.

Теорема 1.1.1. (теорема Вейерштрасса). Если функция $\Phi(X)$ определена и непрерывна в ограниченной замкнутой области D , то она достигает в этой области своего наименьшего и наибольшего значений.

Таким образом, детерминированная задача оптимизации (1.1.7) имеет решение.

Если случайный вектор внешних параметров Q не входит в критерий оптимальности $\Phi(X)$, то этот критерий называется **детерминированным критерием оптимальности**. Если критерий оптимальности имеет вид $\Phi(X, Q)$, где Q – случайный вектор внешних параметров, то этот критерий называется **стохастическим критерием оптимальности**.

Дадим определения некоторых свойств функций.

Точка X^* называется **точкой локального минимума** функции $\Phi(X)$, если для всех точек X , принадлежащих некоторой малой окрестности $d(X)$ точки X^* имеем $\Phi(X^*) \leq \Phi(X), X \in d(X^*) \in D$.

Значение функции $\Phi(X)$ в точке локального минимума называется **локальным минимумом** функции $\Phi(X)$.

Таким образом, если точка X^* является точкой локального минимума функции $\Phi(X)$, то величина $\Phi(X^*)$ есть локальный минимум этой функции.

Точка X^* называется **точкой глобального минимума** функции $\Phi(X)$, если $\Phi(X^*) \leq \Phi(X), X \in D$.

Таким образом, точка наименьшего из всех локальных минимумов называется точкой глобального минимума функции $\Phi(X)$.

Соответствующее значение функции $\Phi(X)$ называется **глобальным минимумом** этой функции.

Например, на рис. 1.1.4, который иллюстрирует одномерный случай ($n=1$) x_1^*, x_2^*, x_3^* – точки локального минимума функции $\Phi(X)$, а величины $\Phi_1^*, \Phi_2^*, \Phi_3^*$ – соответствующие локальные минимумы этой функции, x_3^* – точка глобального минимума функции $\Phi(X)$, а Φ_3^* – глобальный минимум этой функции.

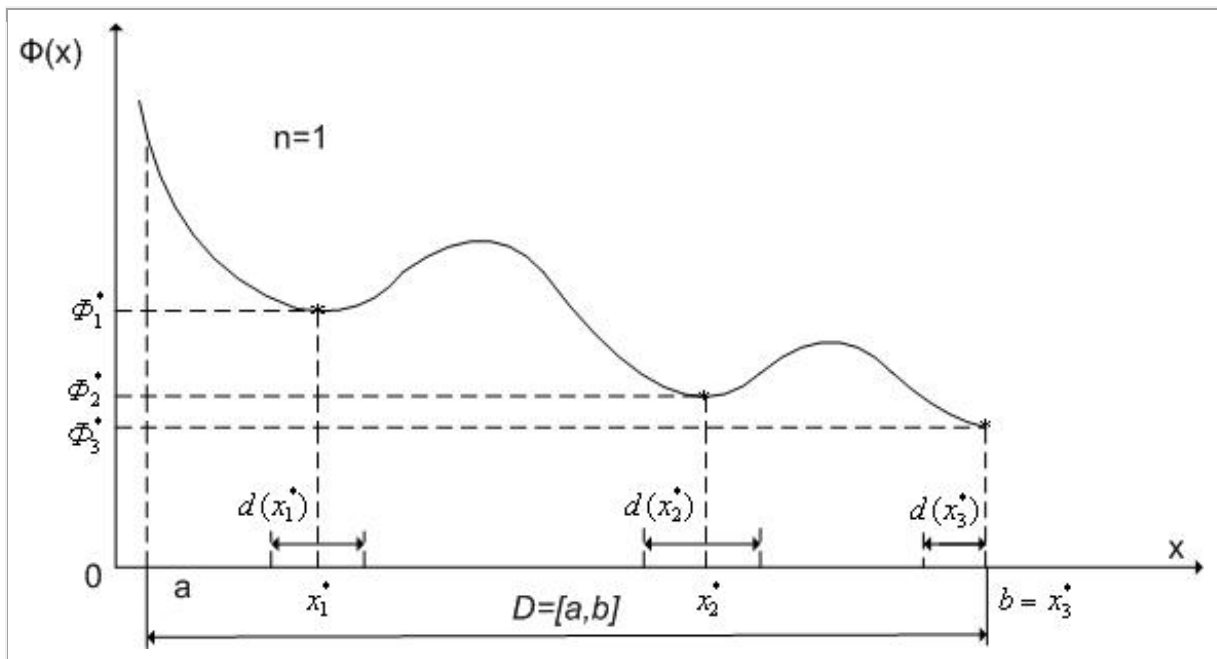


Рис. 1.1.4. К определению локального и глобального минимумов функции

1.2 Классификация критериев оптимальности

Критерий оптимальности $\Phi(X)$, где $x \in [a, b]$ скаляр, называется **уни-модальным критерием оптимальности**, если в области определения $[a, b]$ функции $\Phi(X)$ существует точка $x^* \in [a, b]$ такая, что на полуинтервале $[a, x^*)$ функция $\Phi(X)$ убывает, а на полуинтервале $(x^*, b]$ – возрастает. Заметим, что определение одномерного унимодального критерия оптимальности не требует непрерывности функции $\Phi(X)$.

Например, на рис.1.2.1 одномерная функция $\Phi(X)$ на интервале $[a, b]$ является унимодальной, хотя и имеет в точках x_1, x_2 разрывы. Заметим также, что точка x^* может быть как внутренней точкой отрезка $[a, b]$, так и совпадать с одним из его концов.

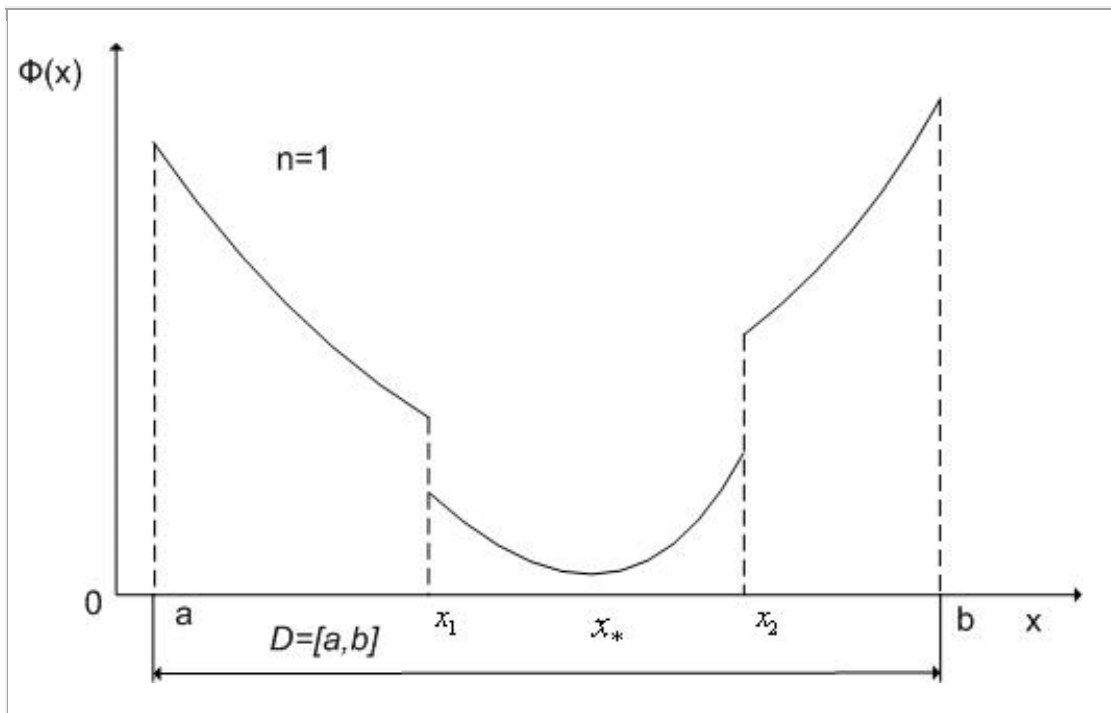


Рис. 1.2.1. К определению унимодального критерия оптимальности: x_1, x_2 - точки разрыва критерия оптимальности $\Phi(X)$

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

$$\Phi(\lambda x_1 + (1-\lambda)x_2) \leq \lambda \Phi(x_1) + (1-\lambda)\Phi(x_2), \quad (1.2.1)$$

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

Приведенное определение имеет простой геометрический смысл: если критерий оптимальности $\Phi(X)$ выпукл на интервале $[a,b]$, то все точки любой дуги его графика лежат под соответствующей хордой (см. рис. 1.2.2).

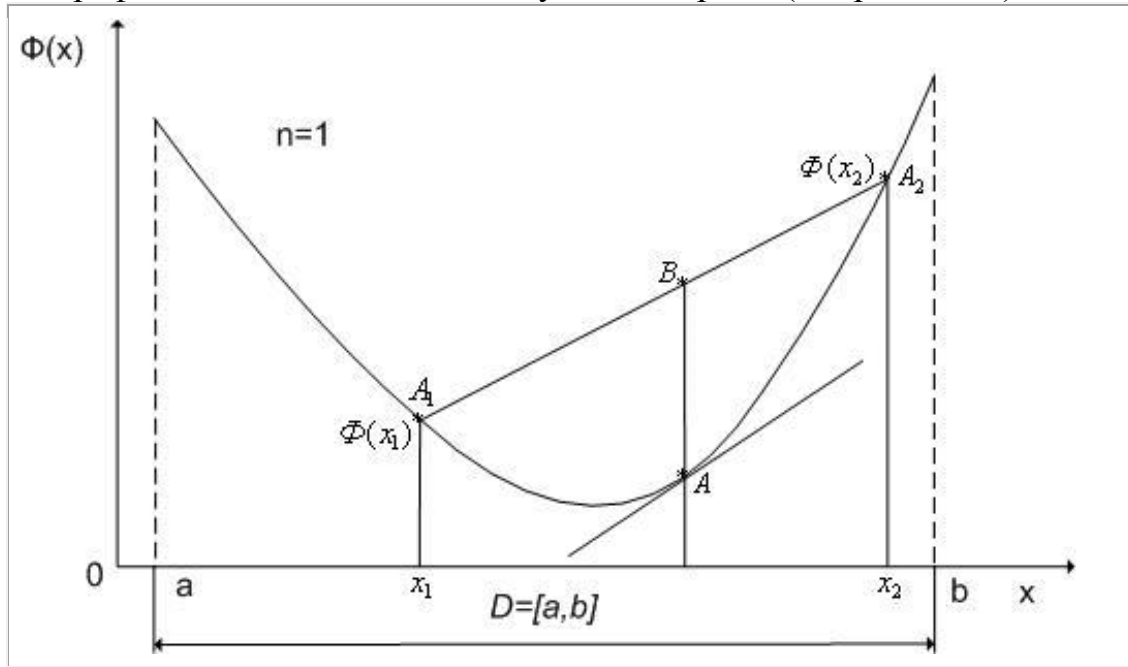


Рис. 1.2.2. К определению выпуклого одномерного критерия оптимальности

Заметим, что определение выпуклого критерия оптимальности не требует его унимодальности. Так что, например, выпуклым является критерий оптимальности, график которого изображен на рис. 1.2.3.

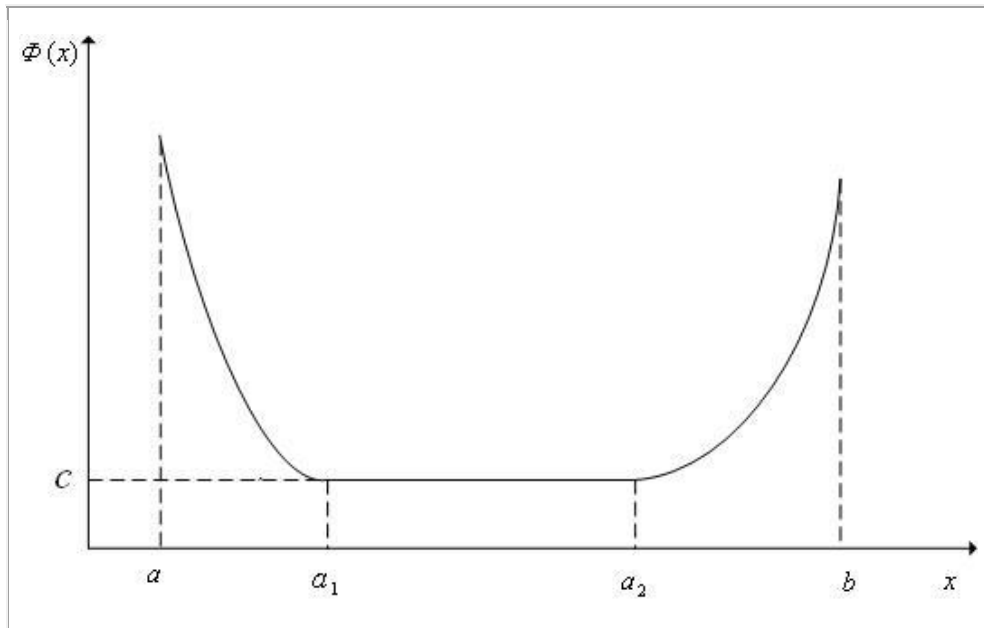


Рис. 1.2.3. Пример выпуклого критерия оптимальности: на интервале $[a_1, a_2]$ значения критерия оптимальности постоянны и равны C

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

$$\Phi(\lambda x_1 + (1-\lambda)x_2) < \lambda\Phi(x_1) + (1-\lambda)\Phi(x_2), \quad (1.2.2)$$

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

Замечание 1.2.1. Строго вогнутый критерий является унимодальным критерием.

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

Непрерывный критерий оптимальности $\Phi(X)$, где $X \in D$ и множество D является выпуклым множеством, называется **выпуклым критерием оптимальности** (выпуклым вниз критерием оптимальности), если для любых $X_1, X_2 \in D$, $X_1 \neq X_2$ и любого $\lambda \in [0, 1]$ выполняется неравенство:

$$\Phi(\lambda X_1 + (1-\lambda)X_2) \leq \lambda\Phi(X_1) + (1-\lambda)\Phi(X_2). \quad (1.2.3)$$

Аналогично, критерий оптимальности $\Phi(X)$, где $X \in D$ и множество D является выпуклым множеством, называется **строго выпуклым критерием оптимальности** (строго выпуклым вниз критерием оптимальности), если для любых $X_1, X_2 \in D$, $X_1 \neq X_2$ и $\lambda \in [0, 1]$ выполняется неравенство:

$$\Phi(\lambda X_1 + (1-\lambda)X_2) < \lambda\Phi(X_1) + (1-\lambda)\Phi(X_2). \quad (1.2.4)$$

Замечание 1.2.2. Выпуклая функция может иметь более одной точки локального минимума, а строго выпуклая функция – только одну точку.

Например, рассмотрим выпуклую квадратичную функцию $\Phi(X) = \Phi(x_1, x_2) = (x_1 + x_2)^2$. Легко видеть, что эта функция достигает в точке $(0, 0)$ минимума, равного нулю. Но это же значение функция принимает во всех точках вида $(x_1, x_2 = -x_1)$.

Критерий оптимальности $\Phi(X)$, имеющий в области определения несколько локальных минимумов, называется **многоэкстремальным критерием оптимальности** или **мультимодальным критерием оптимальности**.

Если размерность вектора варьируемых параметров X больше единицы ($n > 1$), то критерий оптимальности $\Phi(X)$ может быть в своей области допустимых значений D **"овражным"** критерием оптимальности.

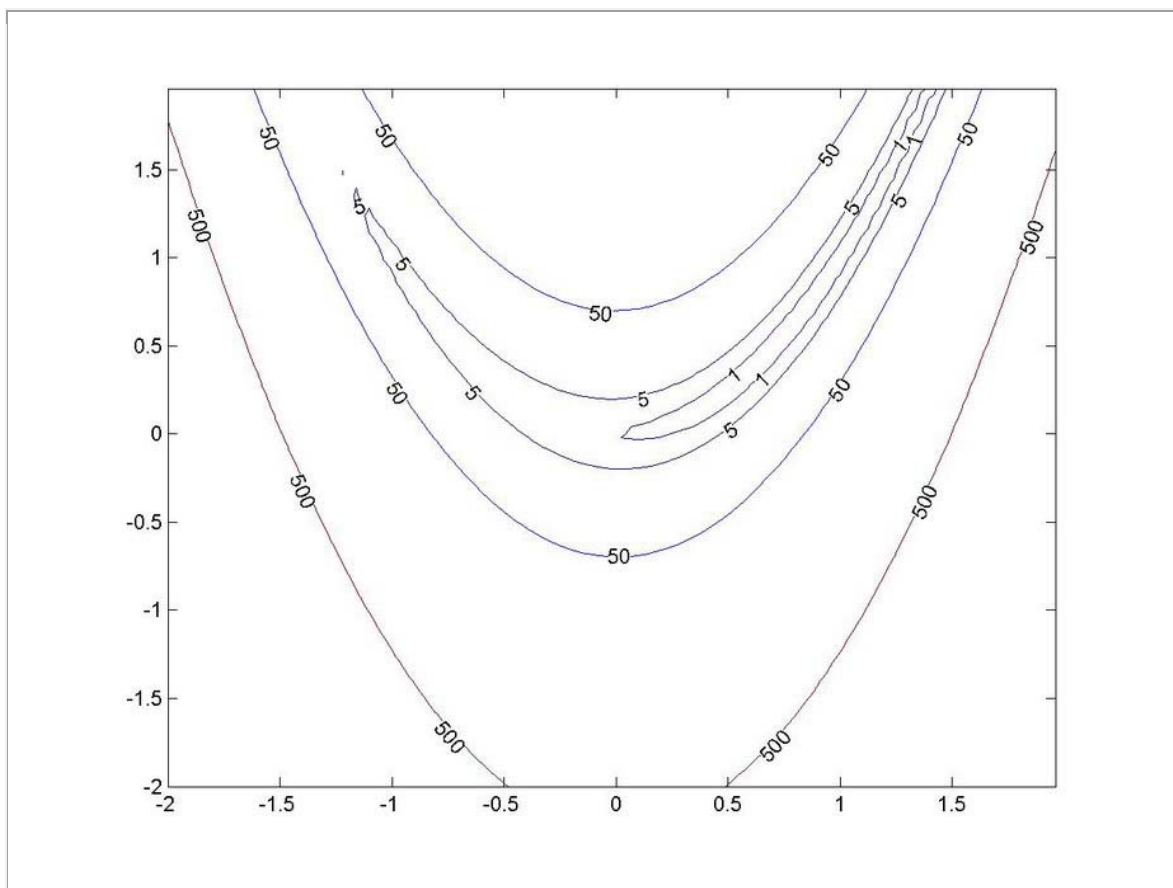


Рис. 1.2.4. Линии уровня функции Розенброка. Функция медленно изменяется вдоль дна V-образного оврага и быстро – перпендикулярно этому дну

Критерий оптимальности называется **овражным** в своей области допустимых значений, если в этой области имеют место слабые изменения частных производных функции $\Phi(X)$ по одним направлениям и значительные изменения этих производных по другим направлениям.

Рассмотрим (см. рис. 1.2.4) функцию Розенброка $\Phi(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ ($n = 2$). Легко видеть, что минимум этой функции достигается в точке $(0, 0)$ и равен нулю. Функция медленно изменяется вдоль дна V-образного оврага и быстро – перпендикулярно этому дну.

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

$$\Phi(X) = \sum_{i=1}^n \phi_i(x_i). \quad (1.2.5)$$

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

$$\Phi(X) = \sum_{i=1}^k c_i p_i(X), \quad (1.2.6)$$

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

$$p_i(X) = \prod_{j=1}^n x_j^{a_{ij}}, \quad (1.2.7)$$

a_{ij} – любые действительные числа.

Одномерный выпуклый критерий оптимальности

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

Теорема 1.2.1. Пусть критерий оптимальности $\Phi(X)$ определен и непрерывен на интервале $[a, b]$ и имеет в нем конечную первую производную $\Phi'(X)$. Для того, чтобы функция $\Phi(X)$ была выпуклой (строго выпуклой) на $[a, b]$, необходимо и достаточно, чтобы ее производная $\Phi'(X)$ не убывала (не возрастала) на этом интервале.

Теорема 1.2.2. Пусть критерий оптимальности $\Phi(X)$ определен и непрерывен вместе со своей первой производной $\Phi'(X)$ на интервале $[a, b]$ и имеет в нем конечную вторую производную $\Phi''(X)$. Для выпуклости функции $\Phi(X)$ на интервале $[a, b]$ необходимо и достаточно, чтобы **внутри** этого интервала имело место неравенство: $\Phi''(X) > 0$.

Замечание 1.2.3. Условие $\Phi''(X) > 0$ не является необходимым для строгой выпуклости функции $\Phi(X)$.

Так функция $\Phi(X) = x^4$ является строго выпуклой на всей числовой оси, хотя в начале координат ее вторая производная равна нулю: $\Phi''(0) = 0$.

Укажем еще одну очевидную, но важную геометрическую характеристику выпуклого критерия оптимальности. Если критерий оптимальности $\Phi(X)$ определен на интервале $[a, b]$, имеет на этом интервале конечную первую производную $\Phi'(X)$ и выпукл, то график функции $\Phi(X)$ всеми своими точками лежит над или на любой своей касательной (см. рис.1.2.2).

Многомерный критерий оптимальности

Суть понятия вогнутого (строго вогнутого) критерия оптимальности в многомерном случае удобно определить с помощью сечения. Сечением критерия оптимальности $\Phi(X)$, $X \in D$, где D – выпуклое множество, называется одномерная функция:

$$\varphi(\lambda) = \lambda\Phi(X_1) + (1-\lambda)\Phi(X_2), \quad (1.2.8)$$

где $\lambda \in [0,1]$ – вещественный скаляр, а X_1, X_2 – любые точки множества D .

Теорема 1.2.3. Для того чтобы критерий оптимальности $\Phi(X)$, определенный на выпуклом множестве D , был выпуклым критерием (строго выпуклым критерием), необходимо и достаточно, чтобы любое сечение этого критерия было выпуклой функцией (строго выпуклой функцией).

Теорема 1.2.4. Если критерий оптимальности $\Phi(X)$, определенный на выпуклом множестве $D \subset R^n$, является дважды дифференцируемым на этом множестве, то необходимым и достаточным условием его выпуклости является неотрицательная определенность на этом множестве матрицы вторых производных функции $\Phi(X)$ – **матрицы Гессе** (гессиана):

$$H(X) = \frac{\partial^2 \Phi(X)}{\partial x_i \cdot \partial x_j}, \quad i, j \in [1, \dots, n]. \quad (1.2.9)$$

Замечание 1.2.4. Симметричная ($n \times n$) матрица A называется неотрицательно определенной, если квадратичная форма $X^T A X = (AX, X)$ для любых векторов $X \in R^n$ принимает неотрицательные значения. Если матрица A неотрицательно определена, то ее собственные числа являются неотрицательными. Для проверки неотрицательной определенности матрицы удобно использовать **критерий Сильвестра**: если все главные миноры матрицы A неотрицательны, то матрица A неотрицательно определена.

По аналогии с одномерным случаем, положительная определенность матрицы Гессе не является достаточным условием строгой выпуклости функции $\Phi(X)$. Так функция $\Phi(X) = x_1^4 + x_2^4$ строго выпукла в пространстве R^2 , но ее матрица Гессе в точке $(0,0)$ является нулевой: $H(0,0) = 0$.

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

$$\nabla \Phi(X) = \frac{\partial \Phi(X)}{\partial x_i}, \quad i \in [1, \dots, n]. \quad (1.2.10)$$

1.3 Классификация задач оптимизации

Рассмотрим задачу оптимизации:

$$\min_{X \in D} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (1.3.1)$$

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

$$D = \{X / g(X) \geq 0, h(X) = 0\} = \{X / g_i(X) \geq 0, h_j(X) = 0, i \in [1, \dots, m], j \in [1, \dots, l]\} \quad (1.3.2)$$

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

Классификация по виду критерия оптимальности и ограничивающих функций

Если критерий оптимальности $\Phi(X)$ – линейная функция, а множество D – выпуклый многогранник, то задача (1.3.1), (1.3.2) называется **задачей линейного программирования**.

Если критерий оптимальности $\Phi(X)$ – есть отношение двух линейных функций, а множество D – выпуклый многогранник, задача (1.3.1), (1.3.2) называется **задачей дробно-линейного программирования**.

Пусть область D определяется только ограничениями типа неравенств:

$$D = \{X / g(X) \geq 0\} = \{X / g_i(X) \geq 0, i \in [1, \dots, m]\} \quad (1.3.3)$$

Тогда если функция $\Phi(X)$ и функции $g_i(X), i \in [1, \dots, m]$ являются сепарабельными, то задача (1.3.1), (1.3.3) называется **задачей сепарабельного программирования**.

Тогда если функция $\Phi(X)$ и ограничивающие функции $g_i(X), i \in [1, \dots, m]$ являются позиномами, то задача (1.3.1), (1.3.3) называется **задачей геометрического программирования**.

Если $\Phi(X)$ – квадратичная функция, т.е. $\Phi(X) = X^T G X + C^T X$, а множество D есть выпуклое множество, то задача (1.3.1), (1.3.2) называется **задачей квадратичного программирования**. Здесь G – $(n \times n)$ симметричная матрица, C – $(n \times 1)$ вектор.

Если множество является конечным множеством, то задача (1.3.1), (1.3.2) называется **задачей дискретного программирования**.

Если множество D является множеством целых чисел, то задача (1.3.1), (1.3.2) называется **задачей целочисленного программирования**.

Если функция $\Phi(X)$ является выпуклой, то задача (1.3.1), (1.3.2) называется **задачей выпуклого программирования**. Заметим, что определение выпуклой функции $\Phi(X)$ требует выпуклости области ее определения D .

В общем случае задача (1.3.1), (1.3.2) называется **задачей нелинейного программирования**.

Часто задачи выпуклого программирования также относят к задачам нелинейного программирования.

Классификация по наличию или отсутствию ограничений

Если ограничения на вектор X отсутствуют ($D = R^n$), то задача (1.3.1), (1.3.2.) называется **задачей безусловной оптимизации**.

Если имеются ограничения на вектор X ($D \subset R^n$), то задача (1.3.1), (1.3.2) называется **задачей условной оптимизации**.

Среди задач условной оптимизации выделяют следующие классы задач:

- задачи условной оптимизации с ограничениями типа неравенств, когда

$$D = \{X / g(X) \geq 0\} = \{X / g_i(X) \geq 0, i \in [1, \dots, m]\};$$

- задачи условной оптимизации с ограничениями типа равенств, когда

$$D = \{X / h(X) = 0\} = \{X / h_j(X) = 0, j \in [1, \dots, l]\};$$

- задачи условной оптимизации с ограничениями общего вида, когда имеются как ограничения типа неравенств, так и ограничения типа равенств, т.е. когда

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

Классификация по размерности вектора X

Если размерность вектора X равна 1 ($n=1$), то задача (1.3.1), (1.3.2) называется **однопараметрической задачей оптимизации** (одномерной задачей оптимизации).

Если размерность вектора X больше 1 ($n > 1$), то задача (1.3.1), (1.3.2) называется **многопараметрической задачей оптимизации** (многомерной задачей оптимизации).

Классификация по количеству точек минимума

Если функция $\Phi(X)$ имеет в области допустимых значений D один минимум, то задача (1.3.1), (1.3.2) называется **одноэкстремальной задачей оптимизации**.

Если функция $\Phi(X)$ имеет в области допустимых значений D более одного минимума, то задача (1.3.1), (1.3.2) называется **многоэкстремальной задачей оптимизации**.

Классификация по характеру искомого решения

Если отыскивается любой локальный минимум функции $\Phi(X)$, то задача (1.3.1), (1.3.2) называется **задачей локальной оптимизации**.

Если отыскивается любой локальный минимум функции $\Phi(X)$ и задача (1.3.1), (1.3.2) является задачей безусловной оптимизации, то эта задача называется **задачей безусловной локальной оптимизации**.

Аналогично, если отыскивается любой локальный минимум функции $\Phi(X)$ и задача (1.3.1), (1.3.2) является задачей условной оптимизации, то эта задача называется **задачей условной локальной оптимизации**.

Если отыскивается глобальный минимум функции $\Phi(X)$, то задача (1.3.1), (1.3.2) называется **задачей глобальной оптимизации**.

Если отыскивается глобальный минимум функции $\Phi(X)$ и задача (1.3.1), (1.3.2) является задачей безусловной оптимизации, то эта задача называется **задачей безусловной глобальной оптимизации**.

Аналогично, если отыскивается глобальный минимум функции $\Phi(X)$ и задача (1.3.1), (1.3.2) является задачей условной оптимизации, то эта задача называется **задачей условной глобальной оптимизации**.

1.4 Одномерная задача оптимизации

Рассмотрим задачу поиска минимума одномерной функции $\Phi(X)$, определенной на интервале $[a, b]$: $\min_{x \in [a, b]} \Phi(x) = \Phi(x^*)$.

Как известно из курса математического анализа, внутренние точки локального и глобального минимума функции $\Phi(X)$ являются **стационарными точками критерия оптимальности** $\Phi(X)$ (см. рис. 1.4.1) или, что то же самое, решениями уравнения:

$$\Phi'(X) = 0. \quad (1.4.1)$$

Но, решениями уравнения (1.4.1) являются не только точки минимума, но и точки максимума и точки перегиба функции $\Phi(X)$ (см. рис. 1.4.1). На рисунке отмечены локальные минимумы x_1^* , x_3^* , локальный максимум x_2^* и точка перегиба x_c^* функции $\Phi(X)$.

Следовательно, уравнение (1.4.1) является только необходимым условием минимума, но не является достаточным условием.

Если существует вторая производная функции $\Phi''(X)$, то для отыскания достаточных условий минимума $\Phi(X)$ можно привлечь эту производную. Из курса математического анализа известно, что если в точке x_0 значение первой производной функции $\Phi(X)$ равно нулю, а второй производной – положительно, то в этой точке функция $\Phi(X)$ имеет минимум (локальный или глобальный).

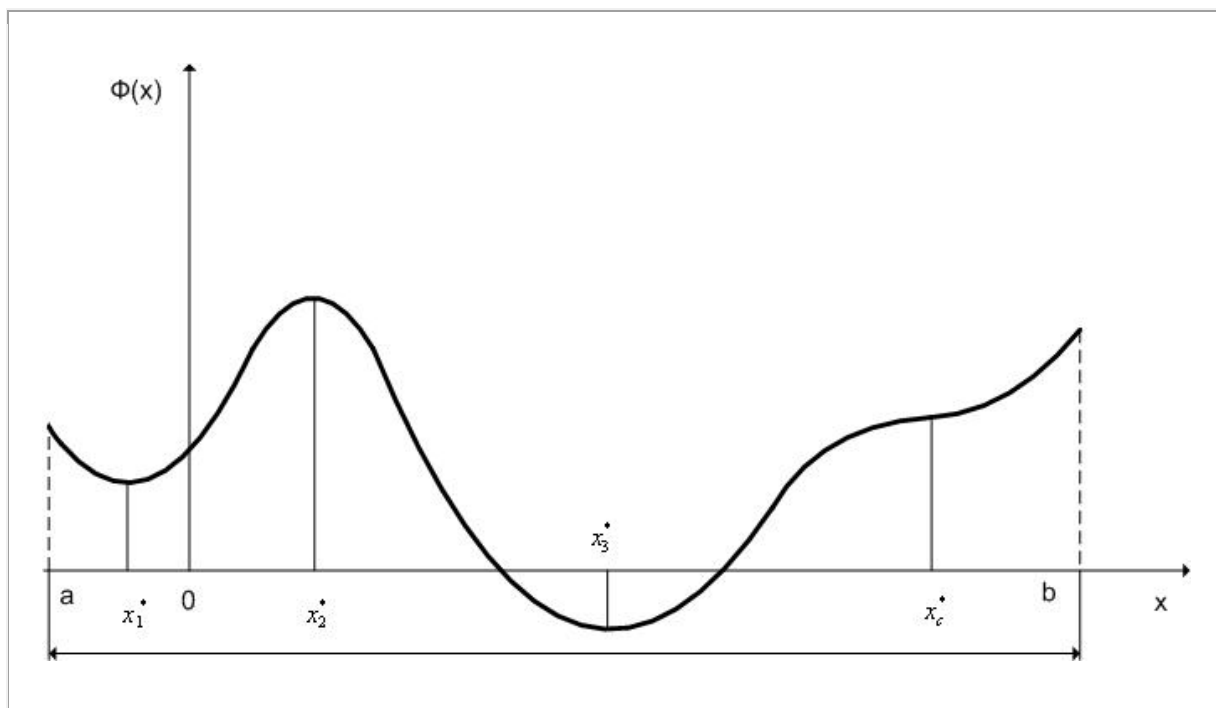


Рис. 1.4.1. Локальные минимумы (x_1^* , x_3^*), локальный максимум (x_2^*) и точка перегиба (x_c^*) функции $\Phi(X)$

Таким образом, имеем следующую теорему:

Теорема 1.4.1. Если функция $\Phi(X)$ определена и дважды непрерывно дифференцируема на интервале $[a, b]$, то необходимыми и достаточными условиями минимума этой функции в точке x_0 являются условия:

$$\Phi'(x_0) = 0, \Phi''(x_0) > 0. \quad (1.4.2)$$

Итак, точками, в которых функция $\Phi(X)$ принимает наименьшее на интервале $[a, b]$ значение, могут быть либо ее стационарные точки, лежащие внутри интервала $[a, b]$, либо ее точки недифференцируемости (**критические точки критерия оптимальности**), к которым следует отнести также концы интервала $[a, b]$.

Поэтому точку, в которой функция $\Phi(X)$ принимает наименьшее на интервале $[a, b]$ значение, нужно искать, сравнивая значения этой функции во всех стационарных и критических точках.

1.5 Многомерная задача безусловной оптимизации

Многие методы решения многомерной задачи нелинейного программирования основаны на сведении этой задачи к задаче безусловной оптимизации. Поэтому рассмотрим n -мерную задачу оптимизации без ограничений:

$$\min_{X \in R^n} \Phi(X) = \Phi(X^*) \quad (1.5.1)$$

По аналогии с одномерной задачей, для того, чтобы точка X_0 являлась минимумом функции $\Phi(X)$ необходимо выполнение условия стационарности функции $\Phi(X)$ в точке X_0 или, что тоже самое, необходимо, чтобы точка X_0 была стационарной точкой функции $\Phi(X)$:

$$\nabla \Phi(X_0) = 0. \quad (1.5.2)$$

Положим, что функция $\Phi(X)$ дважды непрерывно дифференцируема в окрестности точки X_0 . Для поиска достаточного условия достижения этой функцией в точке X_0 минимума, разложим $\Phi(X)$ в окрестности точки X_0 в ряд Тейлора:

$$\Phi(X_0 + \Delta X) - \Phi(X_0) = \Delta X^T \nabla \Phi(X_0) + \frac{1}{2} \Delta X^T H(X_0) \Delta X + \dots \quad (1.5.3)$$

Здесь n -мерный вектор-столбец достаточно малых величин $\Delta X = \{\Delta x_i, i \in [1, \dots, n]\}$, $H(X)$ – $(n \times n)$ – матрица Гессе.

По аналогии с одномерной задачей, для того, что в точке X_0 достигался минимум функции $\Phi(X)$, необходимо, чтобы разность $\Phi(X_0 + \Delta X) - \Phi(X_0)$ была положительной. Поскольку $\nabla \Phi(X_0) = 0$, то из (1.5.3) следует, что для выполнения этого условия необходимо, чтобы матрица Гессе $H(X)$ была положительно определена в точке X_0 .

Таким образом, справедлива

Теорема 1.5.1. Если функция $\Phi(X)$ дважды непрерывно дифференцируема в окрестности точки $X_0 \in R^n$, то необходимыми и достаточными условиями минимума этой функция в точке X_0 являются условия: $\nabla \Phi(X_0) = 0$; $H(X_0)$ – положительно определена.

Таким образом, теорема 1.5.1 определяет **необходимые и достаточные условия минимума в многомерной задаче безусловной оптимизации.**

Заметим, что условие $\nabla \Phi(X_0) = 0$ является только **необходимым условием минимума в многомерной задаче безусловной оптимизации.**

По аналогии с одномерной задачей точками, в которых функция $\Phi(X)$ достигает своего наименьшего значения, могут быть либо ее стационарные точки функции, либо критические точки функции (точки недифференцируемости).

Поэтому так же, как в одномерной задаче, точку, в которой функция $\Phi(X)$ принимает наименьшее значение нужно искать, сравнивая значения этой функции во всех стационарных и критических точках.

1.6 Задача выпуклого программирования

Рассмотрим n -мерную задачу выпуклого программирования $\min_{X \in D} \Phi(X) = \Phi(X^*) = \Phi^*$, $\Phi(X)$ – выпуклая функция, D – выпуклое не пустое ограниченное и замкнутое множество допустимых значений вектора варьируемых переменных. Напомним, что по определению выпуклая функция является непрерывной.

Во внутренних точках множества допустимых значений D функция $\Phi(X)$ достигает минимального значения в точках, которые являются ее либо стационарными точками функции, либо критическими точками функции. Однако функция может достигать своего наименьшего значения и в граничных точках области определения D .

Важные свойства задачи выпуклого программирования определяют две следующие теоремы.

Теорема 1.6.1. Если внутренняя точка X^* множества D является точкой локального минимума в задаче выпуклого программирования, то в этой точке функция $\Phi(X)$ достигает глобального минимума.

Доказательство. Положим, что в точке X^* функция $\Phi(X)$ не достигает наименьшего во множестве D значения. Тогда существует точка $Y \in D$, для которой $\Phi(Y) < \Phi(X^*)$. Рассмотрим сечение $\varphi(\lambda) = \Phi(\lambda Y + (1-\lambda)X^*)$, $\lambda \in [0,1]$.

Функция $\varphi(\lambda)$ достигает в точке наибольшее значение. Действительно, по-

скольку $\varphi(\lambda) = \Phi(\lambda Y + (1-\lambda)X^*) \leq$

$$\leq \lambda \Phi(Y) + (1-\lambda)\Phi(X^*) < \lambda \Phi(X^*) + (1-\lambda)\Phi(X^*) = \Phi(X^*) = \varphi(0).$$

Это значит, что существует окрестность $\delta D \in D$ точки X^* и некоторое

$\tilde{\lambda} \in [0,1]$ такие, что $X = \tilde{\lambda}Y + (1-\tilde{\lambda})X^* \in \delta D$. Но тогда

$\Phi(X) = \varphi(\tilde{\lambda}) < \varphi(0) = \Phi(X^*)$, что противоречит условию теоремы.

Из теоремы следует, что во всех точках локального минимума выпуклая функция имеет одинаковые значения.

Пример 1.6.1. Рассмотрим не строго выпуклую квадратичную функцию $\Phi(X) = \Phi(x_1, x_2) = (x_1 + x_2)^2$, определенную в области $D = \{X / -2 \leq x_1 \leq 2, -2 \leq x_2 \leq 2\}$ (см. рис. 1.6.1). Все локальные минимумы этой функции равны нулю и расположены на прямой $-x_1 + x_2 = 0$.

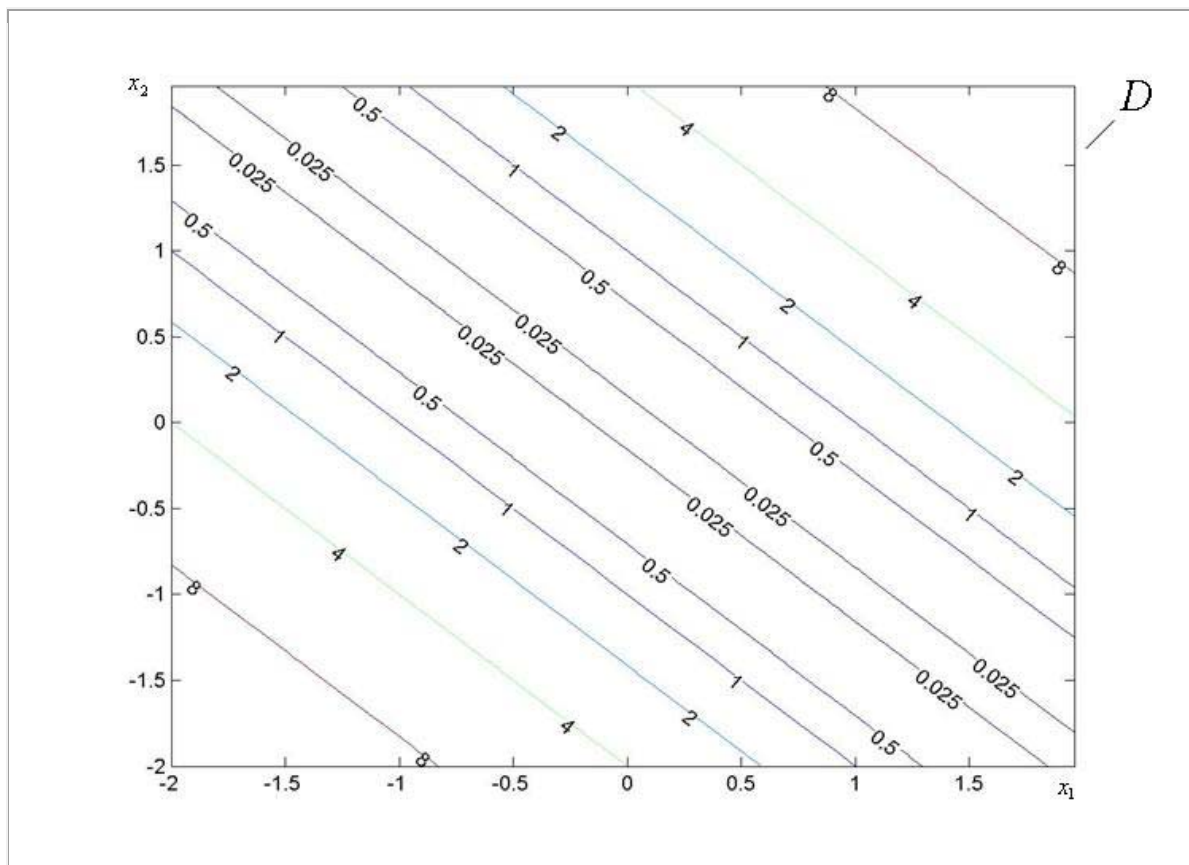


Рис. 1.6.1. К примеру 1.6.1

Теорема 1.6.2. Функция $\Phi(X)$, строго выпуклая функция на выпуклом множестве, имеет в этом множестве не более одной точки минимума (глобального).

Условие существования решения задачи выпуклого программирования дает следующая теорема.

Теорема 1.6.3. Пусть функция $\Phi(X)$ выпукла на выпуклом множестве $D \subset R^n$ и дифференцируема в точке $X^* \in D$. Тогда для того чтобы эта точка была точкой минимума функции $\Phi(X)$, необходимо и достаточно, чтобы для любой точки $X \in D$ выполнялось неравенство:

$$(\nabla \Phi(X^*), (X - X^*)) \geq 0 \quad (1.6.1)$$

Необходимость. Рассмотрим сечение $\varphi(\lambda) = \Phi(\lambda X + (1 - \lambda)X^*)$ функции $\Phi(X)$. Функция $\varphi(\lambda)$ определена на отрезке $[0, 1]$, имеет в точке $\lambda = 0$ ло-

кальный минимум и дифференцируема в этой точке. Следовательно $\varphi'(0) \geq 0$ (равенство нулю имеет место в том случае, когда точка X^* является внутренней точкой множества D). По правилу дифференцирования сложной функции $\varphi'(0) = \Phi'_\lambda \Phi(\lambda X + (1-\lambda)X^*)|_{\lambda=0} = (\nabla \Phi(X^*), (X - X^*)) \geq 0$

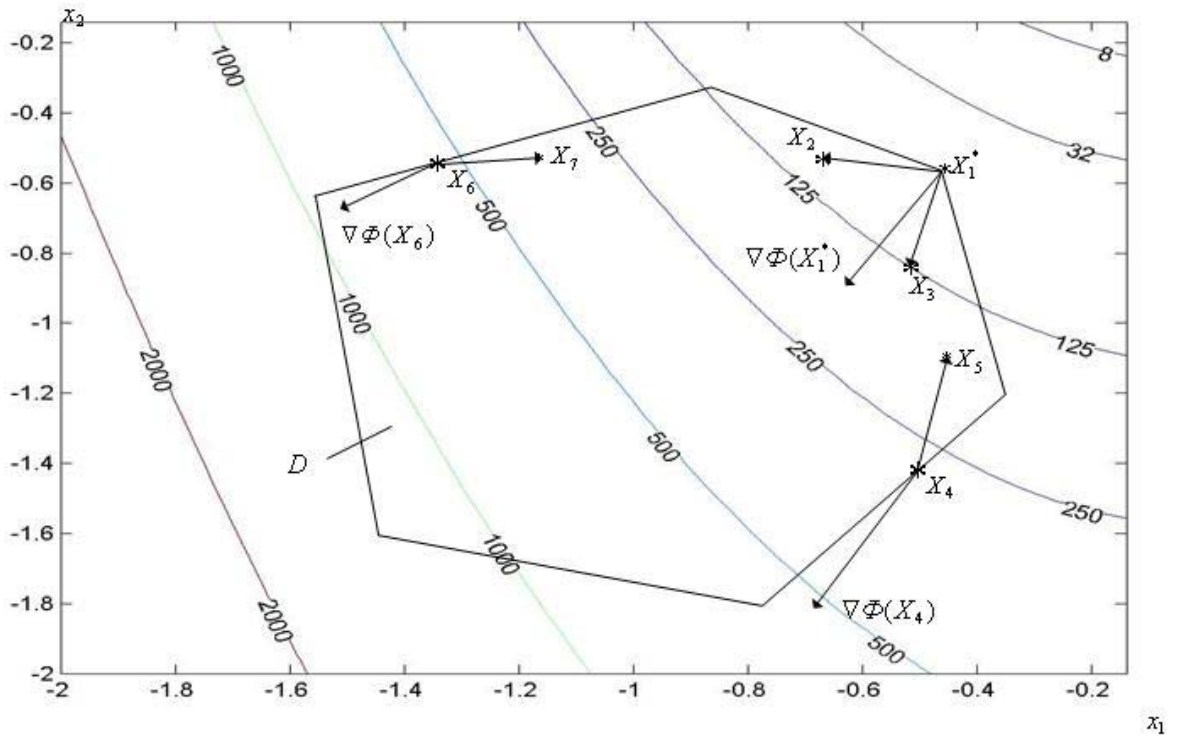


Рис. 1.6.2. К теореме 1.6.3

Достаточность. Пусть в точке $X^* \in D$ выполнено неравенство (1.6.1). Рассмотрим сечение $\varphi(\lambda) = \Phi(\lambda X + (1-\lambda)X^*)$ функции $\Phi(X)$, где X – произвольная точка из множества D . Поскольку $\Phi(X)$ выпукла во множестве D , то функция $\varphi(\lambda)$ также выпукла на отрезке $[0,1]$. Кроме того, из неравенства (1.6.1) следует, что $\varphi'(0) \geq 0$. Это означает, что $\varphi(\lambda)$ – неубывающая отрезке $[0,1]$ функция, т.е. $\varphi(0) \leq \varphi(1)$. Последнее неравенство означает, что $\Phi(X^*) \leq \Phi(X)$ и в точке X^* функция $\Phi(X)$ принимает наименьшее в области D значение.

Теорему 1.6.3 иллюстрирует рис. 1.6.2. Точка X_1^* на рис. 1.6.2 является точкой локального минимума, поскольку не существует такой точки $X \in D$, что скалярное произведение $(\nabla \Phi(X_1^*), (X - X_1^*))$ отрицательно. Точка X_4 , например, не является точкой локального минимума, так как существуют та-

кие точки $X \in D$, что скалярное произведение $(\nabla \Phi(X_4^*), (X - X_4^*))$ отрицательно.

Заметим, что если точка $X^* \in D$ является внутренней точкой множества D , то условие (1.6.1) эквивалентно условию $\nabla \Phi(X^*) = 0$. Таким образом, условие (1.6.1) можно рассматривать как обобщение необходимого условия минимума в многомерной задаче безусловной оптимизации.

1.7 Задача нелинейного программирования с ограничениями типа равенств

Рассмотрим n -мерную задачу нелинейного программирования:

$$\min_{X \in D} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (1.7.1)$$

где

$$D = \{X / h(X) = 0\} = \{X / h_j(X) = 0, j \in [1, \dots, l]\} \subset R^n \quad (1.7.2)$$

– не пустое, ограниченное замкнутое множество.

Нам понадобятся далее понятия **множителей Лагранжа** и **функции Лагранжа**. Функция Лагранжа для задачи (1.7.1) с ограничениями (1.7.2) определяется формулой:

$$L(X, \lambda) = \Phi(X) + \sum_{j=1}^l \lambda_j h_j(X) = \Phi(X) + \lambda^T h(X), \quad (1.7.3)$$

где $\lambda = (\lambda_j, j \in [1, \dots, l])$ – $(l \times 1)$ – вектор множителей Лагранжа.

Нам понадобится также понятие **условия регулярности ограничивающих функций**. Если точка $X^* \in D$, то условие линейной независимости векторов $\nabla h_j(X^*), j \in [1, \dots, l]$ называется условием регулярности задачи (1.7.1), (1.7.2) в точке X^* . Данное условие означает, в частности, что количество ограничивающих функций, проходящих через точку X^* , не может быть больше размерности вектора варьируемых параметров, т.е. должно быть выполнено неравенство $l \leq n$. Например, на рис. 1.7.1 в ситуации (а) количество ограничивающих функций, проходящих через точку X^* , превышает размерность вектора варьируемых параметров, в ситуации (б) в точке X^* градиенты $\nabla h_1(X), \nabla h_2(X)$ ограничивающих функций коллинеарны.

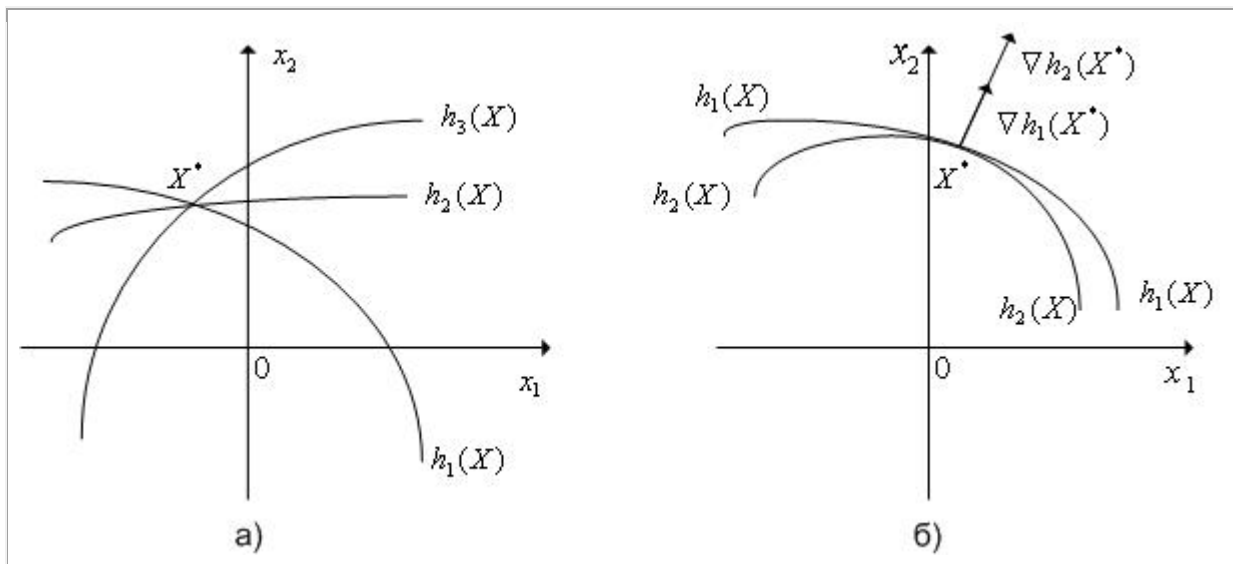


Рис. 1.7.1. Ситуации, в которых в двумерном случае ($n = 2$) не выполняется условие регулярности системы функций $h(X)$ в точке X^* .

Исключительно важное место в теории и практике решения задач нелинейного программирования с ограничениями типа равенств занимает следующая теорема (**правило Лагранжа для задачи оптимизации с ограничениями типа равенств**).

Теорема 1.7.1. Пусть функция $\Phi(X)$ и функции $\nabla h_j(X), j \in [1, \dots, l]$ имеют непрерывные частные производные в некоторой окрестности точки X^* и пусть эта точка является точкой локального минимума функции $\Phi(X)$ при условии $h(X^*) = 0$. Пусть, кроме того, выполняется условие регулярности системы функций $h(X)$ в точке X^* . Тогда существуют такие множители Лагранжа $\lambda_j, j \in [1, \dots, l]$, не все из которых равны нулю одновременно, что для функции Лагранжа $L(X, \lambda)$ точка X^* является стационарной точкой функции, т.е.

$$\nabla_x L(X^*, \lambda) = \nabla \Phi(X^*) + \sum_{j=1}^l \lambda_j \nabla h_j(X^*) = 0. \quad (1.7.4)$$

Отметим, что теорема 1.7.1 не требует знакоопределенности (т.е. положительности или отрицательности) множителей Лагранжа $\lambda_j, j \in [1, \dots, l]$. Теорема требуется лишь того, чтобы не все из этих множителей равнялись нулю одновременно.

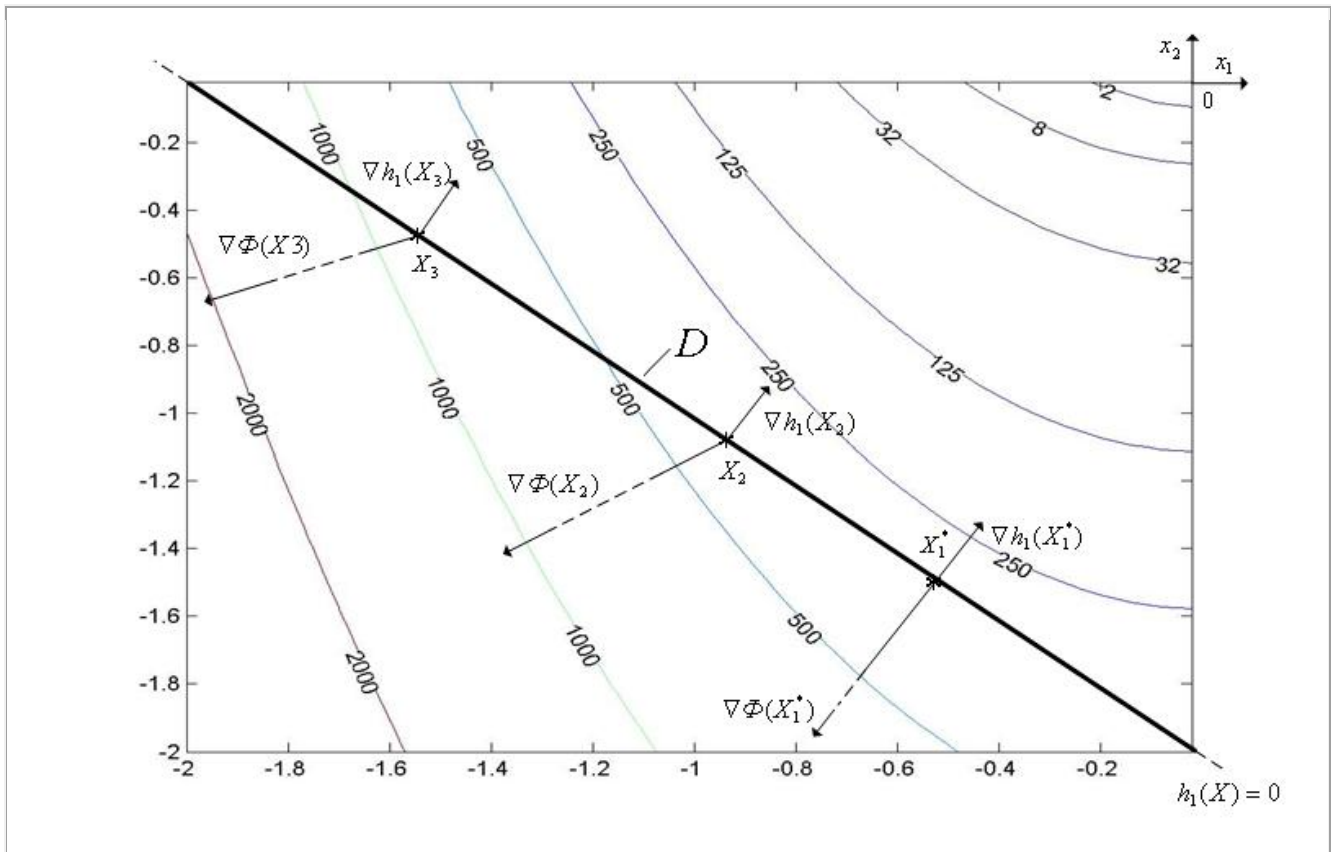


Рис. 1.7.2. К примеру 1.7.1

Пример 1.7.1. Рассмотрим в качестве минимизируемой функции $\Phi(X)$ функцию Розенброка: $\Phi(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, ($n = 2$). Положим, что имеется только одно ограничение типа равенств, которое задается с помощью функции $h_1(X) = x_1 + x_2 + 0.2 = 0$. Легко видеть, что градиенты функций $\Phi(X)$, $h_1(X)$ равны, соответственно:

$$\nabla \Phi(X) = \left(-400(x_2 - x_1^2)x_1 - 2(1 - x_1) \quad 200(x_2 - x_1^2) \right)^T, \quad \nabla h_1(X) = (1 \quad 1)^T.$$

Пример 1.7.1 иллюстрирует рис. 1.7.2. В точках X_2, X_3 векторы градиента функций $\Phi(X)$, $h_1(X)$ не коллинеарны. Поэтому для этих точек не существует не равный нулю множитель Лагранжа λ_1 , при котором функция Лагранжа равна нулю: $\nabla \Phi(X) + \lambda_1 \nabla h_1(X) = 0$. И поэтому точки X_2, X_3 не могут быть точками локального минимума для рассматриваемой задачи. Наоборот, в точке X_1^* векторы градиента функций $\Phi(X)$, $h_1(X)$ коллинеарны и поэтому существует не равный нулю множитель λ_1 при котором справедливо равенство $\nabla \Phi(X^*) + \lambda_1 \nabla h_1(X^*) = 0$. Отметим, что, например, в точке $(-1 \quad -1)^T$, градиент функции Розенброка равен $(-402 \quad -400)^T$.

Теорема 1.7.1. означает, что в ее условиях вместо задачи условной оптимизации (1.7.1), (1.7.2) можно решать задачу безусловной оптимизации

$$\min_{X \in R^n} \left(\Phi(X) + \sum_{j=1}^l \lambda_j h_j(X) \right).$$

Необходимым условием существования локального минимума этой задачи в некоторой точке $X^* \in R^n$ является условие $\nabla_x \left(\Phi(X) + \sum_{j=1}^l \lambda_j h_j(X) \right) = 0$.

Широко известна другая форма теоремы 1.7.1, которую мы сформулируем в виде следствия этой теоремы.

Следствие 1.7.1. В условиях теоремы 1.7.1 существуют такие множители Лагранжа $\lambda_j, j \in [1, \dots, l]$, не все из которых равны нулю одновременно, что имеют место следующие равенства:

$$\nabla_x L(X^*, \lambda) = 0, \quad (1.7.5)$$

$$\nabla_\lambda L(X^*, \lambda) = h(X^*) = 0. \quad (1.7.6)$$

Здесь равенство (1.7.5) повторяет равенство (1.7.4), а справедливость равенства (1.7.6) следует из того факта, что по условиям теоремы точка X^* удовлетворяет всем ограничениям, т.е. $h_j(X^*) = 0, j \in [1, \dots, l]$.

Заметим, что из (1.7.6) следует справедливость еще одного полезного равенства $\lambda^T \nabla_\lambda L(X^*, \lambda) = \sum_{j=1}^l \lambda_j h_j(X^*) = \lambda^T h(X^*) = 0$.

1.8 Теорема Куна-Таккера для задачи нелинейного программирования с ограничениями типа неравенств

Рассмотрим задачу нелинейного программирования:

$$\min_{X \in D} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (1.8.1)$$

где $\Phi(X)$ – произвольная функция,

$$D = \{X / g(X) \geq 0\} = \{X / g_i(X) \geq 0, i \in [1, \dots, m]\} \subset R^n \text{ – не пустое, ограниченное замкнутое множество.}$$

Нам понадобятся далее понятия множителей Лагранжа и функции Лагранжа для задачи нелинейного программирования с ограничениями типа неравенств. Функция Лагранжа для задачи (1.8.1) с ограничениями (1.8.2) определяется формулой:

$$L(X, \lambda) = \Phi(X) + \sum_{i=1}^m \lambda_i g_i(X) = \Phi(X) + \lambda^T g(X), \quad (1.8.2)$$

где $\lambda = (\lambda_i, i \in [1, \dots, m]) - (m \times 1)$ – вектор множителей Лагранжа.

Нам понадобятся также понятия активных и неактивных ограничений. В точке локального минимума задачи (1.8.1), (1.8.2) каждое из ограничений $g_i(X^*) \geq 0, i \in [1, \dots, m]$ выполняется либо в виде равенства $g_i(X^*) = 0$, либо в виде неравенства $g_i(X^*) > 0$. Ограничения первого вида называются **активными ограничениями**. Остальные ограничения называются **неактивными ограничениями**.

Кроме того, нам понадобится также понятие условия регулярности для задачи нелинейного программирования с ограничениями типа неравенств. Если точка $X^* \in D$ и ограничения $g_{i_j}(X^*) \geq 0, i_j \in [1, \dots, s], s \leq m$ активны, то условие линейной независимости векторов называется условием регулярности ограничивающих функций $\nabla g_{i_j}(X^*) \geq 0, i_j \in [1, \dots, s]$ в точке X^* . Это условие означает, что, например, при $n = 2$ количество ограничивающих функций, проходящих через точку X^* , не должно превышать 2 и в точке X^* векторы $\nabla g_1(X), \nabla g_2(X)$ не должны быть коллинеарны.

Например, на рис. 1.8.1 в ситуации (а) количество ограничивающих функций, проходящих через точку X^* , превышает размерность вектора варьируемых параметров, в ситуации (б) в точке X^* градиенты $\nabla g_1(X), \nabla g_2(X)$ ограничивающих функций коллинеарны.

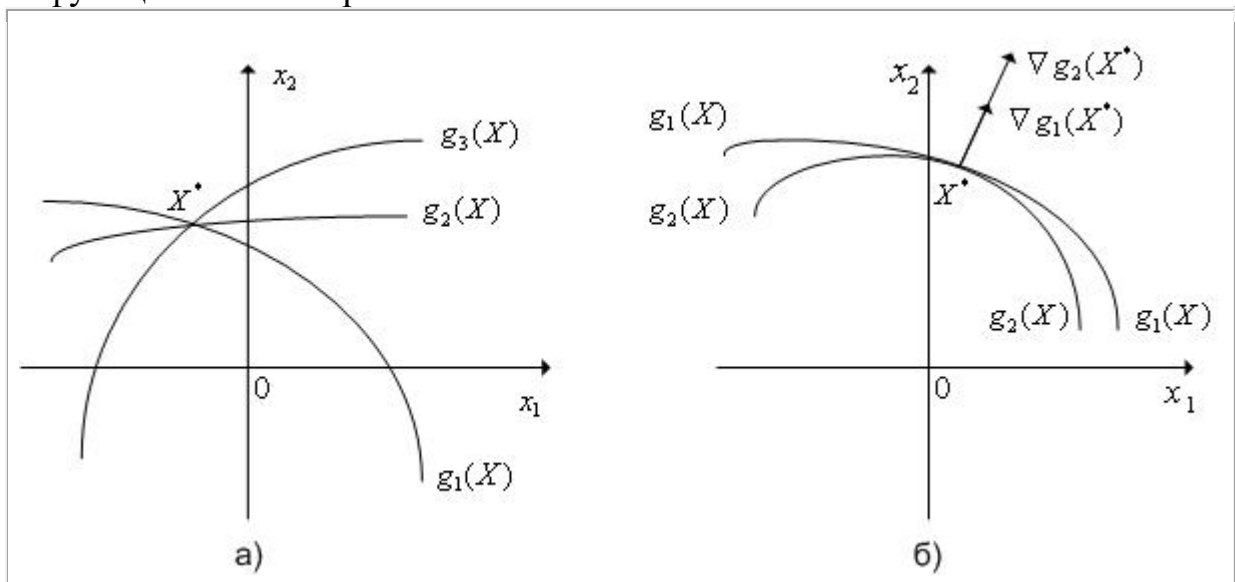


Рис. 1.8.1. Ситуации, в которых не выполняется условие регулярности двумерной задачи.

Исключительно большое значение в теории и практике решения задач нелинейного программирования имеет следующая теорема (**теорема Куна-**

Таккера для задачи условной оптимизации с ограничениями типа неравенств).

Теорема 1.8.1. (Куна-Таккера). Пусть функция $\Phi(X)$ и функции $g_i(X) \geq 0, i \in [1, \dots, m]$ имеют непрерывные частные производные в некоторой окрестности точки X^* и пусть эта точка является точкой локального минимума функции $\Phi(X)$ при ограничениях $g_i(X^*) \geq 0$, удовлетворяющих в точке X^* условию регулярности ограничивающих функций. Тогда существуют такие неотрицательные множители Лагранжа $\lambda_i, i \in [1, \dots, m]$, что для функции Лагранжа $L(X, \lambda)$ точка X^* является стационарной точкой функции, т.е.

$$\nabla_x L(X^*, \lambda) = \nabla \Phi(X^*) + \sum_{i=1}^m \lambda_i \nabla g_i(X^*) = 0. \quad (1.8.3)$$

Заметим, что в отличие от правила множителей Лагранжа, теорема 1.8.1 требует знакоопределенности множителей Лагранжа $\lambda_i, i \in [1, \dots, m]$. Отметим также, что теорема не запрещает того, чтобы все множители Лагранжа $\lambda_i, i \in [1, \dots, m]$ были равны нулю.

Поясним смысл теоремы на примере.

Пример 1.8.1. Рассмотрим двумерную ($n=2$) задачу нелинейного программирования (1.8.1), (1.8.2), в которой область допустимых значений D задается тремя ограничивающими функциями, т.е. $D = \{X / g(X) \geq 0\} = \{X / g_i(X) \geq 0, i \in [1, \dots, 3]\} \subset R^2$. Положим, что множество D имеет вид, представленный на рис. 1.8.2.

Для всех граничных точек области D , очевидно, выполняются условия регулярности ограничивающих функций.

Если точка X^* находится внутри множества D (т.е. является стационарной точкой функции $\Phi(X)$), то теорема будет справедлива, если положить все множители Лагранжа $\lambda_i, i \in [1, \dots, m]$ равными нулю.

Пусть теперь точка X^* находится на одной из дуг, например, на дуге AB , т.е. пусть ограничение $g_1(X) \geq 0$ является активным ограничением, а остальные ограничения – неактивными ограничениями. Тогда в этой точке $g_1(X^*) = 0$ и справедливость теоремы вытекает из правила множителей Лагранжа для задачи с ограничениями типа равенств, если положить $\lambda_2 = \lambda_3 = 0$.

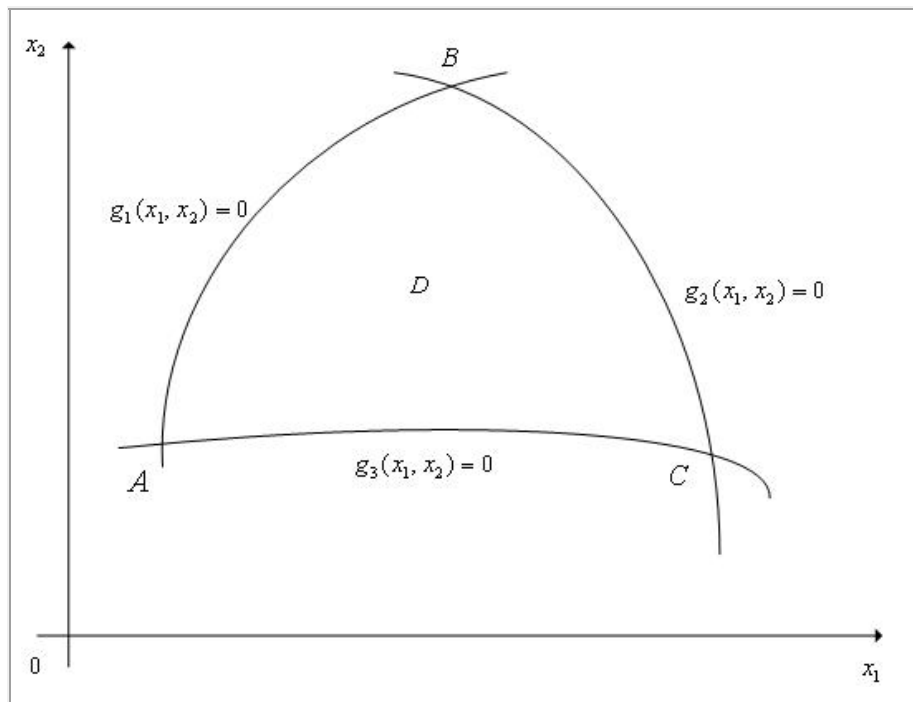


Рис. 1.8.2. К примеру 1.8.1.

Пусть, наконец, точка X^* находится в одной из угловых точек множества D , например, в точке B , т.е. пусть ограничения $g_1(X) \geq 0, g_2(X) \geq 0$ – являются активными ограничениями, а ограничение $g_3(X) \geq 0$ – неактивным ограничением. Тогда можно положить $\lambda_3 = 0$ и справедливость теоремы вытекает из правила множителей Лагранжа для задачи с ограничениями типа равенств.

Теорема 1.8.1 означает, что в ее условиях вместо задачи условной оптимизации (1.8.1), (1.8.2) можно решать задачу безусловной оптимизации

$$\min_{X \in R^n} \left(\Phi(X) + \sum_{i=1}^m \lambda_i g_i(X) \right).$$

Необходимым условием существования локального минимума этой задачи в некоторой точке $X^* \in R^n$ является условие $\nabla_x \left(\Phi(X^*) + \sum_{i=1}^m \lambda_i g_i(X^*) \right) = 0$.

Широко известна другая форма теоремы 1.8.1, которую мы сформулируем в виде следствия этой теоремы.

Следствие 1.8.1. В условиях теоремы 1.8.1 существуют такие неотрицательные множители Лагранжа $\lambda_i, i \in [1, \dots, m]$, что имеют место следующие равенства:

$$\nabla_x L(X^*, \lambda) = 0, \tag{1.8.4}$$

$$\nabla_\lambda L(X^*, \lambda) = g(X^*) \geq 0. \tag{1.8.5}$$

Здесь равенство (1.8.5) повторяет равенство (1.8.4), а справедливость равенства (1.8.6) следует из того факта, что по условиям теоремы точка X^* удовлетворяет всем ограничениям, т.е. $g_i(X^*) \geq 0, i \in [1, \dots, m]$.

Заметим, что из (1.8.6) следует справедливость еще одного полезного равенства $\lambda^T \nabla_{\lambda} L(X^*, \lambda) = \sum_{i=1}^m \lambda_i g_i(X^*) = \lambda^T g(X^*) \geq 0$.

1.9 Теорема Куна-Таккера для общей задачи нелинейного программирования

Рассмотрим общую задачу нелинейного программирования:

$$\min_{X \in D} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (1.9.1)$$

где $\Phi(X)$ – произвольная функция,

$$\begin{aligned} D = \{X / g(X) \geq 0, h(X) = 0\} = \\ = \{X / g_i(X) \geq 0, h_j(X) = 0, i \in [1, \dots, m], j \in [1, \dots, l]\} \subset R^n \end{aligned} \quad (1.9.2)$$

не пустое ограниченное замкнутое множество.

Нам понадобятся далее понятия множителей Лагранжа и функции Лагранжа для общей задачи нелинейного программирования.

Функция Лагранжа для задачи (1.9.1) с ограничениями (1.9.2) определяется формулой

$$L(X, \lambda, \mu) = \Phi(X) + \sum_{i=1}^m \lambda_i g_i(X) + \sum_{j=1}^l \mu_j h_j(X) = \Phi(X) + \lambda^T g(X) + \mu^T h(X),$$

где $\lambda = (\lambda_i, i \in [1, \dots, m])$, $\mu = (\mu_j, j \in [1, \dots, l])$ – $(m \times 1)$ и $(l \times 1)$ – векторы множителей Лагранжа, соответственно.

Нам понадобится также понятие условий регулярности для общей задачи нелинейного программирования. Если точка $X^* \in D$ и ограничения $g_{i_j}(X^*) \geq 0, i_j \in [1, \dots, s], s \leq m$ являются активными ограничениями, то условие линейной независимости векторов $\nabla g_{i_j}(X^*) \geq 0, i_j \in [1, \dots, s]$, а также условие линейной независимости векторов $\nabla h_j(X^*), j \in [1, \dots, l]$ называются условиями регулярности ограничивающих функций в точке X^* .

Смысл условий регулярности раскрыт в предыдущих параграфах.

Теорема 1.9.1. (Куна-Таккера). Пусть функции $\Phi(X)$, $g_j(X) \geq 0, h_j(X) = 0, i \in [1, \dots, m], j \in [1, \dots, l]$, имеют непрерывные частные производные в некоторой окрестности точки $X^* \in D$ и пусть эта точка является точ-

кой локального минимума функции $\Phi(X)$. Пусть, кроме того, выполняются условия регулярности ограничивающих функций $g(X)$, $h(X)$, в точке X^* . Тогда существуют такие множители Лагранжа $\lambda_i, i \in [1, \dots, m]$, $\mu_j, j \in [1, \dots, l]$, не все из которых равны нулю одновременно, что для функции Лагранжа $L(X, \lambda, \mu)$ точка X^* является стационарной точкой функции, т.е.

$$\nabla_x L(X^*, \lambda, \mu) = \nabla \Phi(X^*) + \sum_{i=1}^m \lambda_i \nabla g_i(X^*) + \sum_{j=1}^l \mu_j \nabla h_j(X^*) = 0. \quad (1.9.3)$$

Теорема 1.9.1 означает, что в ее условиях вместо задачи условной оптимизации (1.9.1), (1.9.2) можно решать задачу безусловной оптимизации

$$\min_{X \in R^n} \left(\Phi(X) + \sum_{i=1}^m \lambda_i g_i(X) + \sum_{j=1}^l \mu_j h_j(X) \right).$$

Необходимым условием существования локального минимума этой задачи в некоторой точке $X^* \in D$ является условие

$$\Phi(X) + \sum_{i=1}^m \lambda_i g_i(X) + \sum_{j=1}^l \mu_j h_j(X) = 0.$$

1.10 Аналитическое решение многомерных задач нелинейного программирования

Ограничимся рассмотрением задачи нелинейного программирования с ограничениями типа неравенств:

$$\min_{X \in D} \Phi(X) = \Phi(X^*) = \Phi^*, \quad (1.10.1)$$

где $\Phi(X)$ – произвольная функция,

$$D = \{X / g(X) \geq 0\} = \{X / g_i(X) \geq 0, i \in [1, \dots, m]\} \subset R^n \quad (1.10.2)$$

– не пустое, ограниченное замкнутое множество.

Рассмотрим прямое решение задачи (1.10.1), (1.10.2) (без использования теоремы Куна-Таккера), а также решение этой задачи на основе использования теоремы Куна-Таккера

Схема прямого решения задачи нелинейного программирования:

1. Из условия $\nabla \Phi(X^*) = 0$ определяем все стационарные точки функции $\Phi(X)$ в области D ;
2. Определяем все критически точки функции (точки не дифференцируемости) функции $\Phi(X)$ в области D ;
3. Для каждой из границ области D (ограничивающих функций) решаем соответствующую задачу на условный минимум:

- из уравнения $g_i(X) \geq 0, i \in [1, \dots, m]$ выражаем m переменных через остальные $(n - m)$ переменных и подставляем их в выражение для функции $\Phi(X)$;
 - вместо исходной задачи условной оптимизации получаем задачу безусловной оптимизации с $(n - m)$ переменными;
 - решаем эту задачу – находим стационарные точки полученной функции, лежащие на соответствующей границе области D ;
4. Решаем задачу, аналогичную задаче, рассмотренной в п.3, для каждого из множеств, которое определяется пересечением границ области D ;
 5. Во всех отобранных точках вычисляем значения функции $\Phi(X)$ и выбираем ту (или те), в которой значение функции наименьшее.

Заметим, что в общем случае такой подход трудно реализовать на практике, поскольку далеко не всегда удастся разрешить уравнения $g_i(X) \geq 0, i \in [1, \dots, m]$ относительно указанных переменных.

Схема решения задачи нелинейного программирования с использованием теоремы Куна-Таккера:

1. Записываем функцию Лагранжа

$$L(X, \lambda) = \Phi(X) + \sum_{i=1}^m \lambda_i g_i(X), \quad (1.10.3)$$

2. Находим градиенты $\nabla \Phi(X), \nabla g_i(X), i \in [1, \dots, m]$ функций $\Phi(X), g_i(X), i \in [1, \dots, m]$;

3. Находим стационарные точки функции Лагранжа, т.е. точки, в которых градиент этой функции равен нулю:

$$\nabla_x L(X^*, \lambda) = \nabla \Phi(X^*) + \sum_{i=1}^m \lambda_i \nabla g_i(X^*) = 0, \quad (1.10.4)$$

4. Находим точки, в которых нарушаются условия регулярности ограничивающих функций.
5. Во всех стационарных точках функции, а также точках нарушения условий регулярности ограничивающих функций вычисляем значения функции $\Phi(X)$ и выбираем ту (или те), в которой значение функции наименьшее.

Пример 1.10.1. Пусть дана двумерная задача нелинейного программирования с ограничениями типа неравенств, где

$$\Phi(X) = \Phi(x_1, x_2) = (x_1 - 4)^2 + (x_2 - 4)^2, \quad (1.10.5)$$

$$D = \{X / g_1(X) = x_1 \geq 0, g_2(X) = x_2 \geq 0, g_3(X) = x_1 - x_2 + 5 \geq 0\} \subset R^2. \quad (1.10.6)$$

1. Дайте определение задачи нелинейного программирования (1.10.5), (1.10.6).

2. Изобразите на рисунке область допустимых значений вектора варьируемых параметров, формируемую ограничениями (1.10.6).

3. Найдите аналитически решение этой задачи без использования теоремы Куна-Таккера и изобразите его на рисунке.

4. Найдите аналитически решение этой задачи с использованием теоремы Куна-Таккера и изобразите его на рисунке.

Решение. 1. Задача (1.10.5), (1.10.6) является задачей выпуклого программирования: множество D есть выпуклый многогранник, а функция $\Phi(X)$ – квадратичная.

2. Область допустимых значений вектора варьируемых параметров, формируемая ограничениями (1.10.6), имеет вид, представленный на рис. 1.10.1

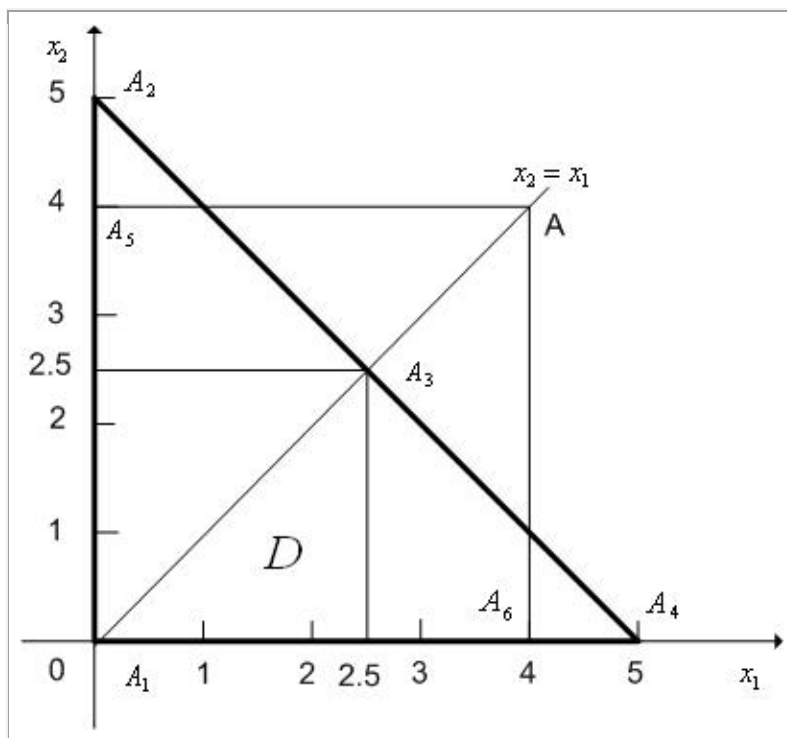


Рис. 1.10.1

3. Аналитическое решение задачи нелинейного программирования (1.10.5), (1.10.6) без использования теоремы Куна-Таккера состоит из следующих шагов.

1. Определяем стационарные точки функции $\Phi(X)$ в области D .

$$\nabla \Phi(X) = \begin{pmatrix} \frac{\partial \Phi(X)}{\partial x_1} \\ \frac{\partial \Phi(X)}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2(x_1 - 4) \\ 2(x_2 - 4) \end{pmatrix} = 0 \Leftrightarrow \begin{pmatrix} x_1^* = 4 \\ x_2^* = 4 \end{pmatrix}.$$

Обозначим A найденную стационарную точку функции $\Phi(X)$. Легко видеть (см. рис. 1.10.1), что точка A не принадлежит множеству D и должна быть исключена из дальнейшего рассмотрения.

2. Определяем критические точки функции $\Phi(X)$ в области D . Функция $\Phi(X)$ всюду дифференцируема. Поэтому критические точки отсутствуют.

3. Для каждой из границ области D решаем соответствующую задачу на условный минимум:

Граница OA_2 . Решаем задачу
$$\begin{cases} \min_{X \in R} \Phi(X) = \Phi(X^*) \\ g_1(X) = 0 \end{cases}.$$

Из условия $g_1(X) = 0$ имеем $x_1 = 0$. Подставив это значение x_1 в выражение для $\Phi(X)$, получим $\Phi(X) = (x_2 - 4)^2 + 16$. Минимум этой функции достигается в точке

A_5 с координатами $(0, 4)$: $\frac{\partial \Phi(X)}{\partial x_2} = 2(x_2 - 4) = 0 \Leftrightarrow x_2^* = 4$.

Граница OA_4 . Решаем задачу
$$\begin{cases} \min_{X \in R} \Phi(X) = \Phi(X^*) \\ g_2(X) = 0 \end{cases}.$$

Аналогично предыдущему имеем точку A_6 с координатами $(0, 4)$.

Граница A_2A_4 . Решаем задачу
$$\begin{cases} \min_{X \in R} \Phi(X) = \Phi(X^*) \\ g_3(X) = 0 \end{cases}.$$

Из условия $g_3(X) = 0$ имеем $x_1 = -x_2 + 5$. Подставив это значение x_1 в выражение для $\Phi(X)$, получим $\Phi(X) = 2x_2^2 - 10x_2 + 17$. Минимум этой функции достигается в точке A_3 с координатами $(2.5, 2.5)$:

$\frac{\partial \Phi(X)}{\partial x_2} = 4x_2 - 10 = 0 \Leftrightarrow x_2^* = 2.5, x_1^* = -2.5 + 2.5 = 2.5$.

4. Пересечением границ области D являются точки $A_1 - A_2 - A_4$.

5. Значения функции $\Phi(X)$ в отобранных точках приведены в табл. 1.10.1.

Таблица 1.10.1

Точка	A_1	A_2	A_3	A_4	A_5	A_6
Координаты	(0,0)	(0,5)	(2.5,2.5)	(5,0)	(0,4)	(4,0)
Значение $\Phi(X)$	32	17	4.5	17	16	16

Из табл. 1.10.1 следует, что искомое минимальное значение $\Phi(X)$ достигается в точке $(2.5, 2.5)$ и равно 4.5.

4. Аналитическое решение задачи нелинейного программирования (1.10.5), (1.10.6) с использованием теоремы Куна-Таккера состоит из следующих шагов.

1. Записываем функцию Лагранжа:

$$L(X, \lambda) = (x_1 - 4)^2 + (x_2 - 4)^2 + \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 (-x_1 - x_2 + 5).$$

2. Находим градиенты функций $\Phi(X), g_i(X), i \in [1, \dots, 3]$:

$$\nabla \Phi(X) = \begin{pmatrix} 2(x_1 - 4) \\ 2(x_2 - 4) \end{pmatrix}, \quad \nabla g_1(X) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \nabla g_2(X) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \nabla g_3(X) = \begin{pmatrix} -1 \\ -1 \end{pmatrix}.$$

3. Записываем необходимое условие минимума функции Лагранжа:

$$\begin{cases} 2(x_1 - 4) + \lambda_1 - \lambda_3 = 0 \\ 2(x_2 - 4) + \lambda_2 - \lambda_3 = 0 \end{cases} \quad (1.10.7)$$

4. а). Положим, что ни одно из ограничений не является активным ограничением (точка лежит внутри области D). В этом случае можно положить $\lambda_1 = \lambda_2 = \lambda_3 = 0$ (напомним, что теорема Куна-Таккера не запрещает этого). Тогда из (1.10.7) имеем стационарную точку функции A с координатами $(4, 4)$. Точка лежит вне области D и из рассмотрения исключается.

б). Пусть активным является ограничение $g_1(X) \geq 0$ т.е. пусть $x_1 = 0$. Тогда можно положить $\lambda_2 = \lambda_3 = 0$. При этом из (1.10.7) следует $\lambda_1 = 8, x_2 = 4$. Таким образом, имеем стационарную точку функции A_5 с координатами $(0, 4)$.

в). Пусть активным является ограничение $g_2(X) \geq 0$, т.е. пусть $x_2 = 0$. Тогда можно положить $\lambda_1 = \lambda_3 = 0$. При этом из (1.10.7) следует $\lambda_2 = 8, x_1 = 4$. Таким образом, имеем стационарную точку функции A_6 с координатами $(4, 0)$.

г). Пусть активным является ограничение $g_3(X) \geq 0$, т.е. пусть $-x_1 - x_2 + 5 = 0$. Тогда можно положить $\lambda_1 = \lambda_2 = 0$. При этом из (1.10.7) следует система

$$\begin{cases} 2(x_1 - 4) - \lambda_3 = 0 \\ 2(x_2 - 4) - \lambda_3 = 0 \end{cases} \quad (1.10.8)$$

Вычитая из первого уравнения второе, получим $(x_1 - 4) - (x_2 - 4) = 0 \Leftrightarrow x_1 = x_2$. Отсюда, из условия $-x_1 - x_2 + 5 = 0$ и системы (1.10.8) имеем $x_1 = 2.5, x_2 = 2.5, \lambda_3 = -3$. Таким образом, имеем стационарную точку A_3 с координатами $(2.5, 2.5)$.

д) Пусть активными являются ограничения $g_1(X) \geq 0, g_2(X) \geq 0$, т.е. пусть $x_1 = 0, x_2 = 0$. Тогда можно положить $\lambda_3 = 0$. При этом из (1.10.7) следует $\lambda_1 = 8,$

$\lambda_2 = 8$. Таким образом, имеем стационарную точку функции A_1 с координатами $(0, 0)$.

е) Аналогично получаем стационарную точку A_2 с координатами $(0, 5)$.

ж) Аналогично получаем стационарную точку A_4 с координатами $(5, 0)$.

5. Легко видеть, что точки, в которых нарушаются условия регулярности ограничивающих функций, отсутствуют. Значения функции $\Phi(X)$ в отобранных точках совпали с приведенными в табл. 1.10.1. Искомое минимальное значение $\Phi(X)$ достигается в точке $(2.5, 2.5)$ и равно 4.5.

1.11 Классификация методов решения детерминированных задач оптимизации

Особенность задач оптимизации состоит в том, что вычисление значения критерия оптимальности и значений ограничивающих функций при фиксированных значениях параметров X может требовать больших затрат компьютерного времени. В связи с этим возникает проблема решения задачи оптимизации при наименьшем числе испытаний.

Рассмотрим для простоты записи только ограничения типа неравенств:

$$\min_{X \in D} \Phi(X) = \Phi(X^*), \quad D = \{X / g(X) \geq 0\}.$$

Испытанием называется операция однократного вычисления функций $\Phi(X)$, $g(X)$ и, быть может, их производных, в некоторой точке X , вообще говоря, не обязательно принадлежащей множеству D .

Далее будем говорить, что детерминированная задача оптимизации решается с помощью **поискового метода оптимизации**, если используется следующая процедура поиска оптимального решения X^* :

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

$$X^0, X^{r+1} = F_{r+1} \left(X^0, \Phi(X^0), g(X^0), \dots, X^r, \Phi(X^r), g(X^r) \right) \quad (1.11.1)$$

где X^0 – начальное приближение;

- в качестве решения задачи берется вектор X^* , который находится из условия

$$\Phi^* = \Phi(X^*) = \min_{r \in [0, N]} \Phi(X^r). \quad (1.11.2)$$

Классификация по наличию или отсутствию ограничений на вектор варьируемых параметров

Метод поиска, ориентированный на решение задач безусловной оптимизации, называется **методом безусловной оптимизации**.

Аналогично, метод поиска, ориентированный на решение задач условной оптимизации, называется **методом условной оптимизации**.

Классификация по размерности вектора X

Если в формулах (1.11.1), (1.11.2) X^r есть скаляр, то метод поиска называется **одномерным методом поиска**; если X^r есть вектор ($n > 1$), то метод поиска называется **многопараметрическим методом поиска**.

Классификация по характеру искомого решения

Если метод поиска гарантирует отыскание только локального минимума функции $\Phi(X)$, то метод называется **методом локального поиска**.

Если делается попытка отыскать глобальный минимум $\Phi(X)$, то метод называется **методом глобального поиска**.

Сразу отметим, что удовлетворительных с точки зрения вычислительной эффективности методов глобального поиска не существует.

Классификация по характеру функции F_r

Если функции F_r , $r \in [1, N]$ являются детерминированными, то метод поиска называется **детерминированным методом поиска**.

Если функции F_r содержат случайные параметры, то метод поиска называется **стохастическим методом поиска**.

Методы пассивного и последовательного поиска

Если все точки X^r , $r = 0, 1, 2, \dots, N-1$ назначаются заранее (до проведения испытаний), то метод поиска называется **пассивным методом поиска**.

Если точка X^{r+1} определяется на основе всей или части информации об испытаниях в предыдущих точках, то метод называется **последовательным методом поиска**.

Классификация по количеству предыдущих учитываемых шагов

Если при вычислении координат точки X^{r+1} учитывается информация только об одном (предыдущем) испытании, то метод поиска называется **одношаговым методом поиска**.

Схема одношагового последовательного метода поиска

$$X^{r+1} = F_{r+1}(X^r, \Phi(X^r), g(X^r))$$

Если при вычислении координат точки X^{r+1} учитывается информация о $s > 1$ предыдущих испытаниях, то метод поиска называется **многошаговым методом поиска** (конкретнее, S -шаговым).

Классификация по виду функций F_r

Если функции F_r на всех N шагах одинаковы, то метод поиска называется **итерационным методом поиска**.

Схема одношагового итерационного метода последовательного поиска:

$$X^{r+1} = F(X^r, \Phi(X^r), g(X^r))$$

Если функции F_r различны на различных шагах поиска, то метод называется **не итерационным методом поиска**.

Классификация по "близости" соседних точек, в которых производятся испытания

Если точка X^{r+1} принадлежит некоторой малой окрестности точки X^r , т.е. $X^{r+1} \in d(X^r)$, то метод поиска называется **локальным методом поиска**.

Если точка X^{r+1} может принадлежать любой точке множества D , т.е. $X^{r+1} \in D$, то метод поиска называется **нелокальным методом поиска**.

Классификация по порядку используемых производных

Если при вычислении значений функции F_r производные не используются, то метод поиска называется **прямым методом поиска** или методом поиска нулевого порядка.

Если при этом используются производные k -го порядка, то метод поиска называется **методом поиска k -го порядка**. Метод поиска первого порядка называется также **градиентным методом поиска**.

Способ выбора начальной точки X^0 и конкретная совокупность функций $\{F_r\}$ называются **алгоритмом поисковой оптимизации**.

Таким образом, понятие алгоритма является более частным по сравнению с понятием метода (одному и тому же методу могут соответствовать различные алгоритмы).

Важной проблемой при построении методов решения задач оптимизации является проблема выбора условия окончания поиска (критерия окончания поиска). Простейшими, но широко используемыми в вычислительной практике, являются следующие критерии окончания поиска:

$$\|X^{r+1} - X^r\| \leq \varepsilon_X, \quad (1.11.3)$$

где ε_X – константа, определяющая требуемую точность решения по X ;

$$\|\Phi(X^{r+1}) - \Phi(X^r)\| \leq \varepsilon_\Phi, \quad (1.11.4)$$

где ε_Φ – константа, определяющая требуемую точность решения по Φ . Здесь $\|\cdot\|$ – некоторая векторная норма (например, евклидова).

Будем далее условия окончания поиска (1.11.3), (1.11.4) называть **стандартными условиями окончания поиска** (стандартными критериями окончания поиска).

РАЗДЕЛ 2 НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ. БЕЗУСЛОВНАЯ ОДНОМЕРНАЯ ОПТИМИЗАЦИЯ

2.1 Алгоритм равномерного поиска

Рассмотрим следующую задачу условной оптимизации: найти минимум одномерной унимодальной функции $\Phi(x)$, определенной в замкнутой области допустимых значений $D = [a, b]$, $\min_{x \in [a, b]} \Phi(x) = \Phi(x^*)$.

Идея алгоритмов, относящихся к методу сокращения текущего интервала неопределенности, состоит в исключении в процессе поиска из рассмотрения тех подынтервалов, в которых в силу унимодальности функции $\Phi(x)$ точка x^* отсутствует.

Текущий интервал неопределенности будем обозначать ТИН, а его длину |ТИН|. Так что, если ТИН = $[a, b]$, то |ТИН| = $b - a$.

В алгоритме равномерного поиска испытания проводятся в точках, которые определяются путем равномерного деления интервала $[a, b]$ на N одинаковых подынтервалов. Из вычисленных значений функции $\Phi(x)$ выбирается наименьшее. Пусть это значение достигается в точке x_k . Тогда в связи с унимодальностью функции $\Phi(x)$ подынтервалы $[a, x_{k-1}]$, $[x_{k+1}, b]$ можно исключить из рассмотрения, т.е. сделать очередным интервалом неопределенности интервал $[x_{k-1}, x_{k+1}]$.

Алгоритм относится к классу пассивных методов поиска.

Алгоритм равномерного поиска

1. Выполняем присваивания $r = 1$, $a^1 = a$, $b^1 = b$, ТИН₁ = $[a^1, b^1]$.
2. На текущем ТИН строим равномерную сетку с $N + 1$ узлами (см. рис. 2.1.1).
3. Вычисляем значения функции $\Phi(x)$ в узлах построенной сетки: $\Phi(x_0^r), \dots, \Phi(x_N^r)$.
4. Находим минимальное из этих значений:
 $\min(\Phi(x_0^r), \dots, \Phi(x_N^r)) = \Phi(x_k^r)$.
5. Выполняем присваивания $a^{r+1} = x_{k-1}^r$, $b^{r+1} = x_{k+1}^r$, ТИН_{r+1} = $[a^{r+1}, b^{r+1}]$.

6. Если $|\text{ТИН}_{r+1}| \leq \varepsilon_x$, то заканчиваем вычисления. Иначе – выполняем присваивание $r = r + 1$ и переходим на п.2. Здесь ε_x – требуемая точность решения.

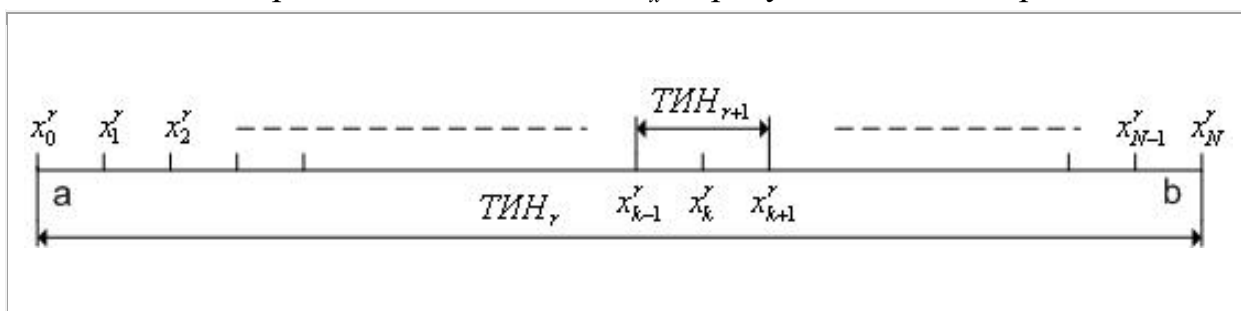


Рис. 2.1.1. Построение сетки на текущем интервале неопределенности

В качестве приближенного значения точки минимума x^* с равными основаниями может быть принята любая точка последнего текущего интервала неопределенности.

Первую итерацию приведенной схемы алгоритма равномерного поиска иллюстрирует рис.2.1.2. Легко видеть, что после одной итерации алгоритма равномерного поиска ТИН уменьшается в $\frac{N}{2}$ раз. Поэтому количество итераций r , необходимых для нахождения минимума функции с точностью ε_x , может быть найдено из условия $\left(\frac{2}{N}\right)^r (b-a) \leq \varepsilon_x$.

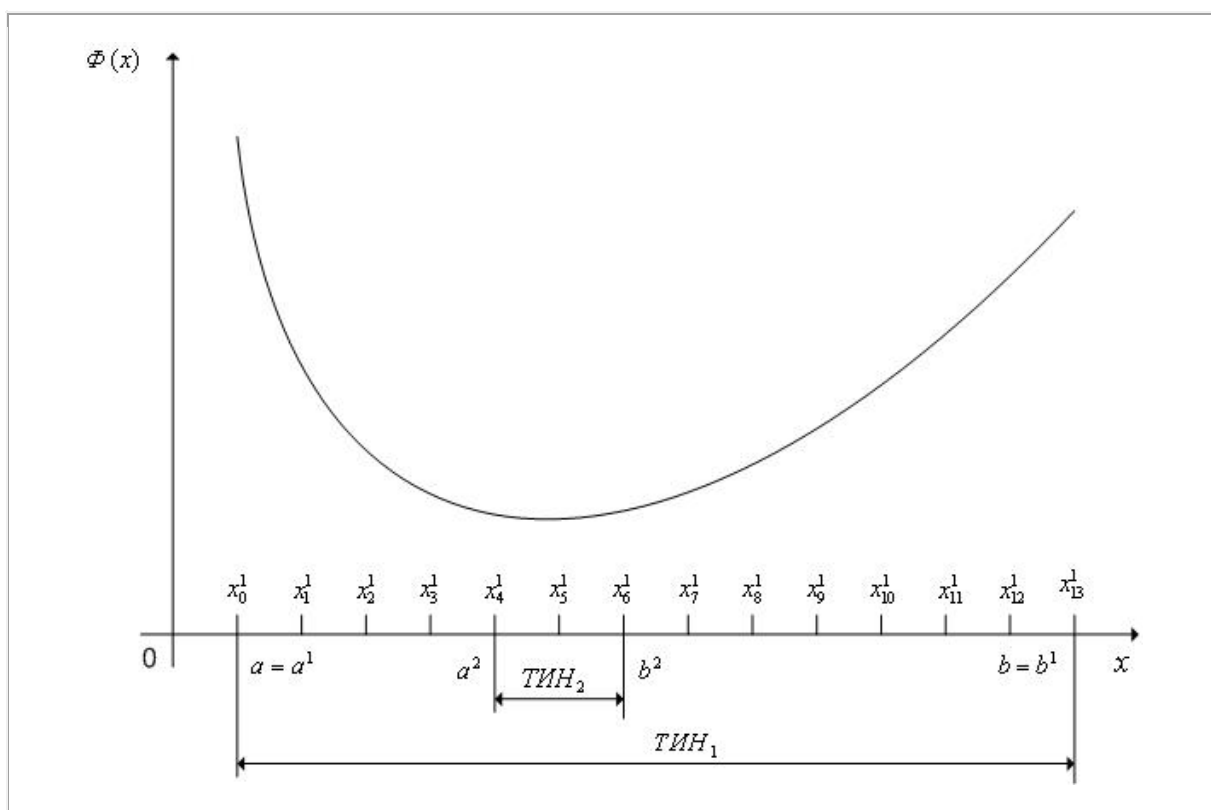


Рис. 2.1.2. Первая итерация поиск минимума одномерной унимодальной функции $\Phi(x)$ с помощью алгоритма равномерного поиска: $N = 13$

2.2 Алгоритм деления пополам

Рассмотрим следующую задачу условной оптимизации: найти минимум одномерной унимодальной функции $\Phi(x)$, определенной в замкнутой области допустимых значений $D = [a, b]$, $\min_{x \in [a, b]} \Phi(x) = \Phi(x^*)$.

В алгоритме деления пополам или алгоритме равномерного дихотомического поиска испытания проводятся парами. Координаты каждой последующей пары испытаний разнесены между собой на величину $\delta_x < \varepsilon_x$, где ε_x – требуемая точность решения. Испытания производятся в середине ТИН. По значениям $\Phi(x)$, полученным в этих точках, одна половина ТИН в силу унимодальности функции $\Phi(x)$ исключается из дальнейшего рассмотрения. Величина δ_x определяется требуемой точностью решения. Алгоритм относится к классу методов последовательного поиска.

Алгоритм деления пополам

1. Выполняем присваивания $r = 1$, $a^1 = a$, $b^1 = b$, $\text{ТИН}_1 = [a^1, b^1]$.
2. Вычисляем величины (см. рис. 2.2.1) $x_0^r = \frac{a^r - b^r}{2}$, $x_1^r = x_0^r - \frac{\delta_x}{2}$, $x_2^r = x_0^r + \frac{\delta_x}{2}$.
3. Вычисляем значения $\Phi(x_1^r), \Phi(x_2^r)$ функции $\Phi(x)$.
4. Если $\Phi(x_1^r) < \Phi(x_2^r)$, то выполняем присваивания $a^{r+1} = a^r$, $b^{r+1} = x_0^r$, $\text{ТИН}_{r+1} = [a^{r+1}, b^{r+1}]$. Иначе – выполняем присваивания $a^{r+1} = x_0^r$, $b^{r+1} = b^r$, $\text{ТИН}_{r+1} = [a^{r+1}, b^{r+1}]$.
5. Если $|\text{ТИН}_{r+1}| \leq \varepsilon_x$, то заканчиваем вычисления. Иначе – выполняем присваивание $r = r + 1$ и переходим на п.2.

В качестве приближенного значения точки минимума с равными основаниями может быть принята любая точка последнего текущего интервала неопределенности.

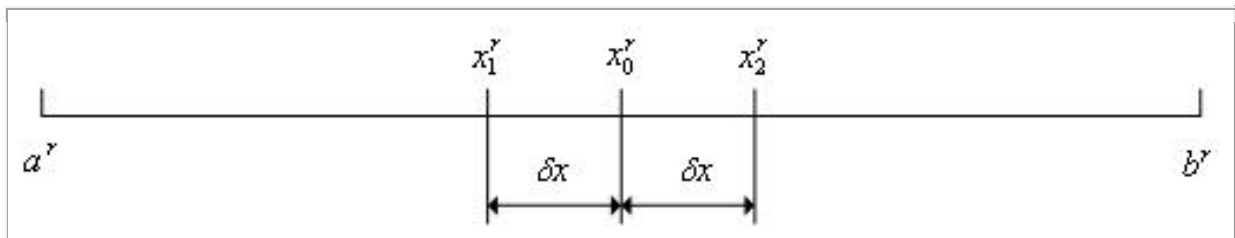


Рис. 2.2.1. К определению величин x_0^r, x_1^r, x_2^r

Приведенную схему алгоритма равномерного дихотомического поиска иллюстрирует рис. 2.2.2.

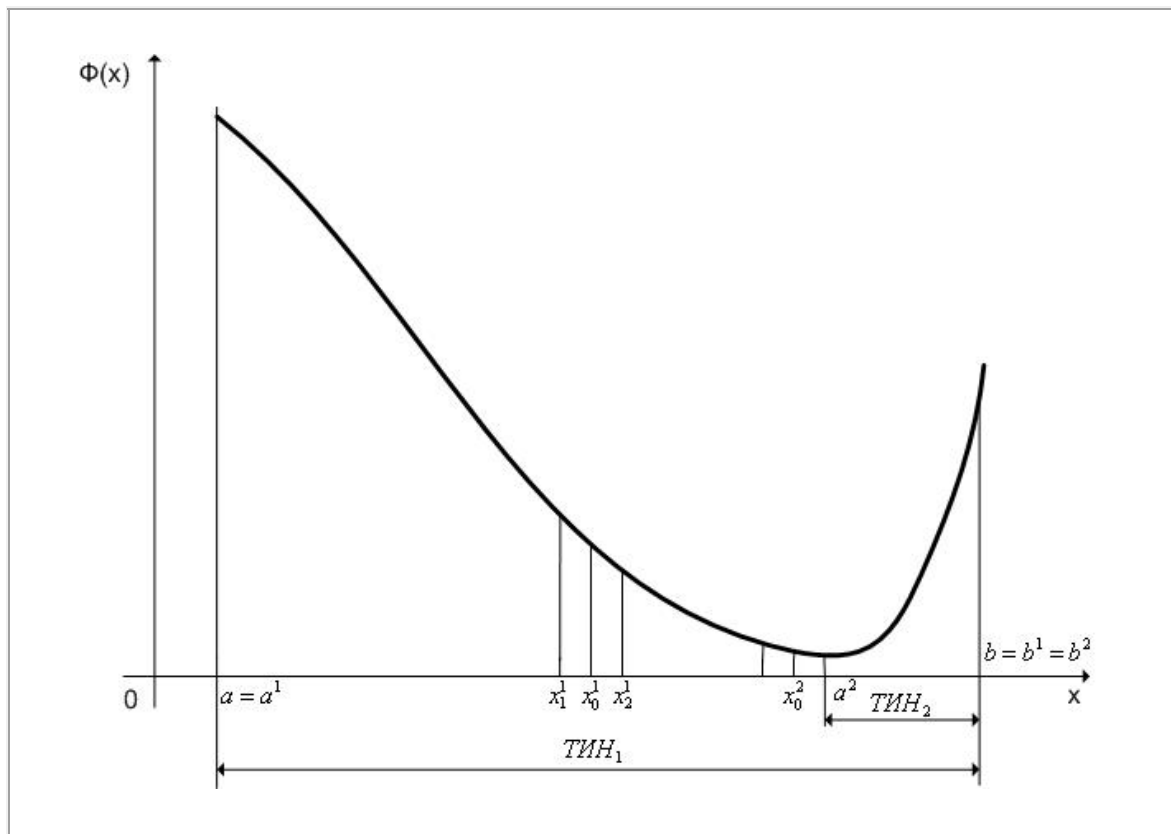


Рис. 2.2.2. Первые две итерации поиска минимума одномерной унимодальной функции с помощью алгоритма равномерного дихотомического поиска

Легко видеть, что после одной итерации алгоритма равномерного поиска ТИН уменьшается в 2 раза. Поэтому количество итераций r , необходимых для нахождения минимума функции с точностью ε_x , находится из условия

$$\frac{b-a}{2^r} \leq \varepsilon_x.$$

2.3 Алгоритм Фибоначчи

Рассмотрим следующую задачу условной оптимизации: найти минимум одномерной унимодальной функции $\Phi(x)$, определенной в замкнутой области допустимых значений $D = [a, b]$, $\min_{x \in [a, b]} \Phi(x) = \Phi(x^*)$.

Числа Фибоначчи задаются следующим рекуррентным уравнением:

$$i_N = i_{N-1} + i_{N-2}, N \geq 2, i_0 = i_1 = 1. \quad (2.3.1)$$

Числа Фибоначчи i_0, \dots, i_9 приведены в таблице 2.3.1.

Таблица 2.3.1

N	0	1	2	3	4	5	6	7	8	9	...
i_N	1	1	2	3	5	8	13	21	34	55	...

Общее выражение для N -го числа Фибоначчи можно получить из решения уравнения (2.3.1):

$$i_N = \frac{\left(\frac{1}{\tau}\right)^{N+1} - (-\tau)^{N+1}}{\sqrt{5}}, \text{ где } \tau = \frac{\sqrt{5}-1}{2} \approx 0,618$$

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

$$i_N \approx \frac{\left(\frac{1}{\tau}\right)^{N+1}}{\sqrt{5}} \quad (2.3.2)$$

Отсюда следует, что $\frac{i_{N-1}}{i_N} \approx \tau$. Т.е. отношение двух соседних чисел Фибоначчи примерно постоянно и равно τ .

Алгоритм Фибоначчи относится к классу поисковых методов оптимизации и включает в себя два этапа.

Алгоритм Фибоначчи

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

1. Вычисляем величины $x_1^r = a^r + |\text{ТИН}_r| \frac{i_{N-1-r}}{i_{N+1-r}}$, $x_2^r = a^r + |\text{ТИН}_r| \frac{i_{N-r}}{i_{N+1-r}}$.
2. Вычисляем значения $\Phi(x_1^r), \Phi(x_2^r)$ функции $\Phi(x)$.
3. Если $\Phi(x_1^r) < \Phi(x_2^r)$, то выполняем присваивания $a^{r+1} = a^r$, $b^{r+1} = x_2^r$, $\text{ТИН}_{r+1} = [a^{r+1}, b^{r+1}]$. Иначе – выполняем присваивания $a^{r+1} = x_1^r$, $b^{r+1} = b^r$, $\text{ТИН}_{r+1} = [a^{r+1}, b^{r+1}]$.

Алгоритм Фибоначчи обладает тем свойством, что после выполнения $(N-1)$ -й итерации имеет место следующая ситуация: $x_1^{N-1} = x_2^{N-1} = x^{N-1}$. Т.е. в результате $(N-1)$ -й итерации сужение текущего интервала неопределенности не происходит: $\text{ТИН}_{N-1} = [a^{N-1}, b^{N-1}] = \text{ТИН}_{N-2} = [a^{N-2}, b^{N-2}]$.

Второй этап призван решить по какую сторону от точки x^{N-1} лежит точка минимума функции $\Phi(x)$.

Второй этап выполняется по следующей схеме:

1. Находим точку $x^N = x^{N-1} + \delta_x$, где $\delta_x \in \text{ТИН}_{N-1}$ – свободный параметр алгоритма.
2. Вычисляем значение функции $\Phi(x^N)$.
3. Если $\Phi(x^N) > \Phi(x^{N-1})$, то выполняем присваивания $\text{ТИН}_N = [a^{N-1}, x^{N-1}]$
Иначе – выполняем присваивания $\text{ТИН}_N = [x^{N-1}, b^{N-1}]$.

В качестве приближенного значения точки минимума x^* с равными основаниями может быть принята любая точка ТИН_N .

Некоторые свойства алгоритма Фибоначчи

Утверждение 2.3.1. Для любого $r \in [1, \dots, N-2]$ алгоритм Фибоначчи обладает следующим свойством: одна из точек x_1^{r+1}, x_2^{r+1} совпадает с одной из точек x_1^r, x_2^r (см. рис. 2.3.1).

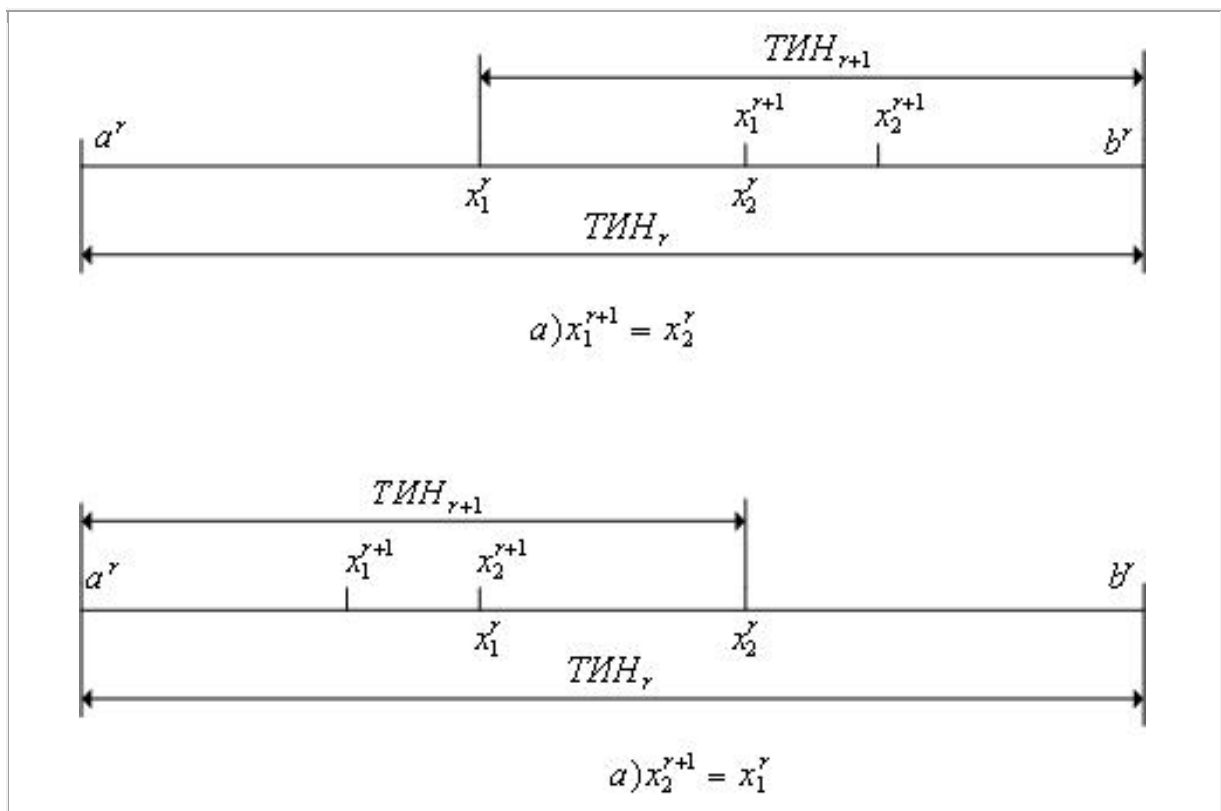


Рис. 2.3.1. К утверждению 2.3.1

Доказательство. Пусть на r -й итерации $\Phi(x_1^r) \leq \Phi(x_2^r)$ – ситуация б на рис. 2.3.1. В соответствии с алгоритмом Фибоначчи $\text{ТИН}_{r+1} = [a^r, x_2^r]$ причем, очевидно, $x_1^r = \text{ТИН}_{r+1}$.

Рассмотрим точку:

$$x_2^{r+1} = a^r + \left| \text{ТИН}_{r+1} \right| \frac{i_{N-1-r}}{i_{N-r}} = a^r + (x_2^r - a^r) \frac{i_{N-1-r}}{i_{N-r}}.$$

Подставим сюда значение координаты точки:

$$x_2^r = a^r + \left| \text{ТИН}_r \right| \frac{i_{N-r}}{i_{N+1-r}},$$

$$x_2^{r+1} = a^r + \left(a^r + \left| \text{ТИН}_r \right| \frac{i_{N-r}}{i_{N+1-r}} - a^r \right) \frac{i_{N-1-r}}{i_{N-r}} = a^r + \left| \text{ТИН}_r \right| \frac{i_{N-1-r}}{i_{N-r}} = x_1^r.$$

Аналогично проводится доказательство для случая $\Phi(x_1^r) > \Phi(x_2^r)$ – ситуация а на рис. 2.3.1.

Указанное свойство алгоритма Фибоначчи позволяет на каждой итерации (кроме первой) производить испытания только в одной точке.

Утверждение 2.3.2. Точки x_1^r, x_2^r расположены симметрично относительно концов текущего интервала неопределенности $\text{ТИН}_r = [a^r, b^r]$, т.е. расстояние точки x_1^r до точки a^r равно расстоянию точки x_2^r до точки b^r или, что тоже самое, $x_2^r - a^r = b^r - x_1^r$.

Доказательство. В соответствии с алгоритмом Фибоначчи имеем:

$$x_1^r - a^r = a^r + (b^r - a^r) \frac{i_{N-1-r}}{i_{N+1-r}} - a^r = (b^r - a^r) \frac{i_{N-1-r}}{i_{N+1-r}},$$

$$b^r - x_2^r = b^r - a^r - (b^r - a^r) \frac{i_{N-r}}{i_{N+1-r}} - a^r = (b^r - a^r) \frac{i_{N-1-r} - i_{N-r}}{i_{N+1-r}}.$$

Но из формулы (2.3.1) следует, что $i_{N+1-r} = i_{N-r} + i_{N-1-r}$. Подставляя это в предыдущую формулу, получим: $b^r - x_2^r = (b^r - a^r) \frac{i_{N-1-r}}{i_{N+1-r}} = x_1^r - a^r$.

Утверждение 2.3.3. В результате любой итерации $r \in [1, \dots, N-2]$ алгоритма Фибоначчи длина текущего интервала неопределенности уменьшается в $\frac{i_{N-r}}{i_{N+1-r}}$ раз.

Доказательство. Поскольку $x_2^r - a^r = b^r - x_1^r$ (см. утверждения 2.3.2), достаточно рассмотреть только один из интервалов $[x_2^r - a^r], [b^r - x_1^r]$. Рассмотрим первый из указанных интервалов:

$$x_2^r - a^r = a^r + (b^r - a^r) \frac{i_{N-r}}{i_{N+1-r}} - a^r = (b^r - a^r) \frac{i_{N-r}}{i_{N+1-r}}.$$

Утверждение 2.3.4. При достаточно больших N в результате одной итерации алгоритма Фибоначчи длина текущего интервала неопределенности уменьшается примерно в τ раз.

Доказательство. Справедливость утверждения следует из утверждения 2.3.3 и из того факта, что при достаточно больших N имеем (см. (2.3.2)):

$$\frac{i_{N-r}}{i_{N+1-r}} \approx \frac{\left(\frac{1}{\tau}\right)^{N+1-r}}{\left(\frac{1}{\tau}\right)^{N+2-r}} = \tau$$

Из утверждения 2.3.4 следует, что при достаточно больших N алгоритм Фибоначчи практически идентичен алгоритму золотого сечения (см. следующий параграф 2.4).

Утверждение 2.3.5. В результате N итераций алгоритма Фибоначчи длина текущего интервала неопределенности становится равной

$$|\text{ТИН}_N| = \frac{b-a}{i_N}$$

Доказательство. Из утверждения 2.3.3 следует, что:

- после первой итерации длина ТИН равна $(b-a) \frac{i_{N-1}}{i_N}$;
- после второй итерации $-(b-a) \frac{i_{N-1}}{i_N} \frac{i_{N-2}}{i_{N-1}} = (b-a) \frac{i_{N-2}}{i_N}$;
- после итерации номер $(N-2)$ длина ТИН равна $(b-a) \frac{i_2}{i_N} = (b-a) \frac{2}{i_N}$
- после итерации номер $(N-1)$ длина ТИН не меняется;
- после итерации номер N длина ТИН уменьшается в два раза и становится равной $\frac{b-a}{i_N}$.

Поэтому количество итераций N , необходимых для нахождения минимума функции с точностью ε_x , находится из условия $\varepsilon_x \leq \frac{b-a}{i_N}$.

Пример 2.3.1. С использованием $N=6$ итераций алгоритма Фибоначчи найдите минимум одномерной унимодальной функции $\Phi(x) = 2(x-12)^2 + 3$, где $x \in [a, b] = [0, 13]$. Положите $\delta_x = 0.1$.

Решение. Первый этап.

Итерация № 1. $r=1$ $\text{ТИН}_1 = [a_1, b_1] = [a, b] = [0, 13]$:

$$\begin{cases} x_1^1 = a_1 + \left| \text{ТИН}_1 \right| \frac{i_{N-2}}{i_N} = 13 \frac{5}{13} = 5, \\ x_2^1 = a_1 + \left| \text{ТИН}_1 \right| \frac{i_{N-1}}{i_{N+1-r}} = 13 \frac{8}{13} = 8; \end{cases}$$

$$\begin{cases} \Phi(x_1^1) = 2(5-12)^2 + 3 = 101 \\ \Phi(x_2^1) = 2(8-12)^2 + 3 = 39 \end{cases} \Leftrightarrow \Phi(x_1^1) > \Phi(x_2^1)$$

Итерация №2. $r = 2$ ТИН₂ = $[x_1^1, b_1] = [5, 13] = [a_2, b_2]$:

$$\begin{cases} x_1^2 = a_2 + \left| \text{ТИН}_2 \right| \frac{i_{N-3}}{i_{N-1}} = 5 + 8 \frac{3}{8} = 8 = x_2^1, \\ x_2^2 = a_2 + \left| \text{ТИН}_2 \right| \frac{i_{N-2}}{i_{N-1}} = 5 + 8 \frac{5}{8} = 10; \\ \Phi(x_1^2) = \Phi(x_2^1) = 39 \\ \Phi(x_2^2) = 2(10-12)^2 + 3 = 11 \end{cases} \Leftrightarrow \Phi(x_1^2) > \Phi(x_2^2)$$

Итерация №3. $r = 3$ ТИН₃ = $[x_1^2, b_2] = [8, 13] = [a_3, b_3]$:

$$\begin{cases} x_1^3 = a_3 + \left| \text{ТИН}_3 \right| \frac{i_{N-4}}{i_{N-2}} = 8 + 5 \frac{2}{5} = 10 = x_2^2, \\ x_2^3 = a_3 + \left| \text{ТИН}_3 \right| \frac{i_{N-3}}{i_{N-2}} = 8 + 5 \frac{3}{5} = 11; \\ \Phi(x_1^3) = \Phi(x_2^2) = 11 \\ \Phi(x_2^3) = 2(11-12)^2 + 3 = 5 \end{cases} \Leftrightarrow \Phi(x_1^3) > \Phi(x_2^3)$$

Итерация №4. $r = 4$ ТИН₄ = $[x_1^3, b_3] = [10, 13] = [a_4, b_4]$:

$$\begin{cases} x_1^4 = a_4 + \left| \text{ТИН}_4 \right| \frac{i_{N-5}}{i_{N-3}} = 10 + 3 \frac{1}{3} = 11 = x_2^3, \\ x_2^4 = a_4 + \left| \text{ТИН}_4 \right| \frac{i_{N-4}}{i_{N-3}} = 10 + 3 \frac{2}{3} = 12; \\ \Phi(x_1^4) = \Phi(x_2^3) = 5 \\ \Phi(x_2^4) = 2(12-12)^2 + 3 = 3 \end{cases} \Leftrightarrow \Phi(x_1^4) > \Phi(x_2^4)$$

Итерация №5. $r = 5$ ТИН₅ = $[x_1^4, b_4] = [11, 13] = [a_5, b_5]$:

$$\begin{cases} x_1^5 = a_5 + \left| \text{ТИН}_5 \right| \frac{i_{N-6}}{i_{N-4}} = 11 + 1 = 12 = x_2^4, \\ x_2^5 = a_5 + \left| \text{ТИН}_5 \right| \frac{i_{N-5}}{i_{N-4}} = 11 + 1 = 12 = 12; \\ \Phi(x_1^5) = \Phi(x_2^5) = \Phi(x_2^4) = 3 \end{cases}$$

Второй этап.

Итерация №6. $r = 6$ $\text{ТИН}_5 = \text{ТИН}_6 = [11, 13]$:

$$\begin{cases} x_1^6 = x_1^5 = x_2^5 = 12, \\ x_2^6 = x_1^5 + \delta_x = 12.1; \\ \begin{cases} \Phi(x_1^6) = \Phi(x_1^5) = 3 \\ \Phi(x_2^6) = 2(0.1)^2 + 3 = 3.02 \end{cases} \Leftrightarrow \Phi(x_2^6) > \Phi(x_1^6). \end{cases}$$

Ответ: $\text{ТИН} = [11, 12]$.

2.4 Алгоритм золотого сечения

Рассмотрим следующую задачу условной оптимизации: найти минимум одномерной унимодальной функции $\Phi(x)$, определенной в замкнутой области допустимых значений $D = [a, b]$, $\min_{x \in [a, b]} \Phi(x) = \Phi(x^*)$.

Свойства золотого сечения

Рассмотрим интервал $[a, b]$ (см. рис. 2.4.1).

Говорят, что точка c выполняет золотое сечение интервала $[a, b]$, если

$$\frac{c-a}{b-a} = \tau \quad (2.4.1)$$

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

$$\tau^2 + \tau - 1 = 0 \quad (2.4.2)$$

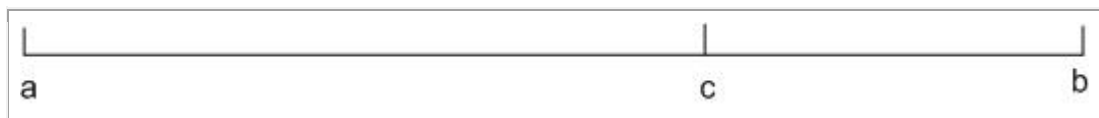


Рис. 2.4.1. К определению золотого сечения отрезка

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

Действительно, $\frac{b-c}{b-a} = \frac{(b-a) - (c-a)}{b-a} = 1 - \frac{c-a}{b-a} = 1 - \tau$.

Алгоритм золотого сечения

Алгоритм золотого сечения относится к классу последовательных методов поиска.

1. Выполняем присваивания $r = 1$, $a^1 = a$, $b^1 = b$, $\text{ТИН}_1 = [a^1, b^1]$.

2. Вычисляем величины (см. рис. 2.4.2)

$$x_1^r = b^r - (b^r - a^r)\tau, \quad x_2^r = a^r + (b^r - a^r)\tau \quad (2.4.3)$$

3. Вычисляем значения $\Phi(x_1^r), \Phi(x_2^r)$ функции $\Phi(x)$.

4. Если $\Phi(x_1^r) < \Phi(x_2^r)$, то выполняем присваивания $a^{r+1} = a^r$, $b^{r+1} = x_2^r$,

$\text{ТИН}_{r+1} = [a^{r+1}, b^{r+1}]$. Иначе – выполняем присваивания $a^{r+1} = x_1^r$, $b^{r+1} = b^r$,

$\text{ТИН}_{r+1} = [a^{r+1}, b^{r+1}]$.

5. Если $|\text{ТИН}_{r+1}| \leq \varepsilon_x$, то заканчиваем вычисления. Иначе – выполняем присваивание $r = r + 1$ и переходим на п.2. Здесь ε_x – требуемая точность решения.

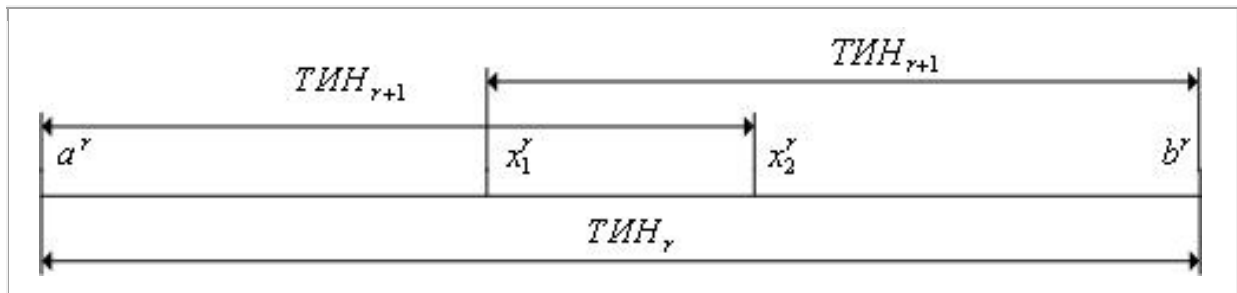


Рис. 2.4.2. К определению величин x_1^r, x_2^r

В качестве приближенного значения точки минимума x^* с равными основаниями может быть принята любая точка последнего текущего интервала неопределенности.

Некоторые свойства алгоритма золотого сечения

Утверждение 2.4.1. Точки x_1^r, x_2^r расположены симметрично относительно концов текущего интервала неопределенности.

Действительно, из (2.4.3) 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 2.4.2. Для любого $r \geq 1$ алгоритм золотого сечения обладает следующим свойством: одна из точек x_1^{r+1}, x_2^{r+1} совпадает с одной из точек x_1^r, x_2^r .

Доказательство. Пусть на r -й итерации $\Phi(x_1^r) < \Phi(x_2^r)$. В соответствии с алгоритмом золотого сечения $\text{ТИН}_{r+1} = [a^r, x_2^r]$ причем, очевидно,

$x_1^r \in \text{ТИН}_{r+1}$. Для того, чтобы доказать справедливость утверждения достаточно показать, что верно отношение

$$\frac{x_1^r - a^r}{x_2^r - a^r} = \tau \quad (2.4.4)$$

Из соотношений (2.4.3) следует, что

$$\begin{aligned} b^r - x_1^r &= (b^r - a^r)\tau \\ b^r - x_1^r - a^r + a^r &= (b^r - a^r)\tau \\ (b^r - a^r) - (x_1^r - a^r) &= (b^r - a^r)\tau \\ x_1^r - a^r &= (b^r - a^r)(1 - \tau). \end{aligned}$$

Аналогично имеем $x_2^r - a^r = (b^r - a^r)\tau$.

Разделив первый из этих результатов на второй, получим:

$$\frac{x_1^r - a^r}{x_2^r - a^r} = \frac{1 - \tau}{\tau}. \quad (2.4.5)$$

Из уравнения (2.4.2) следует, что $1 - \tau = \tau^2$. Отсюда и из (2.4.5) следует справедливость (2.4.4).

Аналогично проводится доказательство для случая $\Phi(x_1^r) > \Phi(x_2^r)$.

Указанное свойство алгоритма золотого сечения позволяет на каждой итерации (кроме первой) производить испытания только в одной точке.

Из схемы алгоритма золотого сечения имеем.

Утверждение 2.4.3. В результате одной итерации алгоритма золотого сечения длина текущего интервала неопределенности сокращается в τ раз. Поэтому количество итераций N , необходимых для нахождения минимума функции с точностью \mathcal{E}_x , находится из условия $\mathcal{E}_x \leq (b - a)\tau^N$.

Из утверждения 2.4.3 и результатов предыдущего параграфа следует, что при достаточно больших N алгоритм Фибоначчи практически идентичен алгоритму золотого сечения (см. параграф 2.3).

2.5 Метод хорд (метод секущих)

Рассмотрим следующую задачу условной оптимизации: найти минимум одномерной унимодальной функции $\Phi(x)$, определенной в замкнутой области допустимых значений $D = [a, b]$, $\min_{x \in [a, b]} \Phi(x) = \Phi(x^*)$.

Необходимым условием минимума функции $\Phi(x)$ является условие:

$$\Phi'(x) = 0, x \in [a, b]. \quad (2.5.1)$$

Рассматриваемый в данном параграфе метод первого порядка решения задачи основан на поиске стационарной точки функции $\Phi(x)$, т.е. на решении за-

дачи (2.5.1), которая представляет собой задачу нахождения корней функции $\Phi'(x)$, принадлежащих интервалу $[a, b]$.

Метод хорд ориентирован на нахождение корня уравнения (2.5.1) в случае, когда на границах интервала $[a, b]$ знаки производной $\Phi'(x)$ различны. Такая ситуация, очевидно, возможна, если точка минимума функции $\Phi(x)$ является внутренней точкой интервала $[a, b]$ – см. рис. 2.5.1.

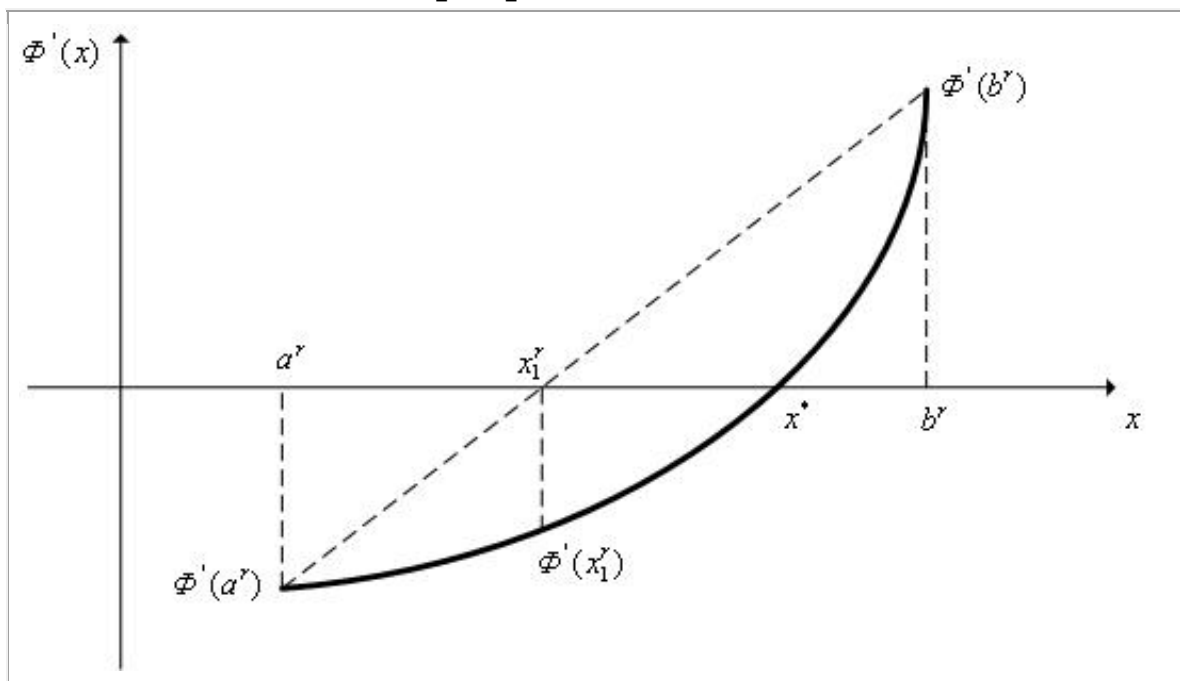


Рис. 2.5.1. К схеме метода хорд

Далее нам понадобится значение x_1^r . Из подобия треугольника $a^r, x_1^r, \Phi'(a^r)$ и треугольника $b^r, x_1^r, \Phi'(b^r)$ имеем $\frac{\Phi'(b^r)}{\Phi'(a^r)} = \frac{b^r - x_1^r}{x_1^r - a^r}$.

Отсюда следует, что

$$x_1^r = b^r - \frac{\Phi'(b^r)(b^r - a^r)}{\Phi'(b^r) - \Phi'(a^r)} \quad (2.5.2)$$

Алгоритм метода хорд

1. Выполняем присваивания $r = 1$, $a^1 = a$, $b^1 = b$.
2. Вычисляем значения производных $\Phi'(a^r)$, $\Phi'(b^r)$.
3. Если производные $\Phi'(a^r)$, $\Phi'(b^r)$ имеют одинаковые знаки – завершаем вычисления (точки a, b выбраны неверно).

4. По формуле (2.5.2) вычисляем приближение x_1^r к стационарной точке функции $\Phi(x)$, значение производной $\Phi'(x_1^r)$.
5. Если $|\Phi'(x_1^r)| \leq \varepsilon$, где ε – требуемая точность решения, то принимаем $x^* \approx x_1^r$ и заканчиваем вычисления.
6. Если производные $\Phi'(a^r)$, $\Phi'(x_1^r)$ имеют разные знаки, то выполняем присваивания $a^{r+1} = a^r$, $b^{r+1} = x_1^r$, $r = r + 1$ и переходим на п.4.
7. Если производные $\Phi'(x_1^r)$, $\Phi'(b^r)$ имеют разные знаки, (как на рис. 2.5.1.) то выполняем присваивания $a^{r+1} = x_1^r$, $b^{r+1} = b^r$, $r = r + 1$ и переходим на п.4.

В случае квадратичной функции $\Phi(x)$ производная этой функции $\Phi'(x)$ линейна. Поэтому метод хорд гарантирует нахождение стационарной точки функции $\Phi(x)$ всего за одну итерацию.

Поскольку поиск заканчивается при выполнении условия $|\Phi'(x_1^r)| \leq \varepsilon$, возможно появление ложных корней. Например, для уравнения $x^2 + 0.0001 = 0$ ложный корень $x = 0$ появляется в том случае, если $\varepsilon > 0.0001$. В подобных случаях увеличивая точность поиска, можно избавиться от ложных корней. Однако, возможны уравнения, для которых такой подход не приводит к успеху. Например, уравнение $\frac{1}{x} = 0$ не имеет действительных корней, однако для сколь угодно малого ε найдется точка, удовлетворяющая условию окончания поиска.

Возможна модификация метода хорд, когда значения производной $\Phi'(x)$ вычисляются приближенно с использованием первых разностей. В этом случае метод становится прямым (нулевого порядка).

2.6 Метод касательных (метод Ньютона решения нелинейных уравнений)

Рассмотрим следующую задачу условной оптимизации: найти минимум одномерной унимодальной функции $\Phi(x)$, определенной в замкнутой области допустимых значений $D = [a, b]$, $\min_{x \in [a, b]} \Phi(x) = \Phi(x^*)$.

Метод касательных ориентирован на нахождение корня уравнения $\Phi'(x) = 0$, $x \in [a, b]$ в случае, когда на границах интервала $[a, b]$ знаки производной $\Phi'(x)$ различны. Такая ситуация, очевидно, возможна, если точка минимума функции $\Phi(x)$ является внутренней точкой интервала $[a, b]$ (см. рис. 2.6.1), метод требует, чтобы функция $\Phi(x)$ была определена и дважды дифференцируема в области допустимых значений $D = [a, b]$.

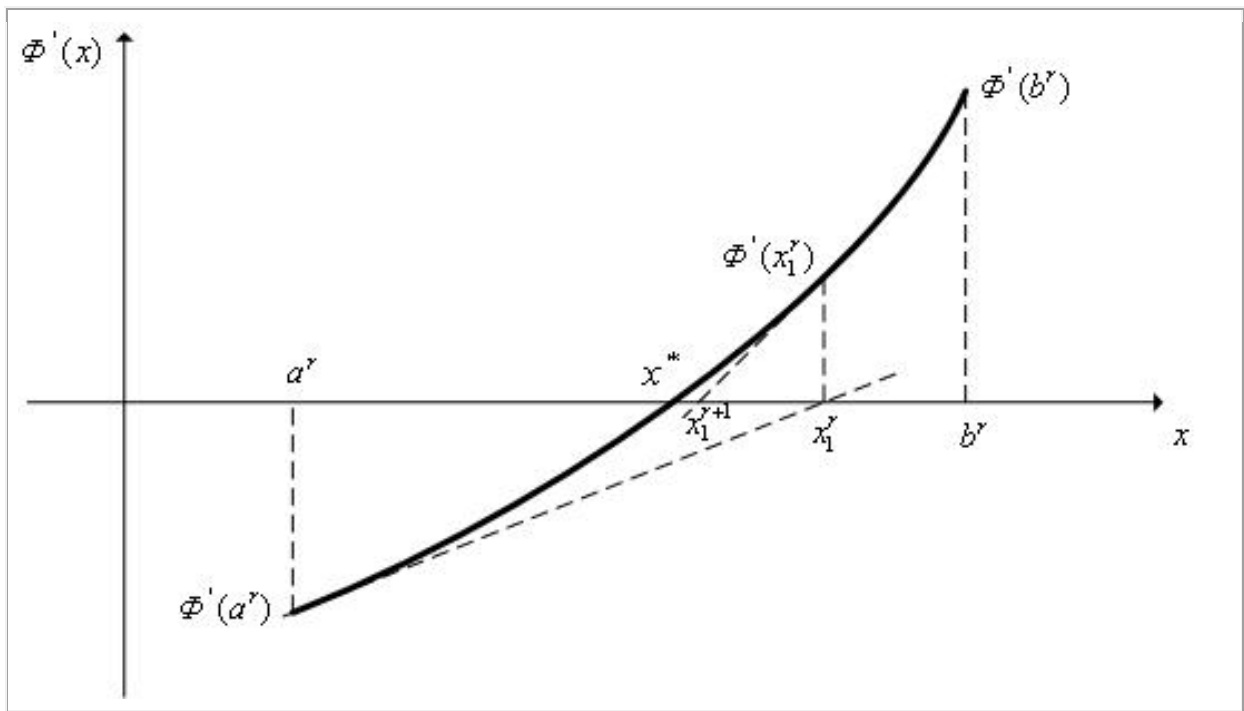


Рис. 2.6.1. К схеме метода касательных

Нам далее понадобится значение x_1^r . Линейная функция, аппроксимирующая функцию $\Phi'(x)$ в точке a^r , записывается в виде

$$\tilde{\Phi}'(x, a^r) = \Phi'(a^r) + \Phi''(a^r)(x - a^r). \quad (2.6.1)$$

Приравняв правую часть уравнения (2.6.1) к нулю, получим:

$$x_1^r = a^r - \frac{\Phi'(a^r)}{\Phi''(a^r)} \quad (2.6.2)$$

Алгоритм метода касательных

1. Выполняем присваивания $r = 1$, $a^1 = a$, $b^1 = b$.
2. Вычисляем значения производных $\Phi'(a^r)$, $\Phi'(b^r)$.
3. Если производные $\Phi'(a^r)$, $\Phi'(b^r)$ имеют одинаковые знаки – завершаем вычисления (точки a, b выбраны неверно).
4. По формуле (2.6.2.) вычисляем приближение x_1^r к стационарной точке функции $\Phi(x)$ и значение $\Phi'(x_1^r)$.
5. Если $|\Phi'(x_1^r)| \leq \varepsilon$, где ε – требуемая точность решения, то принимаем $x^* \approx x_1^r$ и заканчиваем вычисления.
6. Если разные знаки имеют производные $\Phi'(a^r)$, $\Phi'(x_1^r)$, (как на рис. 2.6.1) то выполняем присваивания $a^{r+1} = a^r$, $b^{r+1} = x_1^r$, $r = r + 1$ и переходим на п.4.

7. Если разные знаки имеют производные $\Phi'(x_1^r)$, $\Phi'(b^r)$, то выполняем присваивания $a^{r+1} = x_1^r$, $b^{r+1} = b^r$, $r = r + 1$ и переходим на п.4.

В случае квадратичной функции $\Phi(x)$ производная этой функции $\Phi'(x)$ линейна. Поэтому метод касательных гарантирует нахождение стационарной точки функции $\Phi(x)$ всего за одну итерацию.

Возможна модификация метода касательных, когда значения производной $\Phi'(x)$ вычисляются приближенно с использованием первых разностей. В этом случае метод становится прямым (нулевого порядка).

2.7. Метод перебора

Рассматривается следующая одномерная задача условной глобальной оптимизации): найти минимум, вообще говоря, многоэкстремальной функции $\Phi(x)$, определенной в замкнутой области допустимых значений $D = [a, b]$,

$$\min_{x \in [a, b]} \Phi(x) = \Phi(x^*).$$

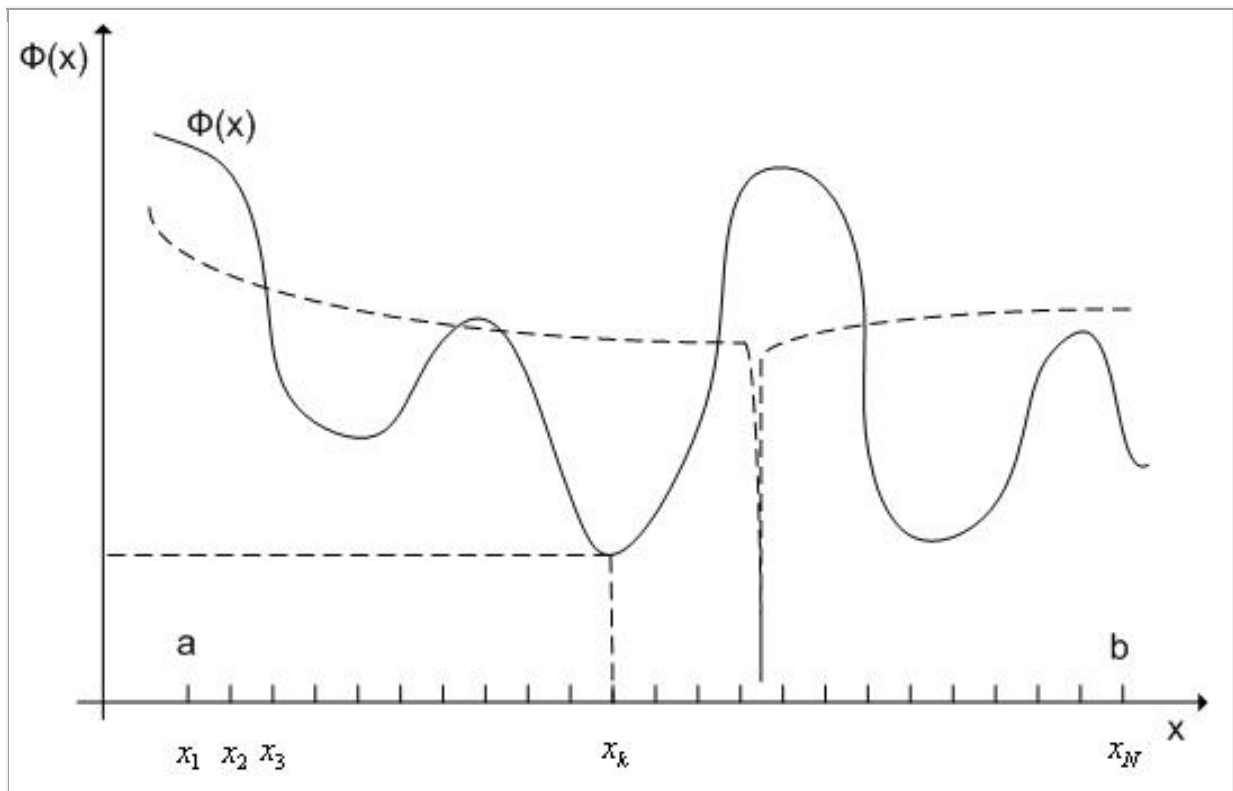


Рис. 2.7.1. К схеме метода перебора

Алгоритм метода перебора

1. Покрываем интервал $[a, b]$ некоторой сеткой с узлами $x_i, i \in [1, \dots, N]$.

2. Производим испытание в точке $x_1 = \tilde{x}^*$, т.е. вычисляем значения $\tilde{\Phi}^*$ функции $\Phi(x)$ в этой точке.
 3. Полагаем $r = 2$.
 4. Производим испытание в точке x_r – вычисляем значение $\Phi_r = \Phi(x_r)$ функции $\Phi(x)$ в этой точке.
 5. Если $\Phi_r < \tilde{\Phi}^*$, то выполняем присваивания $\tilde{x}^* = x_r$, $\tilde{\Phi}^* = \Phi_r$.
 6. Если $r < N$, то выполняем присваивание $r = r + 1$ и переходим на п.4. Иначе – заканчиваем вычисления.
 7. Принимаем \tilde{x}^* в качестве приближенного значения точки глобального минимума функции $\Phi(x)$ на интервале $[a, b]$ или каким-либо из рассмотренных одномерных методов локальной оптимизации организуем в окрестности точки \tilde{x}^* поиск локального минимума этой функции.
- При выборе количества узлов сетки $x_i, i \in [1, \dots, N]$ можно исходить из требуемой точности решения \mathcal{E}_x – максимальный шаг сетки принять равным этой величине.
- Отметим, что метод перебора, как и любой другой метод глобальной оптимизации, при отсутствии априорной информации о свойствах минимизируемой функции не гарантирует нахождение глобального минимума (см. пунктирный график на рис. 2.7.1).

2.8 Одномерный метод Монте-Карло

Рассматривается следующая одномерная задача условной глобальной оптимизации): найти минимум, вообще говоря, многоэкстремальной функции $\Phi(x)$, определенной в замкнутой области допустимых значений $D = [a, b]$,

$$\min_{x \in [a, b]} \Phi(x) = \Phi(x^*).$$

Алгоритм метода Монте-Карло

1. Генерируем с помощью программного генератора случайных чисел, равномерно распределенных в интервале $[a, b]$, случайное число $x_1 = \tilde{x}^*$.
2. Производим испытание в точке $x_1 = \tilde{x}^*$ – вычисляем значения $\tilde{\Phi}^*$ функции $\Phi(x)$ в этой точке.
3. Полагаем $r = 2$.
4. Аналогично п. 1 генерируем случайное число $x_r \in [a, b]$.

5. Производим испытание в точке x_r – вычисляем значение $\Phi_r = \Phi(x_r)$ функции $\Phi(x)$ в этой точке.
6. Если $\Phi_r < \tilde{\Phi}^*$, то выполняем присваивания $\tilde{x}^* = x_r$, $\tilde{\Phi}^* = \Phi_r$.
7. Если $r < N$, то выполняем присваивание $r = r + 1$ и переходим на п. 4. Иначе – заканчиваем вычисления. Здесь N – количество испытаний.
8. Принимаем \tilde{x}^* в качестве приближенного значения точки глобального минимума функции $\Phi(x)$ на интервале $[a, b]$ или каким-либо из рассмотренных одномерных методов локальной оптимизации организуем в окрестности точки \tilde{x}^* поиск локального минимума этой функции.
При достаточно большом N метода гарантирует нахождение глобального минимума с высокой вероятностью.

2.9 Метод выделения интервалов унимодальности

Рассмотрим одномерную задачу условной глобальной оптимизации: найти минимум одномерной многоэкстремальной функции $\Phi(x)$, определенной в **замкнутой** области допустимых значений $D = [a, b]$ и имеющей в этой области конечное число минимумов $\min_{x \in [a, b]} \Phi(x) = \Phi(x^*)$.

Метод выделения интервалов унимодальности функции $\Phi(x)$ требует априорного знания оценки $d > 0$ минимального расстояния между локальными минимумами этой функции.

Алгоритм метода выделения интервалов унимодальности

1. Полагаем $r = 1$.
2. Если функция $\Phi(x)$ в точке a возрастает, то полагаем $x_0^r = a$ переходим на п. 4.
3. Если функция $\Phi(x)$ в точке a убывает, то полагаем $x_1^r = a$ и переходим на п. 5.
4. Последовательно увеличивая x на величину Δ по формуле $x = x_0^r + i\Delta \leq b$, $i = 1, 2, \dots$, находим первую точку x_1^r , в которой функция $\Phi(x)$ убывает. Здесь и далее Δ – величина в несколько раз меньшая величины d .
5. Последовательно увеличивая x на величину Δ по формуле $x = x_1^r + i\Delta \leq b$, $i = 1, 2, \dots$, находим первую точку x_2^r , в которой функция $\Phi(x)$ возрастает.
6. В качестве r -го интервала унимодальности принимаем интервал $[x_1^r, x_2^r]$.

7. Если интервал $[a, b]$ исчерпан, переходим на п.8, иначе полагаем $x_0^{r+1} = x_2^r$, $r = r + 1$ и переходим на п.4.

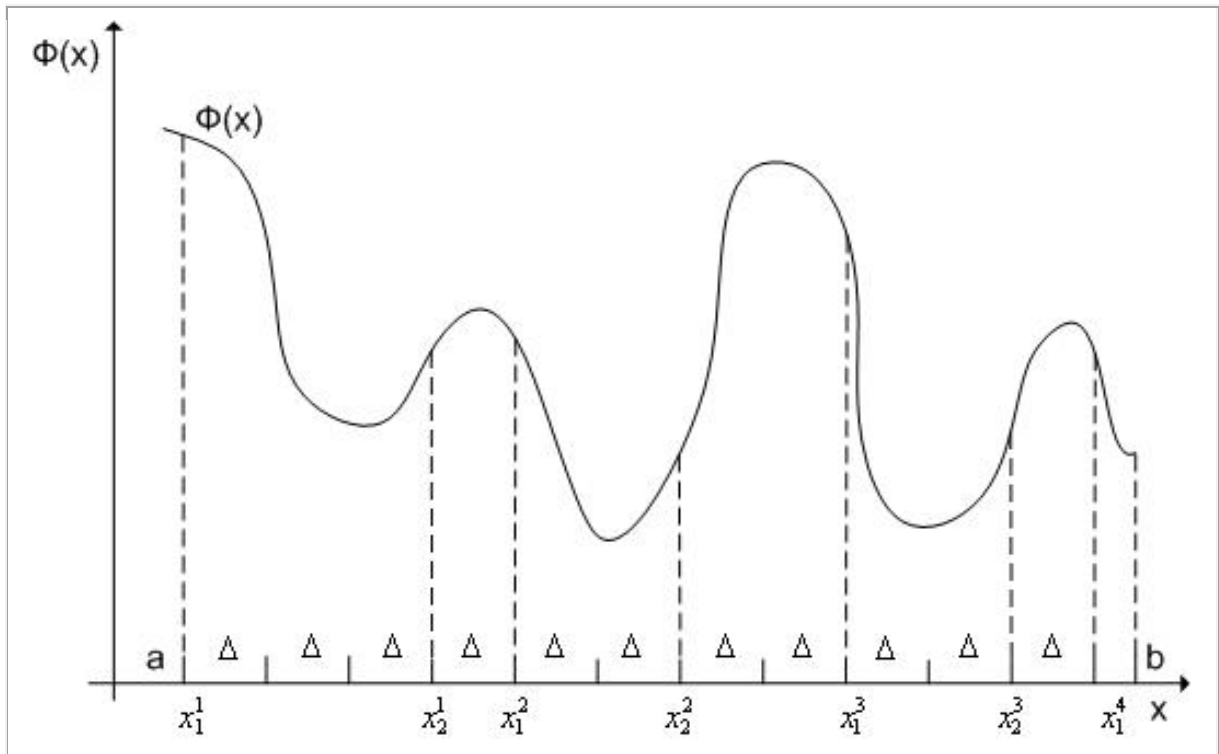


Рис. 2.9.1. К схеме выделения интервалов унимодальности

8. Положим, что общее количество найденных интервалов унимодальности функции $\Phi(x)$ равно $N-1$. Каким-либо одномерным методом локальной оптимизации находим локальный минимум функции $\Phi(x)$ в каждом из $N-1$ интервалов унимодальности. Обозначим точки этих минимумов $\tilde{x}_i^*, i \in [1, \dots, N-1]$. Соответствующие значения функции $\Phi(x)$ обозначим $\tilde{\Phi}_i^*, i \in [1, \dots, N-1]$. Добавим к точкам \tilde{x}_i^* точки $\tilde{x}_0^*, \tilde{x}_N^*$ и вычислим соответствующие значения $\tilde{\Phi}_0^*, \tilde{\Phi}_N^*$ функции $\Phi(x)$.
9. Найдем минимальную из величин $\tilde{\Phi}_i^*, i \in [1, \dots, N]$ и соответствующее значение аргумента: $\min_{i \in [0, N]} \tilde{\Phi}_i^* = \tilde{\Phi}_k^* = \Phi(\tilde{x}_k^*)$.
10. В качестве решения задачи глобальной оптимизации примем точку $(\tilde{x}_k^*, \tilde{\Phi}_k^*)$
- На рис. 2.9.1 интервалами унимодальности являются интервалы: $[x_1^1, x_2^1]$, $[x_1^2, x_2^2]$, $[x_1^3, x_2^3]$.

Для определения того, возрастает или убывает в данной точке функция $\Phi(x)$, может использоваться ее первая разность в этой точке $\Delta\Phi_x = \Phi(x) - \Phi(x + \Delta x)$, где Δx – некоторая малая величина. А именно, если $\Delta\Phi_x > 0$, то функция возрастает в точке x ; иначе – убывает. Заметим, что при этом в каждой точке требуется выполнить дополнительное испытание функции $\Phi(x)$.

Если функция $\Phi(x)$ непрерывно дифференцируема в интервале $[a, b]$, то для определения того, возрастает или убывает в данной точке эта функция, можно, очевидно, использовать значения первой производной функции $\Phi(x)$ в этой точке. А именно, если $\Phi'(x) > 0$, то в точке x функция $\Phi(x)$ возрастает; в противном случае – убывает.

Замечание 2.9.1. Если априорная оценка d минимального расстояния между локальными минимумами функции $\Phi(x)$ отсутствует, то никаких оснований полагать, что в интервалах, выделенных с помощью рассмотренного алгоритма, функция $\Phi(x)$ является унимодальной функцией.

Пусть, например, функция $\Phi(x)$ на интервале $[c, d]$ постоянна (см. рис. 2.9.2).

Если к такой функции применить алгоритм выделения интервалов унимодальности с любым $\Delta > 0$, то в качестве интервала унимодальности будет выделен интервал $[x_1^1, x_2^1] \supset [c, d]$, на котором функция $\Phi(x)$ имеет бесконечное количество минимумов, т.е. не является унимодальной функцией.

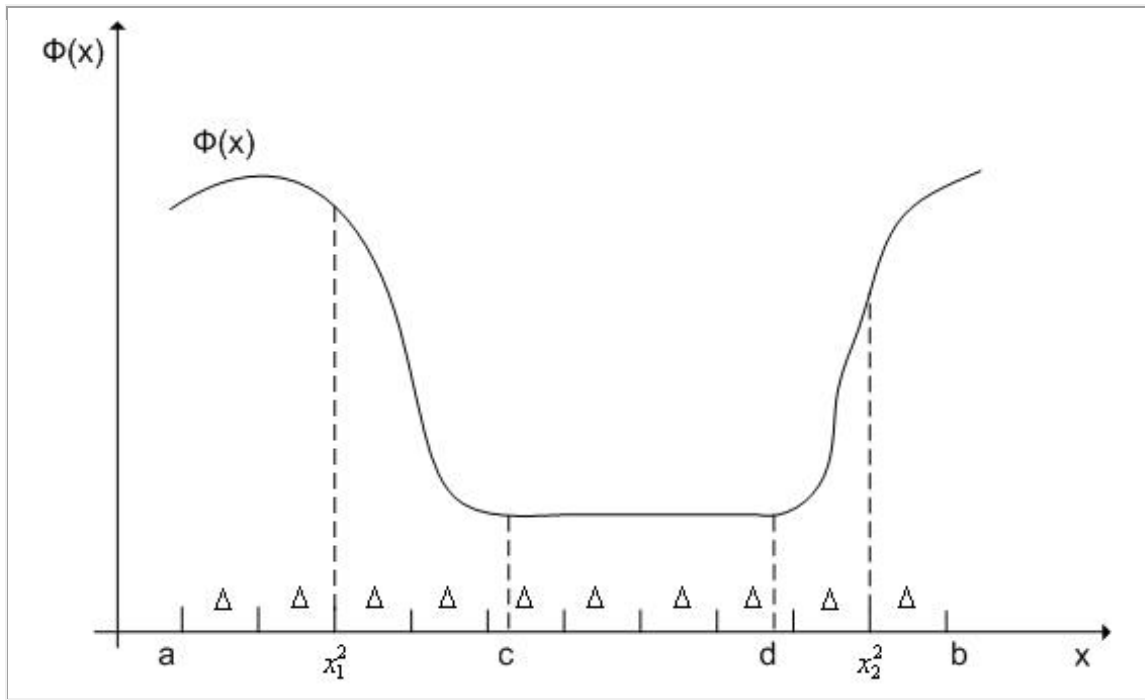


Рис.2.9.2.К замечанию 2.9.1

2.10 Метод аппроксимирующих моделей

Рассмотрим одномерную задачу условной глобальной оптимизации): найти минимум одномерной мультимодальной функции $\Phi(x)$, определенной в замкнутой области допустимых значений $D = [a, b]$, $\min_{x \in [a, b]} \Phi(x) = \Phi(x^*)$.

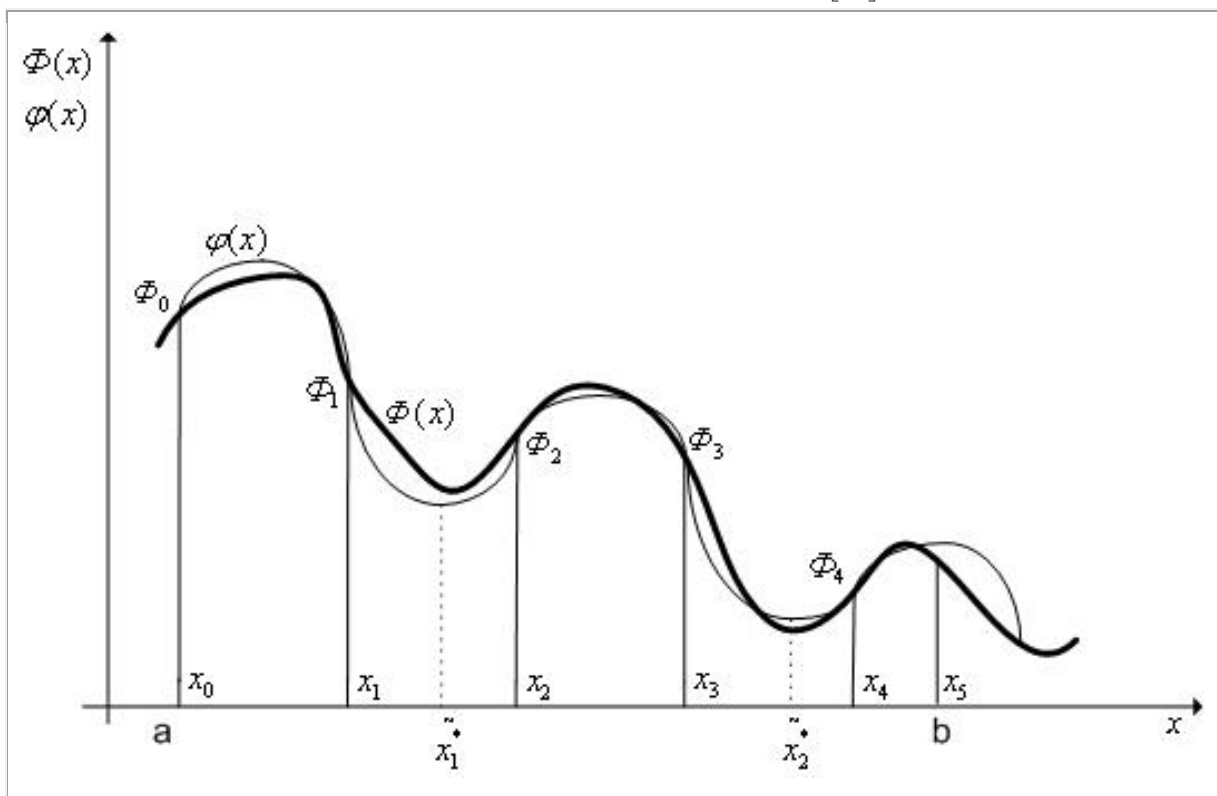


Рис. 2.10.1. К схеме метода аппроксимирующих моделей. $N = 5$

Алгоритм метода аппроксимирующих моделей

1. Покрываем интервал $[a, b]$ некоторой сеткой с узлами $x_i \in [a, b], x_i \neq x_j, i, j \in [0, \dots, N]$ и производим испытания в точках $x_i, i \in [0, \dots, N]$, т.е. вычисляем значения функции $\Phi(x)$ в этих точках $\Phi(x_i) = \Phi_i, i \in [0, \dots, N]$.
2. Строим аппроксимирующую функцию $\varphi(x)$, проходящую через точки $(x_i, \Phi_i), i \in [0, \dots, N]$. Эту функцию принято называть **математической моделью минимизируемой функции $\Phi(x)$** или модельной функцией.
3. Оцениваем адекватность построенной модели $\varphi(x)$. Для этого:
 - производим дополнительные испытания функции $\Phi(x)$ в некоторых точках $x_k \in [a, b], k \in [1, \dots, M]$;
 - вычисляем значения модельной функции $\varphi(x)$ и функции $\Phi(x)$ в этих точках $\varphi(x_k), \Phi(x_k), k \in [1, \dots, M]$;
 - вычисляем погрешность аппроксимации, например, $\max_{k \in [1, \dots, M]} |\varphi(x_k) - \Phi(x_k)|$.
4. Если погрешность аппроксимации превышает заданную, то по результатам всех предшествующих испытаний строим новую модельную функцию $\varphi(x)$ и переходим на п. 3.
5. Определяем положение глобального минимума модельной функции $\varphi(x)$, который или принимается в качестве глобального минимума функции $\Phi(x)$, или уточняется с помощью какого-либо метода локальной оптимизации.

На рис. 2.10.1 x_1^*, x_2^* – точки локального минимума модельной функции $\varphi(x)$; точка \tilde{x}_2^* – приближенное значение точки глобального минимума функции $\Phi(x)$ на интервале $[a, b]$.

В качестве модельных функций $\varphi(x)$ чаще всего используют полиномы.

Рассмотрим использование в качестве модельной функции полиномов.

Аппроксимирующий полином Лагранжа

Будем искать аппроксимирующий полином в виде

$$\varphi(x) = \Phi(x_0)\psi_0(x) + \Phi(x_1)\psi_1(x) + \dots + \Phi(x_N)\psi_N(x) = \sum_{i=0}^N \Phi(x_i)\psi_i(x),$$

где $\psi_i(x), i \in [0, \dots, N]$ – неизвестные полиномы от x , независимые от аппроксимируемой функции $\varphi(x)$.

Из того условия, что модельная функция $\varphi(x)$ должна совпадать с аппроксимируемой функцией $\Phi(x)$ в узлах сетки $x_i, i \in [0, \dots, N]$, имеем систему из $N+1$ равенств

$$\varphi(x_i) = \Phi(x_i) = \Phi(x_0)\psi_0(x_i) + \Phi(x_1)\psi_1(x_i) + \dots + \Phi(x_N)\psi_N(x_i), \quad i \in [0, \dots, N] \quad (2.10.1)$$

Для выполнения равенств (2.10.1) полиномы $\psi_i(x), i \in [0, \dots, N]$, очевидно, должны удовлетворять условиям:

$$\psi_i(x_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}, \quad (2.10.2)$$

или, другими словами, полином $\psi_i(x), i \in [0, \dots, N]$ должен иметь в качестве корней все числа $x_i, i \in [0, \dots, N]$, кроме числа x_j , а при $x = x_j$ должен иметь значение, равное единице.

Условию (2.10.2) удовлетворяют только полиномы вида

$$\psi_i(x) = A_i \cdot (x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_N),$$

где A – неизвестная константа.

Найдем эту константу из условия $\psi_i(x_i) = 1$:

$$1 = A_i \cdot (x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_N);$$

$$A_i = \frac{1}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_N)}.$$

Таким образом,

$$\psi_i(x) = \frac{(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_N)}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_N)} = \prod_{j=0, j \neq i}^N \frac{(x - x_j)}{(x_i - x_j)} \quad (2.10.3)$$

и искомый аппроксимирующий полином определяют выражением:

$$\varphi(x) = \sum_{i=0}^N \Phi(x_i) \psi_i(x) = L_N(x). \quad (2.10.4)$$

Полином (2.10.4) называется аппроксимирующим полиномом Лагранжа.

Использование аппроксимирующего полинома Лагранжа (2.10.4) в качестве модельной функции идейно очень просто, но обладает существенным недостатком. Пусть после построения этого полинома на сетке $x_i, i \in [0, \dots, N]$ и проверке его адекватности выясняется, что погрешность аппроксимации превышает заданную. Тогда, в соответствии с рассмотренной выше схемой метода, необходимо построить новый полином Лагранжа на сетке, полученной объединением сеток $x_i, i \in [0, \dots, N]$, $x_k, k \in [0, \dots, M]$, что требует пересчета всех посчитанных ранее функций $\psi_i(x), i \in [0, \dots, N]$. От этого недостатка свободна модификация аппроксимирующего полинома Лагранжа – аппроксимирующий полином Ньютона.

Нахождение стационарных точек аппроксимирующего полинома

После построения аппроксимирующего полинома возникает задача нахождения стационарных точек функции $\varphi(x)$ (см. схему метода).

Поскольку аппроксимирующий полином непрерывен и, по крайней мере, один раз непрерывно дифференцируем, его стационарные точки удобно искать как нули первой производной – т.е. как корни уравнения

$$\varphi'(x) = 0 \quad (2.10.5)$$

Для поиска корней уравнения чаще всего используют метод хорд и метод касательных, использующие линейную интерполяцию функции $\varphi'(x)$ (см. параграфы 2.5 и 2.6), а также другие методы.

Если положить $X_0^{r+1} = X^r$, $X_n^{r+1} = X^{r+1}$, то формулы (3.1.1), (3.1.2) можно записать в виде:

$$X_i^{r+1} = X_{i-1}^{r+1} + \lambda_i^r L_i, i = 1, 2, \dots, n, \quad (3.1.3)$$

$$\min_{\lambda \in (-\infty, +\infty)} \Phi(X_{i-1}^{r+1} + \lambda L_i) = \Phi(X_{i-1}^{r+1} + \lambda_i^r L_i) = \Phi(X_i^{r+1}), \quad i = 1, 2, \dots, n. \quad (3.1.4)$$

Таким образом, каждая итерация по методу Гаусса-Зейделя включает в себя n шагов. Каждая последующая итерация начинается из точки, полученной на последнем шаге предыдущей итерации. Поиск заканчивается при выполнении одного из стандартных условий окончания итераций:

$$\|X^{r+1} - X^r\| \leq \varepsilon_x, \quad (3.1.5)$$

$$|\Phi(X^{r+1}) - \Phi(X^r)| \leq \varepsilon_\Phi. \quad (3.1.6)$$

Заметим, что задачи (3.1.4) даже в случае одноэкстремальной функции $\Phi(X)$ могут быть задачами многоэкстремальной оптимизации и могут быть решены рассмотренными в разделе 2 методами решения задач одномерной оптимизации.

Схема метода Гаусса-Зейделя

1. Задаем начальную точку X^0 и полагаем $r = 0$, $i = 1$.
2. Последовательно для $i = 1, 2, \dots, n$ решаем задачи (3.1.4), т.е. исходя из предыдущей точки, отыскиваем минимум функции $\Phi(X)$ вдоль i -го координатного направления.
3. Если условие окончания поиска (3.1.5) или (3.1.6) выполнено, то полагаем $X^* \approx X^{r+1}$ и заканчиваем вычисления. Иначе – полагаем $r = r + 1$ и переходим к п. 2.

Метод Гаусса-Зейделя иллюстрирует рис. 3.1.1, на котором показан фрагмент линий уровня функции Химмельблау $\Phi(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$. На рис. 3.1.1 точка X_1^1 представляет собой локальный минимум функции $\Phi(X)$ вдоль оси X_1 при исходной точке X^0 . Точка X_2^1 представляет собой локальный минимум функции $\Phi(X)$ вдоль оси

X_2 при исходной точке X_1^1 . Отыскание точки X_2^1 завершает первую итерацию. Следующая итерация начинается из точки $X^1 = X_2^1$. И т.д.

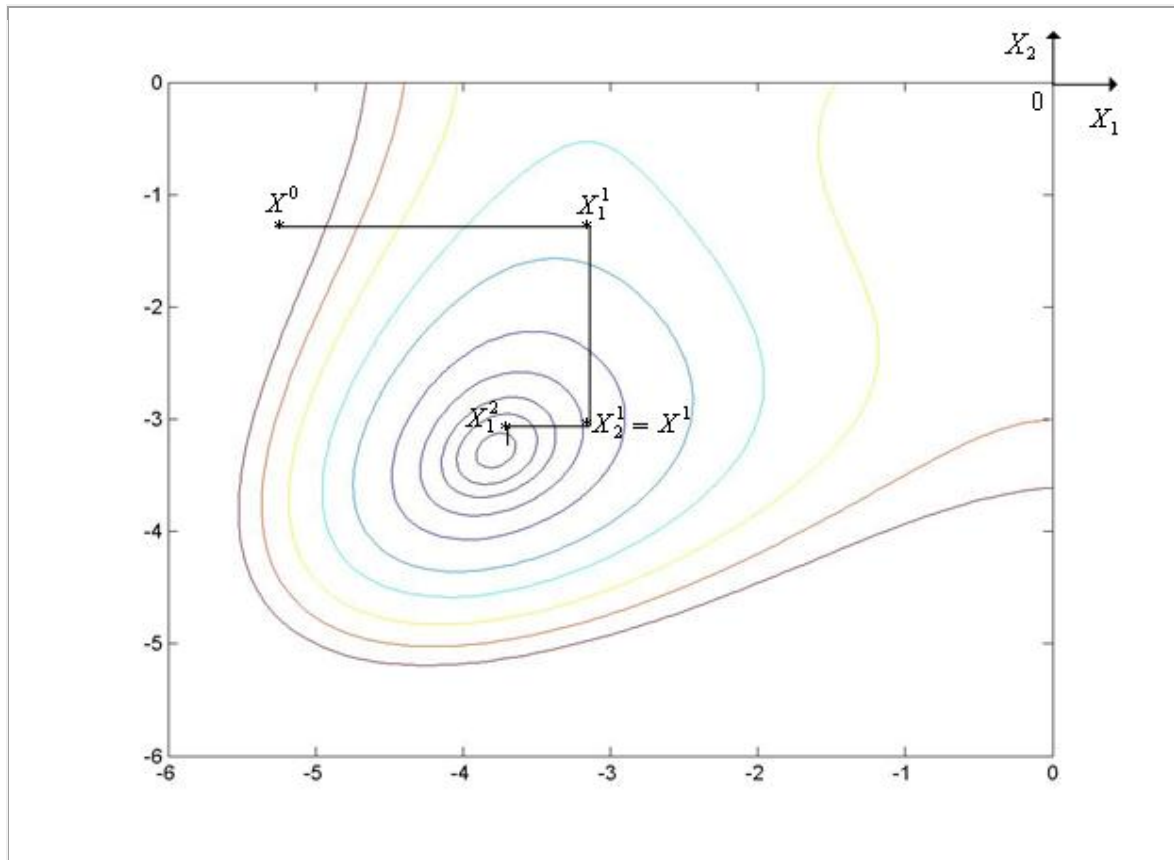


Рис. 3.1.1. Траектория поиска минимума не овражной функции Химмельблау методом Гаусса-Зейделя

Метод Гаусса-Зейделя медленно сходится на овражных функциях, в которых овраг не ориентирован в направлении какой-либо из координатных осей (см. рис. 3.1.2). На рисунке показаны линии уровня функции Розенброка ($n = 2$).

На рисунке 3.1.2. показывается, что текущая точка быстро (в данном случае – за один шаг) «скатывается» на дно оврага и очень медленно движется по дну оврага к минимуму функции $\Phi(X)$.

Пример 3.1.1. Выполните несколько итераций решения двумерной локальной задачи безусловной оптимизации $\min_{X \in R^2} \Phi(X) = \Phi(X^*) = \Phi^*$, где

$\Phi(X) = \Phi(x, y) = x^2 + y^2 + 3(x + y)^2 = 4x^2 + 4y^2 + 6xy$ методом Гаусса-Зейделя, исходя из точки $X^0 = (x^0, y^0) = (-1.5, 1.5)$.

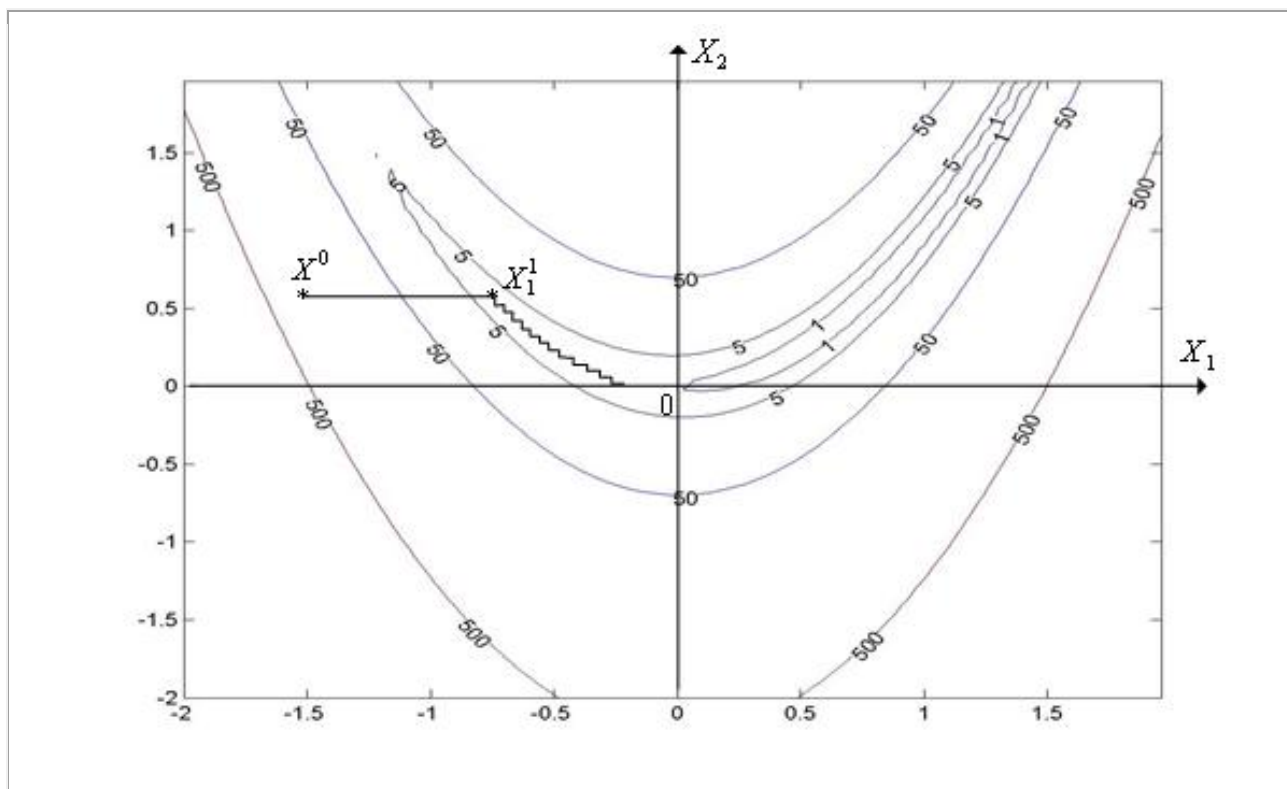


Рис. 3.1.2. Траектория поиска минимума овражной функции Розенброка методом Гаусса-Зейделя. Текущая точка быстро (в данном случае – за один шаг) «скатывается» на дно оврага и очень медленно движется по дну оврага к минимуму функции $\Phi(X)$

Решение. Каждая итерация метода Гаусса-Зейделя состоит из двух шагов и имеет вид

$$X_1^{r+1} = \begin{pmatrix} x^{r+1} = x^r + \lambda_1^r \\ y^r \end{pmatrix}, \quad (3.1.7)$$

$$X_2^{r+1} = X^{r+1} = \begin{pmatrix} x^{r+1} \\ y^{r+1} = y^r + \lambda_2^r \end{pmatrix}, \quad (3.1.8)$$

где величины λ_1^r, λ_2^r – определяются из условий:

$$\min_{\lambda \in (-\infty, +\infty)} \Phi(x^r + \lambda, y^r) = \Phi(x^r + \lambda_1^r, y^r) = \Phi(x^{r+1}, y^r) = \Phi(X_1^{r+1}), \quad (3.1.9)$$

$$\min_{\lambda \in (-\infty, +\infty)} \Phi(x^{r+1}, y^r + \lambda) = \Phi(x^{r+1}, y^r + \lambda_2^r) = \Phi(x^{r+1}, y^{r+1}) = \Phi(X_2^{r+1}) = \Phi(X^{r+1}). \quad (3.1.10)$$

Найдем явное решение задачи (3.1.9). Из (3.1.9) имеем

$$\begin{aligned}\Phi(x^r + \lambda, y^r) &= 4(x^r + \lambda)^2 + 4(y^r)^2 + 6(x^r + \lambda)y^r = \\ &= 4(x^r)^2 + 8x^r\lambda + 4\lambda^2 + 4(y^r)^2 + 6x^ry^r + 6\lambda y^r = 4\lambda^2 + (8x^r + 6y^r)\lambda + \dots\end{aligned}\quad (3.1.11)$$

Функция (3.1.11) относительно λ является квадратичной функцией с положительным коэффициентом при λ^2 и достигает минимума в точке, удовлетворяющей условию:

$$\Phi'_\lambda(x^r + \lambda, y^r) = 8\lambda + 8x^r + 6y^r = 0.$$

из которого имеем

$$\lambda_1^r = -x^r - \frac{3}{4}y^r. \quad (3.1.12)$$

Аналогично явное решение задачи (3.1.10) равно

$$\lambda_2^r = -y^r - \frac{3}{4}x^{r+1}. \quad (3.1.13)$$

Таким образом, из (3.1.7), (3.1.8), (3.1.10), (3.1.11) имеем искомую итерационную формулу метода Гаусса-Зейделя для нашей задачи:

$$X_1^{r+1} = \begin{pmatrix} x^{r+1} = x^r + \lambda_1^r \\ y^r \end{pmatrix}, \quad \lambda_1^r = -x^r - \frac{3}{4}y^r; \quad (3.1.14)$$

$$X_2^{r+1} = X^{r+1} = \begin{pmatrix} x^{r+1} \\ y^{r+1} = y^r + \lambda_2^r \end{pmatrix}, \quad \lambda_2^r = -y^r - \frac{3}{4}x^{r+1}. \quad (3.1.15)$$

Первая итерация ($r=1$).

Из формул (3.1.14) имеем $\lambda_1^1 = -x^0 - \frac{3}{4}y^0 = 1.5 - \frac{3}{4} \cdot 1.5 = \frac{3}{8} \approx 0.375$ и $x^1 = x^0 + \lambda_1^1 \approx -1.5 + 0.38 = -1.12$. Аналогично из формул (3.1.15) имеем $\lambda_2^1 = -y^0 - \frac{3}{4}x^1 \approx -1.5 - \frac{3}{4} \cdot 1.12 \approx -0.66$, $y^1 = y^0 + \lambda_2^1 \approx 1.5 - 0.66 = 0.84$.

Таким образом, $X^1 = \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} \approx \begin{pmatrix} -1.12 \\ 0.84 \end{pmatrix}$ (см. рис. 3.1.3).

Вторая итерация ($r=2$).

Аналогично первой итерации, имеем

$$\lambda_1^2 = -x^1 - \frac{3}{4}y^1 = 1.12 - \frac{3}{4} \cdot 0.84 \approx 0.49, \quad x^2 = x^1 + \lambda_1^2 \approx -1.12 + 0.49 = -0.63,$$

$$\lambda_2^2 = -y^1 - \frac{3}{4}x^2 \approx -0.84 + \frac{3}{4} \cdot 0.63 \approx -0.37, \quad y^2 = y^1 + \lambda_2^2 \approx 0.84 - 0.37 = 0.47.$$

Таким образом, $X^1 = \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} \approx \begin{pmatrix} -0.63 \\ 0.47 \end{pmatrix}$ (см. рис. 3.1.3).

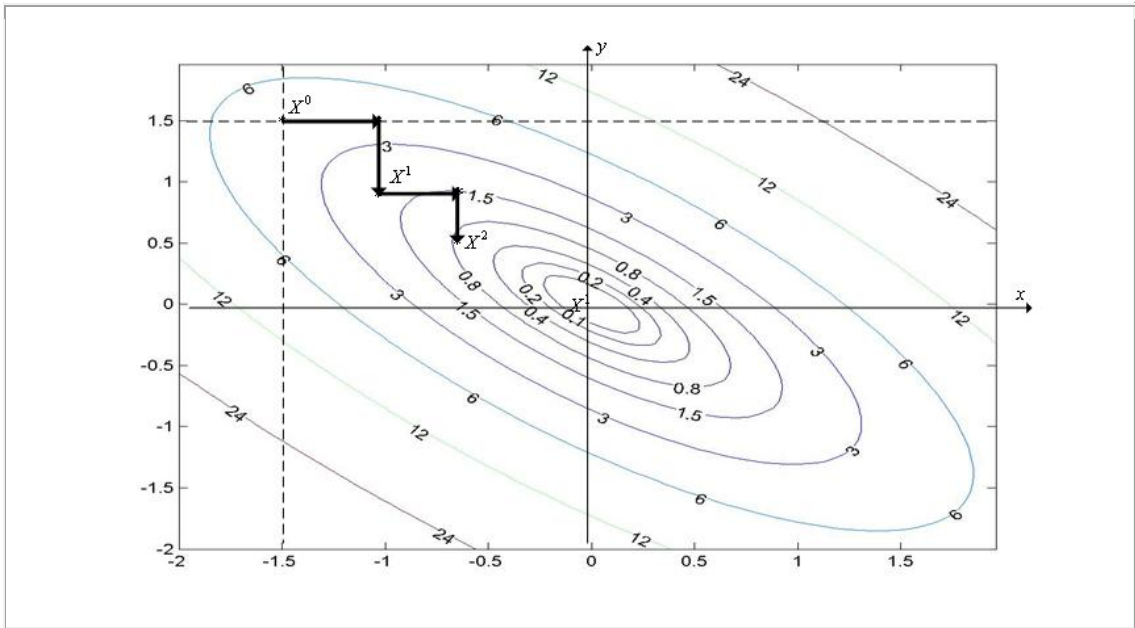


Рис. 3.1.3. Фрагмент (две итерации) траектории поиска минимума функции $\Phi(X) = \Phi(x, y) = x^2 + y^2 + 3(x + y)^2 = 4x^2 + 4y^2 + 6xy$ методом Гаусса-Зейделя из точки $X^0 = (x^0, y^0) = (-1.5, 1.5)$

3.2 Метод Хука-Дживса

Рассмотрим следующую многомерную задачу локальной безусловной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного в n -мерном евклидовом пространстве R^n , $\min_{X \in R^n} \Phi(X) = \Phi(X^*) = \Phi^*$.

При решении задачи методом Хука-Дживса (методом конфигураций, методом пробных шагов) используются итерационные формулы, аналогичные формулам, используемым в методе Гаусса-Зейделя

$$\tilde{X}_i^{r+1} = \tilde{X}_{i-1}^{r+1} + \lambda_i^r L_i, \quad i=1,2,\dots,n, \quad (3.2.1)$$

где принято $\tilde{X}_0^{r+1} = X^r$, $\tilde{X}_n^{r+1} = \tilde{X}^{r+1}$, вектор L_i определяет направление вдоль i -й координатной оси и представляет собой n -мерный вектор с компонентами

$$l_{ij} = \begin{cases} 1, & \text{если } j=i \\ 0, & \text{если } j \neq i, \quad j=1,2,\dots,n, \end{cases}$$

величины $\lambda_1^r, \lambda_2^r, \dots, \lambda_n^r$ – определяются из условий:

$$\lambda_n^r = \begin{cases} \Delta_i^r, & \text{если } \Phi(X_{i-1}^{r+1} + \Delta_i^r L_i) < \Phi(X_{i-1}^{r+1}), \\ -\Delta_i^r, & \text{если } \Phi(X_{i-1}^{r+1} - \Delta_i^r L_i) < \Phi(X_{i-1}^{r+1}) < \Phi(X_{i-1}^{r+1} + \Delta_i^r L_i), \\ 0, & \text{если } \min[\Phi(X_{i-1}^{r+1} + \Delta_i^r L_i), \Phi(X_{i-1}^{r+1} - \Delta_i^r L_i)] > \Phi(X_{i-1}^{r+1}). \end{cases} \quad (3.2.2)$$

После завершения n шагов выполняется спуск в направлении вектора $(\tilde{X}^{r+1} - X^r)$ по формуле:

$$X^{r+1} = X^r + \alpha^r (\tilde{X}^{r+1} - X^r), \quad (3.2.3)$$

где α^r – ускоряющий множитель. В различных модификациях метода Хука-Дживса множитель α^r может

- приниматься постоянным (обычно, равным 2),
- выбираться из условия $\Phi(X^{r+1}) < \Phi(X^r)$,
- находиться из условия локального минимума функции $\Phi(X)$ при движении из точки X^r в направлении вектора $(\tilde{X}^{r+1} - X^r)$:

$$\min_{\alpha \in (-\infty, +\infty)} \Phi(X^r + \alpha(\tilde{X}^{r+1} - X^r)) = \Phi(X^r + \alpha^r(\tilde{X}^{r+1} - X^r)) = \Phi(X^{r+1}). \quad (3.2.4)$$

Заметим, что задачи (3.2.4) даже в случае одноэкстремальной функции $\Phi(X)$ могут быть многоэкстремальными задачами оптимизации и могут быть решены рассмотренными в разделе 2 методами решения задач одномерной оптимизации.

Итерации заканчиваются при выполнении одного из стандартных условий окончания итераций:

$$\|X^{r+1} - X^r\| \leq \varepsilon_x, \quad (3.2.5)$$

$$|\Phi(X^{r+1}) - \Phi(X^r)| \leq \varepsilon_\Phi. \quad (3.2.6)$$

Вектор $\Delta^0 = (\Delta_1^0, \Delta_2^0, \dots, \Delta_n^0)$ является вектором свободных параметров метода – вектором «пробных шагов» по всем n координатным осям.

Известна модификация метода Хука-Дживса, в которой точка \tilde{X}^{r+1} определяется не процедурами (3.2.1), (3.2.2), а методом Гаусса-Зейделя.

Схема метода Хука-Дживса

1. Задаем начальную точку X^0 , вектор «пробных» шагов $\Delta^0 = (\Delta_1^0, \Delta_2^0, \dots, \Delta_n^0)$ и полагаем $r = 0$.
2. Последовательно для $i = 1, 2, \dots, n$ по формулам (3.2.1), (3.2.2) находим точки $\tilde{X}_1^{r+1}, \tilde{X}_2^{r+1}, \dots, \tilde{X}_n^{r+1} = \tilde{X}^{r+1}$.
3. Если $\tilde{X}^{r+1} \neq \tilde{X}^r$, то переходим к п. 4). Иначе уменьшаем длины «пробных» шагов $\Delta_1^0, \Delta_2^0, \dots, \Delta_n^0$, например, вдвое и переходим к п.2).
4. Если условие окончания поиска (3.2.5) или (3.2.6) выполнено, то полагаем $X^* \approx X^{r+1}$ и заканчиваем вычисления. Иначе выполняем спуск в направлении вектора $(\tilde{X}^{r+1} - X^r)$ по формуле (3.2.3), в которой ускоряющий множитель находится, например, из условия (3.2.4). Полагаем $r = r + 1$ и переходим к п. 2).

Метод Хука-Дживса иллюстрирует рис. 3.2.1, на котором показаны линии уровня функции Химмельблау ($n = 2$). Ускоряющий множитель α^r находится из условия локального минимума функции $\Phi(X)$ при движении из точки X^r в направлении вектора $(\tilde{X}^{r+1} - X^r)$.

Метод Хука-Дживса имеет высокую эффективность в случае, если функция $\Phi(X)$ имеет прямолинейный овраг (не обязательно ориентированный вдоль одного из координатных направлений, как в методе Гаусса-Зейделя). При минимизации "овражных" функций, имеющих не прямолинейный овраг, процесс поиска может сильно замедлиться и закончиться далеко от точки истинного минимума (см. рис. 3.2.2). На рисунке показаны линии уровня функции Розенброка ($n = 2$).

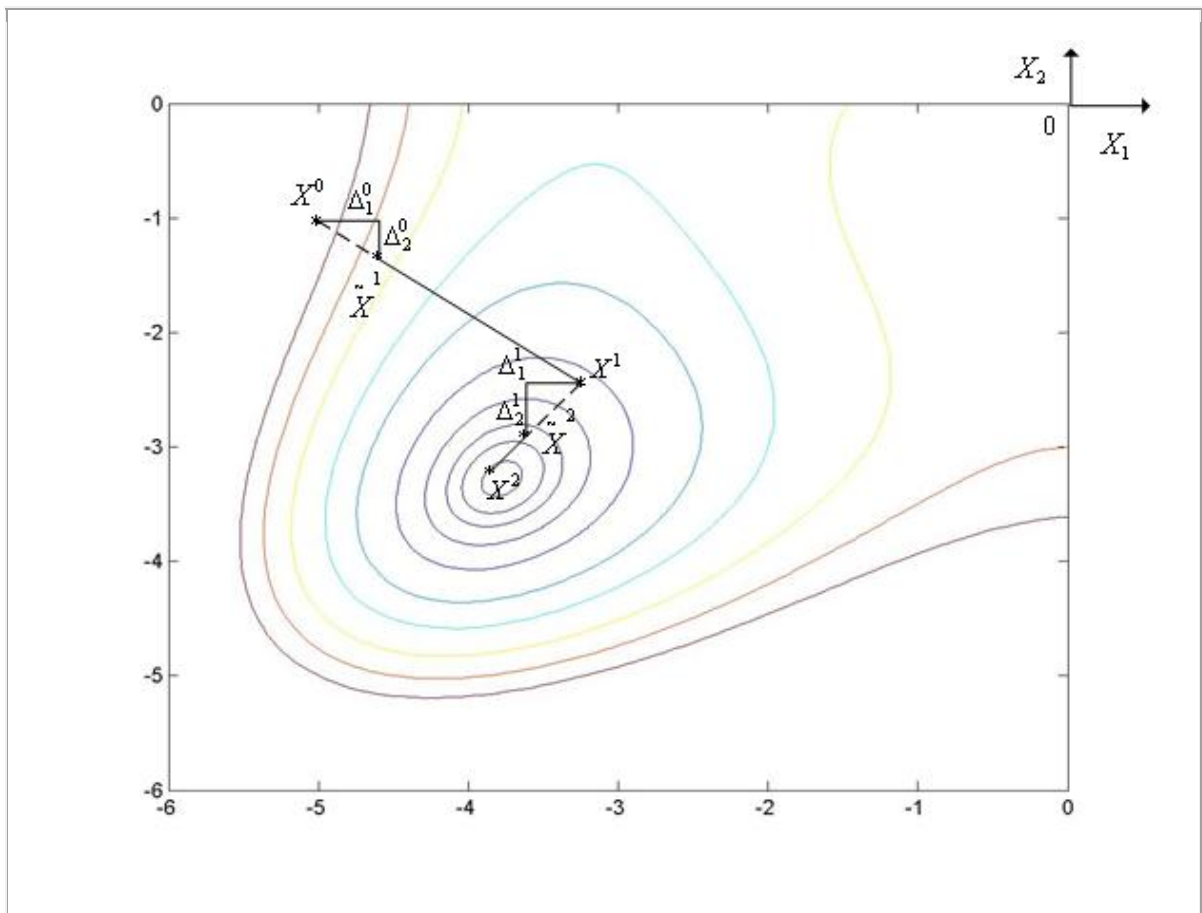


Рис. 3.2.1. Траектория поиска минимума не овражной функции Химмельблау методом Хука-Дживса

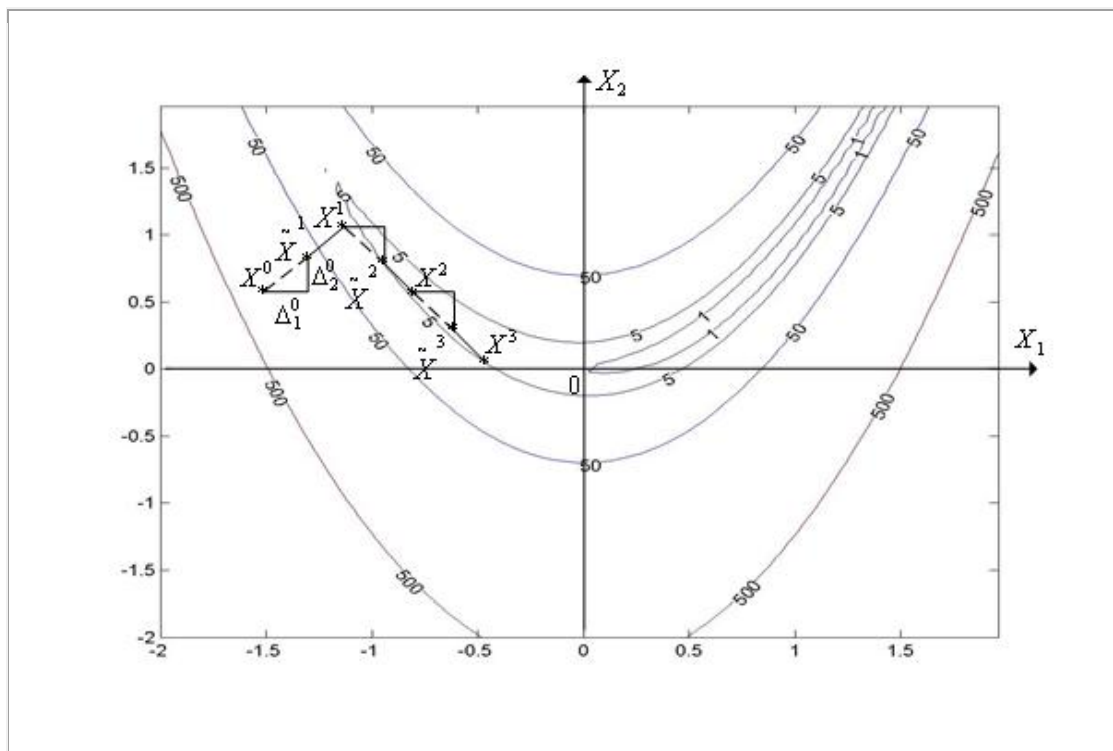


Рис. 3.2.2. Траектория поиска минимума овражной функции Химмельблау методом Хука-Дживса. Ускоряющий множитель $\alpha^r = 2$

3.3 Метод Розенброка

Рассмотрим следующую задачу локальной безусловной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного в n -мерном евклидовом пространстве R^n , $\min_{X \in R^n} \Phi(X) = \Phi(X^*) = \Phi^*$.

При решении задачи методом Розенброка (методом вращающихся координат) используется преобразование на каждой итерации системы координат таким образом, чтобы в новой системе координат одна из осей совпадала с направлением предыдущего шага. Остальные оси новой системы координат обычно находят с помощью процедуры ортогонализации Грамма-Шмидта.

Ортогонализация Грамма-Шмидта

Рассмотрим произвольный набор векторов p_1, p_2, \dots, p_n пространства R^n .

Поставим задачу построить на основе этих векторов ортонормированный набор векторов e_1, e_2, \dots, e_n того же пространства R^n .

Напомним, что набор векторов e_1, e_2, \dots, e_n называется ортонормированным, если для любых двух векторов из этого набора выполняется условие:

$$(e_i, e_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j. \end{cases} \quad (3.3.1)$$

Или, другими словами, набор векторов e_1, e_2, \dots, e_n ортонормирован, если эти векторы линейно независимы и скалярное произведение любых двух из них равно единице.

Для построения векторов e_1, e_2, \dots, e_n применим индуктивный подход. Положим, что

$$e_1 = \frac{p_1}{\rho_{11}}, \quad \rho_{11} = \|p_1\|, \quad (3.3.2)$$

где $\|*\|$ – символ евклидовой нормы. Полагая векторы e_1, e_2, \dots, e_{k-1} уже построенными, будем искать вектор e_k в виде

$$\rho_{kk} e_k = p_k - \sum_{i=1}^{k-1} \rho_{ik} e_i \quad (3.3.3)$$

Для отыскания неизвестных множителей ρ_{ik} умножим (3.3.3) скалярно на вектор e_j , $j = 1, 2, \dots, k-1$:

$$\rho_{kk}(e_k, e_1) = 0 = (p_k, e_1) - \sum_{i=1}^{k-1} \rho_{ik}(e_i, e_1) = (p_k, e_1) - \rho_{1k}(e_1, e_1),$$

$$\rho_{kk}(e_k, e_2) = 0 = (p_k, e_2) - \sum_{i=1}^{k-1} \rho_{ik}(e_i, e_2) = (p_k, e_2) - \rho_{2k}(e_2, e_2),$$

.....

$$\rho_{kk}(e_k, e_{k-1}) = 0 = (p_k, e_{k-1}) - \sum_{i=1}^{k-1} \rho_{ik}(e_i, e_{k-1}) = (p_k, e_{k-1}) - \rho_{k-1,k}(e_{k-1}, e_{k-1}).$$

Поскольку $(e_i, e_j) = 1$, имеем

$$\rho_{jk}(e_k, e_j) = (p_k, e_j), \quad j \in [1, \dots, k-1]. \quad (3.3.4)$$

Множитель ρ_{kk} найдем из условия $\|e_k\| = 1$:

$$\rho_{kk} = \left\| p_k - \sum_{i=1}^{k-1} \rho_{ik} e_i \right\|. \quad (3.3.5)$$

Определение 3.3.1. Процесс перехода от векторов p_1, p_2, \dots, p_n к векторам q_1, q_2, \dots, q_n согласно формулам (3.3.2)–(3.3.5) называется **ортогонализацией Грамма-Шмидта**.

Каждая итерация метода Розенброка состоит из двух этапов. В зависимости от модификации метода первый этап может выполняться с использованием различных методов. Рассмотрим применение на первом этапе итерационной формулы метода Гаусса-Зейделя. Приведем формулировку этой формулы, несколько отличную от формулировки, рассмотренной в параграфе 3.1.

Положим $X_0^{r+1} = X^r$, $X_n^{r+1} = X^{r+1}$ и пусть $e_1^r, e_2^r, \dots, e_n^r$ – орты системы координат, используемой на r -ой итерации. Тогда итерационную формулу метода Гаусса-Зейделя можно записать в виде

$$X_i^{r+1} = X_{i-1}^{r+1} + \lambda_i^r e_i^r = X_{i-1}^{r+1} + q_i^r, \quad i = 1, 2, \dots, n, \quad (3.3.6)$$

где коэффициенты λ_i^r находятся из условий:

$$\min_{\lambda \in (-\infty, +\infty)} \Phi(X_{i-1}^{r+1} + \lambda e_i^r) = \Phi(X_{i-1}^{r+1} + \lambda_i^r e_i^r) = \Phi(X_i^{r+1}), \quad i = 1, 2, \dots, n. \quad (3.3.7)$$

На втором этапе каждой из итераций система векторов $e_1^r, e_2^r, \dots, e_n^r$ с использованием ортогонализации Грамма-Шмидта заменяется новой системой линейно независимых векторов $e_1^{r+1}, e_2^{r+1}, \dots, e_n^{r+1}$.

Схема метода Розенброка

1. Задаем начальную точку X^0 , полагаем $r=0$, $i=1$, и орты исходной системы координат обозначаем $e_1^0, e_2^0, \dots, e_n^0$.
2. Исходя из точки X^r по формулам (3.3.6), (3.3.7) выполняем одну итерацию по методу Гаусса-Зейделя – получаем точку X^{r+1} и совокупность векторов $q_1^r, q_2^r, \dots, q_n^r$.
3. Если одно из стандартных условий окончания итераций

$$\|X^{r+1} - X^r\| \leq \varepsilon_x, \quad (3.3.8)$$

$$|\Phi(X^{r+1}) - \Phi(X^r)| \leq \varepsilon_\phi \quad (3.3.9)$$

выполнено, то полагаем $X^* \approx X^{r+1}$, и заканчиваем вычисления. Иначе переходим к п.4.

4. На основе векторов $q_1^r, q_2^r, \dots, q_n^r$ находим векторы $p_1^r, p_2^r, \dots, p_n^r$:

$$p_i^r = \sum_{j=1}^n q_j^r, \quad i = 1, 2, \dots, n. \quad (3.3.10)$$

5. С помощью процедуры ортогонализации Грамма-Шмидта (3.3.2) – (3.3.5) выполняем переход от системы векторов $p_1^r, p_2^r, \dots, p_n^r$ к системе векторов $e_1^{r+1}, e_2^{r+1}, \dots, e_n^{r+1}$, полагаем $r = r+1$ и переходим к п. 2. Заметим, что из формулы (3.3.10) следует равенство $p_n^r = q_n^r$.

По сравнению с методом Гаусса-Зейделя и методом Хука-Дживса метод Розенброка имеет, как правило, более высокую эффективность на овражных функциях с не прямолинейным оврагом.

Метод Розенброка иллюстрирует рис. 3.3.1, на котором показаны линии уровня функции Химмельблау ($n = 2$),

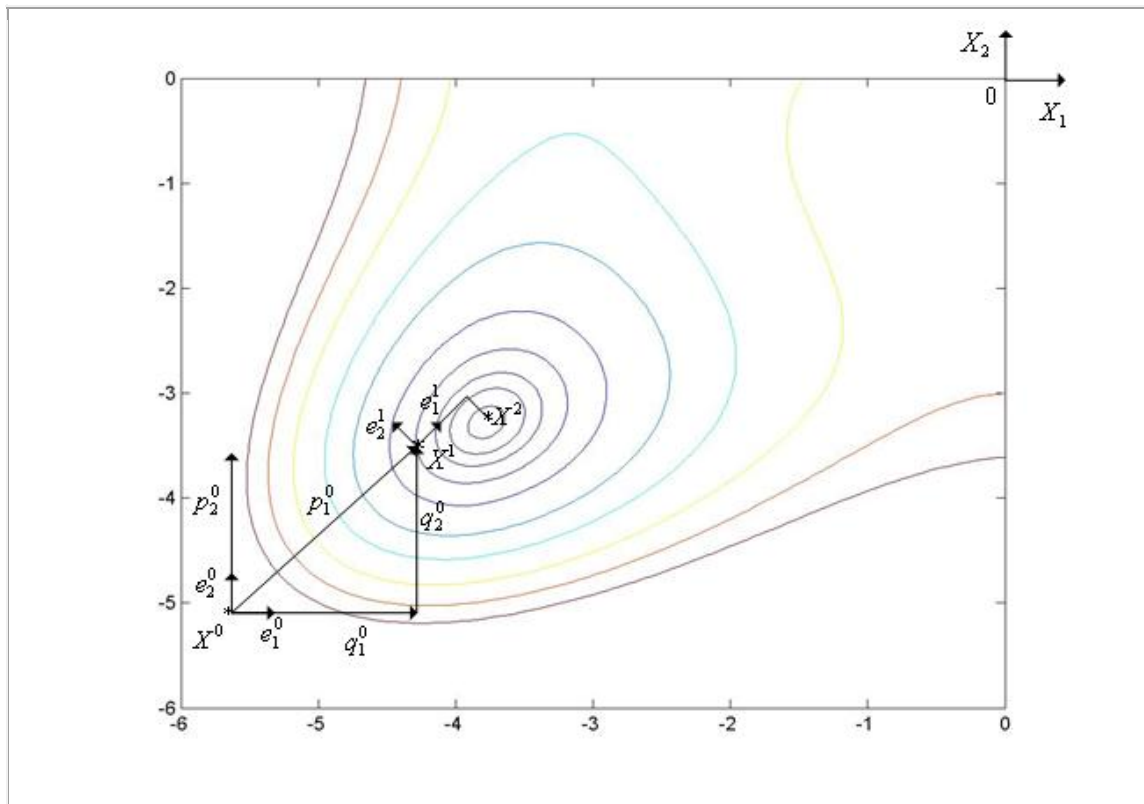


Рис. 3.3.1. Траектория поиска минимума функции Химмельблау методом Розенброка

3.4 Метод сопряженных направлений

Рассмотрим следующую задачу локальной безусловной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного в n -мерном евклидовом пространстве R^n , $\min_{X \in R^n} \Phi(X) = \Phi(X^*) = \Phi^*$.

Введем прежде следующие понятия: векторы p_1, p_2, \dots, p_k , принадлежащие пространству R^n , называются векторами сопряженными относительно матрицы $A(n \times n)$, если $(Ap_i p_j) = 0$ для всех $i \neq j, i, j \in [1, \dots, k]$.

В методе сопряженных направлений применяется итерационная формула метода Гаусса-Зейделя в виде, близком к использованному в параграфе 3.3.

Положим $X_0^{r+1} = X^r$ и пусть e_1, e_2, \dots, e_n – орты используемой системы координат.

Тогда итерационную формулу метода Гаусса-Зейделя можно записать в виде

$$X_i^{r+1} = X_{i-1}^{r+1} + \lambda_i^r e_i = X_{i-1}^{r+1} + p_i^r, i=1,2,\dots,n, \quad (3.4.1)$$

где коэффициенты λ_i^r находятся из условий:

$$\min_{\lambda \in (-\infty, +\infty)} \Phi(X_{i-1}^{r+1} + \lambda e_i) = \Phi(X_{i-1}^{r+1} + \lambda_i^r e_i) = \Phi(X_i^{r+1}), \quad i=1,2,\dots,n. \quad (3.4.2)$$

Схема метода сопряженных направлений

1. Задаем начальную точку X^0 и полагаем $r=0, i=1$.
2. Последовательно для $i=1,2,\dots,n$ по формулам (3.4.1), (3.4.2) находим точки $X_1^{r+1}, X_2^{r+1}, \dots, X_n^{r+1}$.
3. Исходя из точки X_n^{r+1} , еще раз находим минимум функции $\Phi(X)$ вдоль первого координатного направления – вычисляем координаты точки

$$X_{n+1}^{r+1} = X_n^{r+1} + \tilde{\lambda}_1^r e_1, \quad (3.4.3)$$

где коэффициент $\tilde{\lambda}_1^r$ находится из условия:

$$\min_{\lambda \in (-\infty, +\infty)} \Phi(X_{n+1}^{r+1} + \lambda e_1) = \Phi(X_n^{r+1} + \tilde{\lambda}_1^r e_1) = \Phi(X_{n+1}^{r+1}). \quad (3.4.4)$$

4. Исходя из точки X_1^{r+1} , находим минимум функции $\Phi(X)$ вдоль вектора $p_{n+1}^r = X_{n+1}^{r+1} - X_1^{r+1}$, вычисляем

$$X^{r+1} = X_1^r + \lambda_{n+1}^{r+1} p_{n+1}^r, \quad (3.4.5)$$

где коэффициент λ_{n+1}^r находится из условия:

$$\min_{\lambda \in (-\infty, +\infty)} \Phi(X_1^{r+1} + \lambda p_{n+1}^r) = \Phi(X^r + \lambda_{n+1}^r p_{n+1}^r) = \Phi(X^{r+1}). \quad (3.4.6)$$

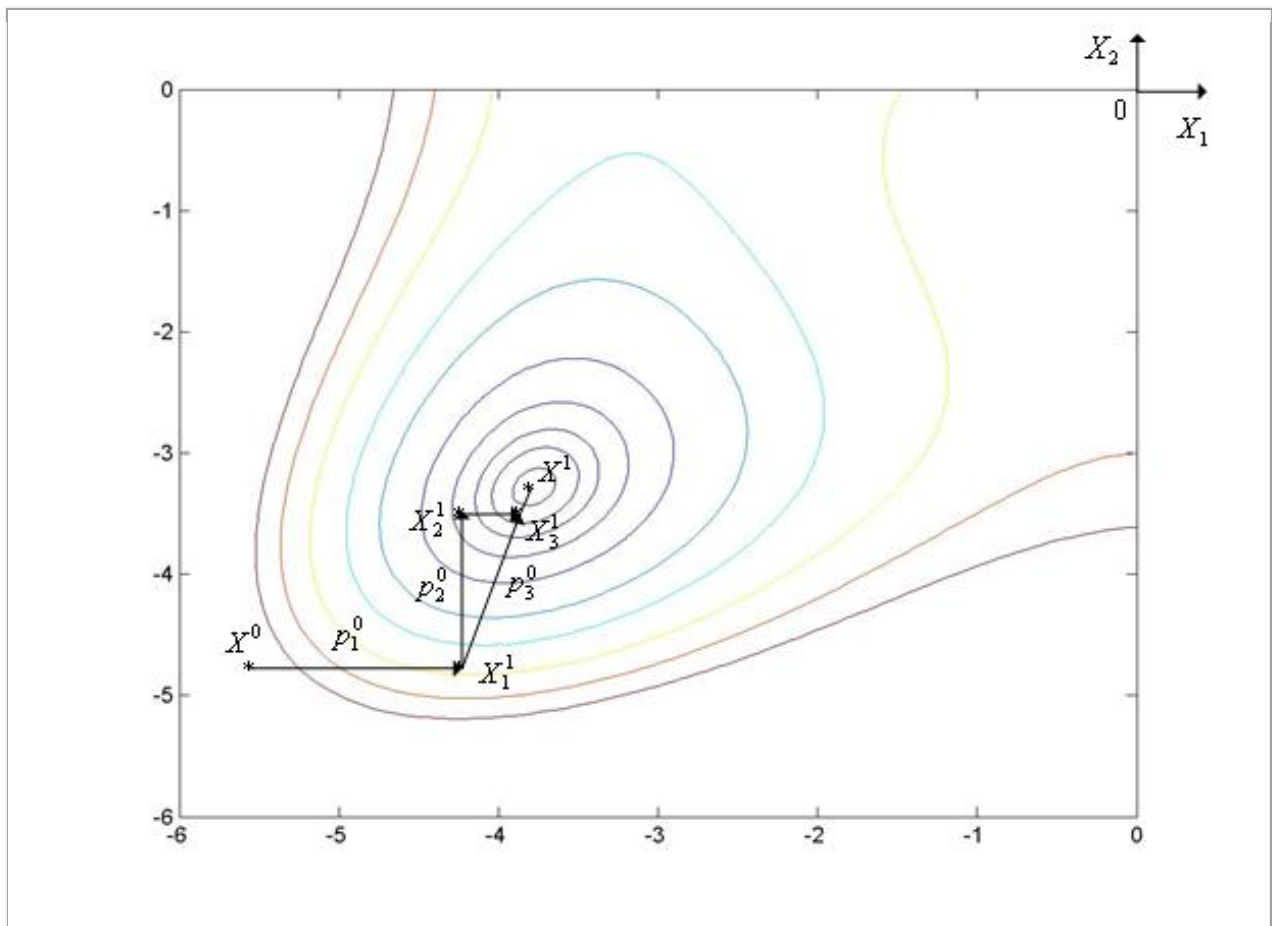


Рис. 3.4.1. Траектория поиска минимума функции Химмельблау методом сопряженных направлений

5. Если одно из стандартных условий окончания итераций

$$\|X^{r+1} - X^r\| \leq \varepsilon_x, \quad (3.4.7)$$

$$|\Phi(X^{r+1}) - \Phi(X^r)| \leq \varepsilon_\Phi \quad (3.4.8)$$

выполнено, то полагаем $X^* \approx X^{r+1}$, и заканчиваем вычисления. Иначе – полагаем $r = r + 1$ и переходим к п.2.

Метод сопряженных направлений иллюстрирует рис. 3.4.1, на котором показаны линии уровня функции Химмельблау ($n = 2$).

Рассмотрим двумерную квадратичную функцию:

$$\Phi(x_1, x_2) = x_1^2 + x_2^2 + 3(x_1 + x_2)^2 = 4x_1^2 + 4x_2^2 + 6x_1x_2. \quad (3.4.9)$$

На рис. 3.4.2 показаны линии уровня двумерной квадратичной функции и траектория поиска минимума квадратичной функции (3.4.9) методом сопряженных направлений.

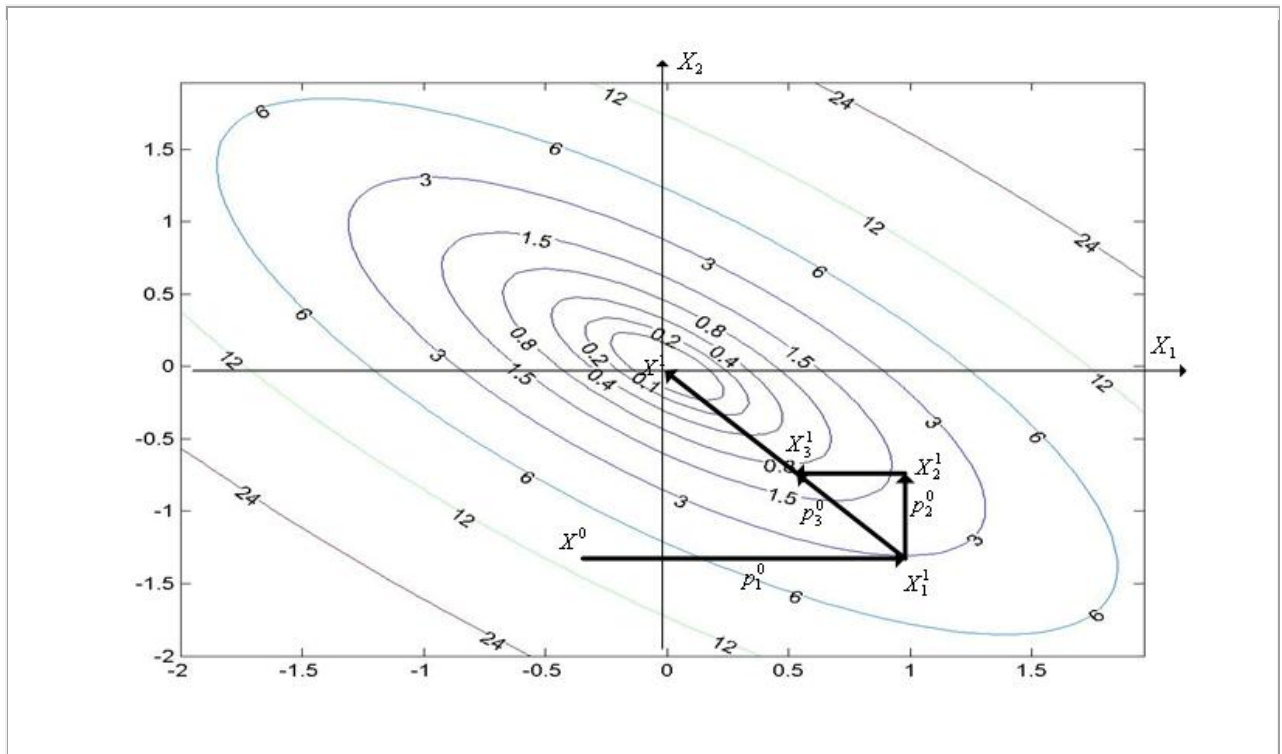


Рис. 3.4.2. Траектория поиска минимума квадратичной функции (3.4.9) методом сопряженных направлений

Произвольную n -мерную квадратичную функцию можно записать в виде

$$\Phi(X) = \frac{1}{2}(AX, X^T) + (b, X) + c, \quad (3.4.10)$$

где A – квадратная $n \times n$ матрица, b – $n \times 1$ столбец, c – скалярная константа.

Например, если положить

$$A = \begin{pmatrix} 8 & 6 \\ 6 & 8 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad c = 0,$$

то имеем функцию (3.4.9):

$$\begin{aligned} \Phi(X) &= \frac{1}{2}(AX, X^T) + (b, X) + c = \frac{1}{2}((8x_1 + 6x_2 \quad 6x_1 + 8x_2), (x_1, x_2)) = \\ &= \frac{1}{2}(8x_1^2 + 6x_1x_2 + 6x_1x_2 + 8x_2^2) = 4x_1^2 + 4x_2^2 + 6x_1x_2. \end{aligned}$$

Утверждение 3.4.1. В случае минимизации двумерной квадратичной функции (3.4.10) методом сопряженных направлений, направления p_1^r, p_3^r являются A - ортогональными.

Доказательство (см. рис. 3.4.2). По определению A - ортогональности для доказательства утверждения достаточно показать, что скалярное произведение

$$(Ap_3^r, p_1^r) = (A(X_3^r - X_1^r), p_1^r) = 0. \quad (3.4.11)$$

Легко видеть, что производная функции (3.4.10) равна

$$\Phi'(X) = AX + b.$$

Поэтому

$$A(X_3^r - X_1^r) = AX_3^r - AX_1^r = \Phi'(X_3^r) - \Phi'(X_1^r).$$

Подставляя этот результат в выражение (3.4.11), получим

$$(Ap_3^r, p_1^r) = \left((\Phi'(X_3^r) - \Phi'(X_1^r)), p_1^r \right) = (\Phi'(X_3^r), p_1^r) - (\Phi'(X_1^r), p_1^r) = 0.$$

Последнее равенство следует из ортогональности пар векторов $(\Phi'(X_3^r), p_1^r), (\Phi'(X_1^r), p_1^r)$.

Утверждение 3.4.1 объясняет название рассмотренного метода.

Заметим, что при минимизации квадратичной функции методом сопряженных направлений минимум достигается за одну итерацию.

3.5 Симплекс-метод

Рассмотрим следующую задачу локальной безусловной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного в n -мерном евклидовом пространстве R^n , $\min_{X \in R^n} \Phi(X) = \Phi(X^*) = \Phi^*$.

Регулярным симплексом в пространстве R^n называется правильный многогранник, образованный $(n+1)$ -ой равноотстоящими друг от друга вершинами.

Для случая $n=2$ – это равносторонний треугольник, для случая $n=3$ – тетраэдр.

Если в пространстве R^n необходимо построить регулярный симплекс, одна из вершин которого находится в точке $X^0 = (x_1^0, x_2^0, \dots, x_n^0)$, то координаты вершин такого симплекса удобно задавать с помощью $n \times (n+1)$ матрицы

$$R_{n \times (n+1)} = \begin{bmatrix} x_1^0 & x_1^0 + r_1 & x_1^0 + r_2 & \dots & x_1^0 + r_n \\ x_2^0 & x_2^0 + r_1 & x_2^0 + r_2 & \dots & x_2^0 + r_n \\ x_3^0 & x_3^0 + r_1 & x_3^0 + r_2 & \dots & x_3^0 + r_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n^0 & x_n^0 + r_1 & x_n^0 + r_2 & \dots & x_n^0 + r_n \end{bmatrix}. \quad (3.5.1)$$

Здесь i -й столбец представляет собой координаты i -й вершины симплекса, $i \in [1, \dots, n+1]$;

$$r_1 = l \frac{\sqrt{n+1} + n - 1}{n\sqrt{2}}, \quad r_2 = l \frac{\sqrt{n+1} - 1}{n\sqrt{2}}, \quad (3.5.2)$$

l – длина ребра симплекса.

Например, регулярный симплекс в двумерном пространстве R^2 с одной из вершин в начале координат (когда $x_1^0 = 0, x_2^0 = 0$) определяется (2×3) матрицей

$$R_{2 \times 3} = \begin{pmatrix} 0 & r_1 & r_2 \\ 0 & r_2 & r_1 \end{pmatrix} \text{ и имеет вид, представленный на рис. 3.5.1.}$$

В алгоритме симплекс-метода используется следующее важное свойство регулярного симплекса: если одну из вершин регулярного симплекса перенести на надлежащее расстояние вдоль прямой, соединяющей данную вершину и центр тяжести оставшихся вершин, то вновь получится регулярный симплекс (см. рис. 3.5.2).

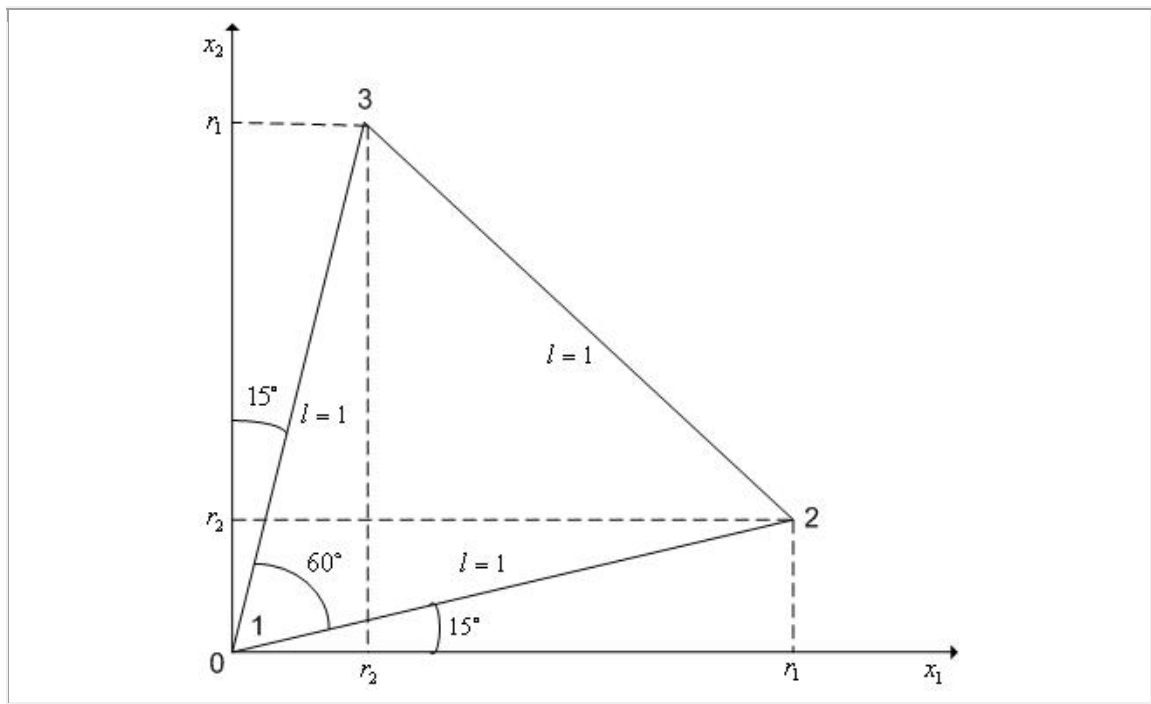


Рис. 3.5.1. Регулярный симплекс в пространстве R^2 с одной из вершин в начале координат

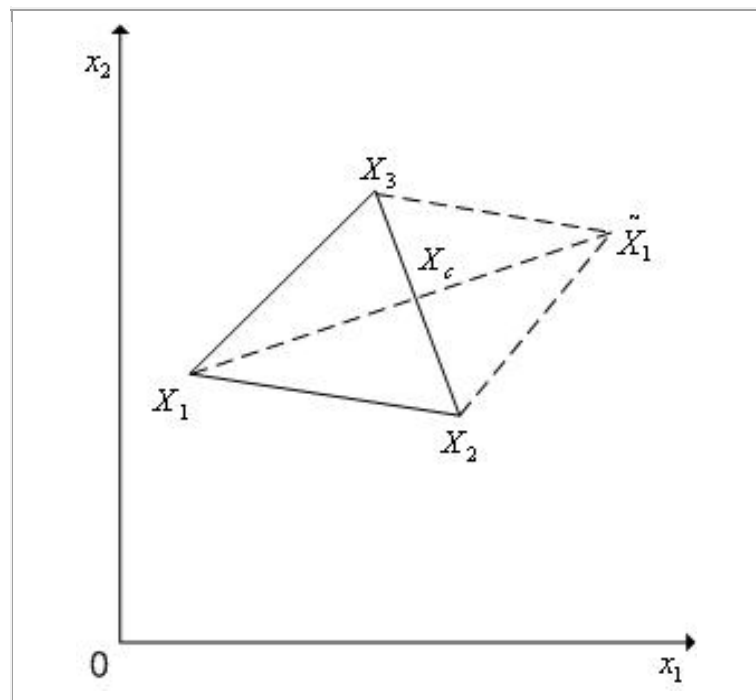


Рис. 3.5.2. Отражение вершины X_1 регулярного симплекса в пространстве R^2 относительно центра тяжести X_c остальных вершин

Будем называть эту процедуру **отражением вершины симплекса** относительно центра тяжести остальных вершин. На рис. 3.5.2 показано отражение вершины X_1 регулярного симплекса в пространстве R^2 относительно центра тяжести X_c остальных вершин.

Пусть $X_i^r, i \in [1, \dots, n+1]$ – векторы координат вершин регулярного симплекса. Тогда при выполнении операции отражения k -й вершины симплекса имеет место следующая связь координат этой вершины и новой вершины:

$$\frac{X_k^{r+1} + X_k^r}{2} = X_c^r. \quad (3.5.3)$$

Здесь

$$X_c^r = \frac{1}{n} \sum_{i=1, i \neq k}^{n+1} X_i^r \quad (3.5.4)$$

– вектор координат центра тяжести остальных вершин симплекса (за исключением отраженной вершины k).

Таким образом, после отражения k -ой вершины симплекса с координатами вершин $X_i^r, i \in [1, \dots, n+1]$, получаем новый симплекс с координатами вершин

$$X_i^{r+1} = X_i^r, i \in [1, \dots, n+1], i \neq k, X_k^{r+1} = 2X_c^r - X_k^r. \quad (3.5.5)$$

Кроме операции отражения вершины симплекса симплекс-метод может использовать операцию **редукции симплекса** – уменьшение длин всех ребер симплекса на одну и ту же величину. На рис 3.5.3 показана редукция вершин регулярного симплекса в пространстве R^2 к вершине X_1 .

Пусть $X_i^r, i \in [1, \dots, n+1]$ – векторы координат вершин регулярного симплекса. Тогда при выполнении операции редукции вершин этого симплекса к вершине X_k новые координаты остальных вершин симплекса определяются по формуле

$$X_i^{r+1} = X_k^r + \gamma (X_i^r - X_k^r), i \in [1, \dots, n+1], i \neq k,$$

где $\gamma \in (0,1)$ – коэффициент редукции. Рекомендуется использовать $\gamma = 0.5$.

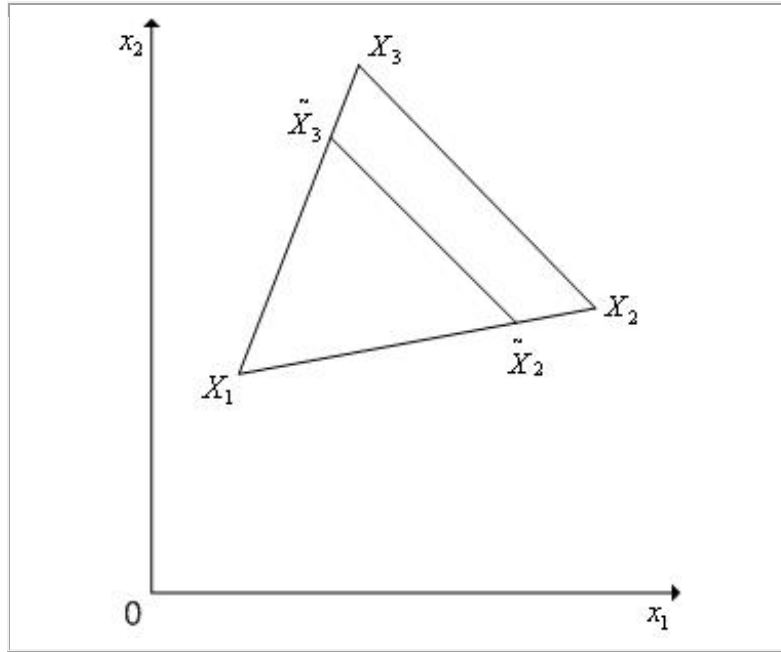


Рис. 3.5.3. редукция вершин регулярного симплекса в пространстве R^2 к вершине X_1

Таким образом, после редукции вершин симплекса $X_i^r, i \in [1, \dots, n+1]$ к вершине X_k получаем новый симплекс с координатами вершин

$$X_i^{r+1} = X_k^r + \gamma(X_i^r - X_k^r), i \in [1, \dots, n+1], i \neq k, \quad X_k^{r+1} = X_k^r. \quad (3.5.6)$$

Суть симплекс-метода раскрывает его простейший вариант.

Схема простейшего варианта симплекс-метода

1. Задаем начальную точку X^0 , длину ребра симплекса l и полагаем $r=0$.
2. По формулам (3.5.1), (3.5.2) находим координаты $X_i^r, i \in [1, \dots, n+1]$ всех вершин симплекса.
3. Вычисляем значения $\Phi(X_i^r), i \in [1, \dots, n+1]$ минимизируемой функции во всех вершинах симплекса.
4. Находим максимальное из значений функции $\Phi(X)$ в вершинах симплекса $X_i^r, i \in [1, \dots, n+1]$

$$\max\left(\Phi(X_1^r), \Phi(X_2^r), \dots, \Phi(X_{n+1}^r)\right) = \Phi(X_k^r).$$

5. По формулам (3.5.4), (3.5.5) отражаем вершину X_k^r относительно центра тяжести остальных вершин симплекса – получаем новый симплекс с координатами вершин $X_i^{r+1}, i \in [1, \dots, n+1]$.

6. Вычисляем значение $\Phi(X_k^{r+1})$ минимизируемой функции в новой вершине симплекса.

7. Если условие окончания итераций (см. ниже) выполнено, то в качестве приближенного значения точки минимума функции $\Phi(X)$ принимаем ту вершину симплекса $X_i^{r+1} = X_i^r, i \in [1, \dots, n+1]$, в которой $\Phi(X)$ имеет минимальное значение, и заканчиваем вычисления. Иначе полагаем $r = r + 1$ и переходим к п. 4.

Поскольку размер симплекса в простейшем варианте симплекс-методе фиксирован, в качестве условия окончания итераций в данном случае можно использовать условие:

$$\max_{i \in [1, \dots, n+1], i \neq k} \left| \Phi(X_i^{r+1}) - \Phi(X_k^{r+1}) \right| \leq \varepsilon_\Phi, \quad (3.5.7)$$

где ε_Φ – требуемая точность решения по Φ , $k \in [1, \dots, n+1]$ – номер произвольной вершины симплекса. Отметим, что выражение в левой части неравенства (3.5.7) есть максимальная разность значений функции $\Phi(X)$ в двух вершинах симплекса $X_i^{r+1} = X_i^r, i \in [1, \dots, n+1]$.

Простейший вариант симплекс-метода иллюстрирует рис. 3.5.4, на котором показаны линии уровня функции Химмельблау ($n = 2$).

Рассмотренный простейший симплекс-метод склонен к закливанию и медленно сходится, если длина ребра симплекса l выбрана малой (выбор же большой длины ребра симплекса обеспечивает высокую скорость сходимости, но дает малую точность решения). Поэтому в вычислительной практике используются различные модификации простейшего метода, направленные на преодоление его указанных недостатков.

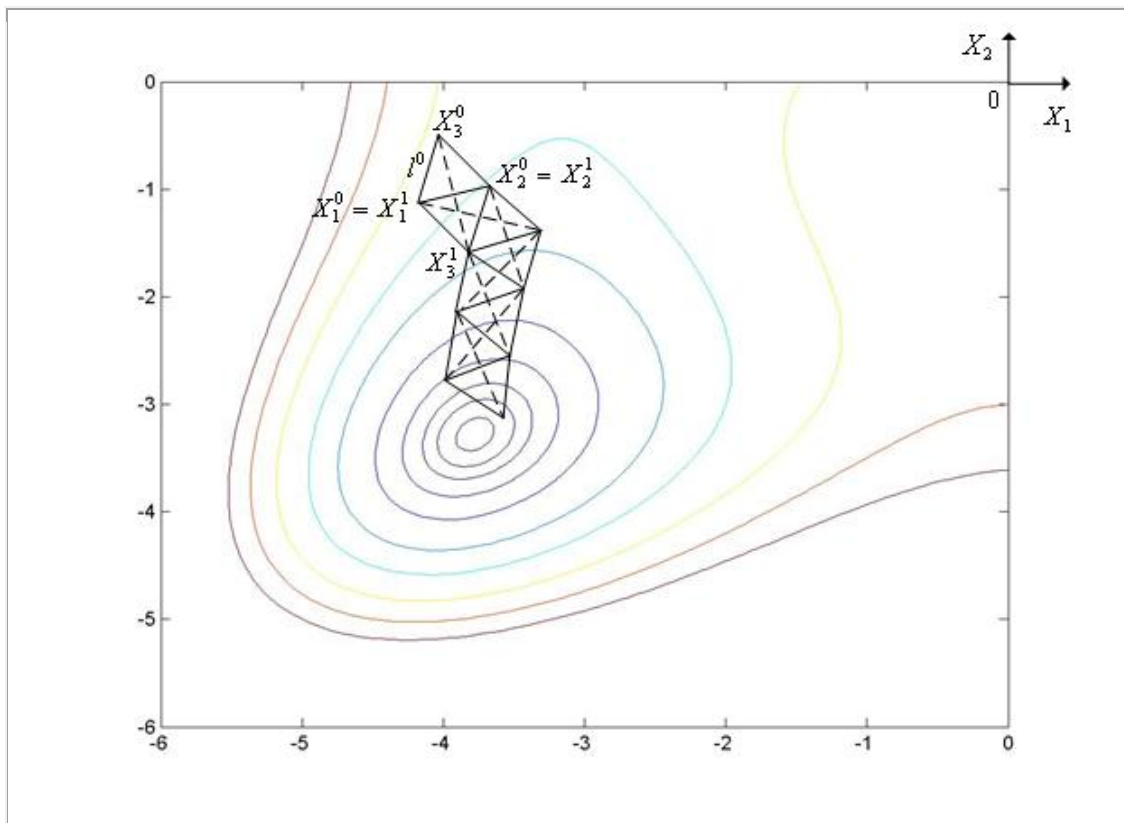


Рис. 3.5.4. Траектория поиска минимума функции Химмельблау простейшим симплекс-методом

Модифицированный симплекс-метод

Основной идеей модифицированного симплекс-метода является изменение по некоторому правилу размера симплекса в процессе поиска.

При этом наряду с условием (3.5.7) в качестве условия окончания итераций можно использовать условие

$$l^r \leq \varepsilon_x, \quad (3.5.8)$$

где l^r – текущая длина ребра симплекса, ε_x - требуемая точность решения по X .

Обычно размер симплекса изменяется при выполнении следующих условий:

- при «накрытии» симплексом дна оврага или точки минимума;
- при циклическом движении.

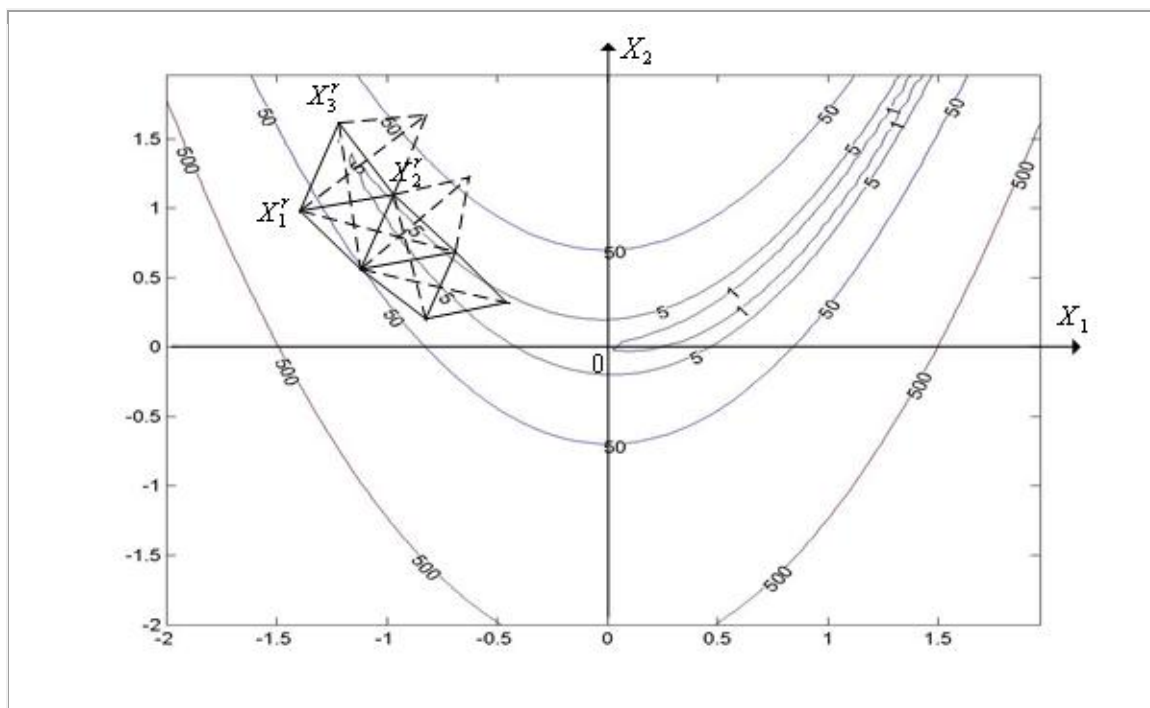


Рис. 3.5.5. Траектория поиска минимума функции Розенброка модифицированным симплекс-методом при «накрытии» дна оврага. Пунктиром показаны отвергнутые симплексы

«Накрытие» симплексом дна оврага или точки минимума. Пусть X_k^{r+1} – вершина, которая получилась на r -ой итерации в результате отражения вершины X_k^r . Так что координаты вершин нового симплекса равны $X_i^{r+1} = X_i^r, i \in [1, \dots, n+1], i \neq k, X_k^{r+1}$.

Ситуация $\Phi(X_k^{r+1}) > \Phi(X_k^r)$ интерпретируется как «накрытие» этим симплексом дна оврага или точки минимума и простейший симплекс-метод модифицируется следующим образом (см. рис. 3.5.5):

1. Полагаем $k = k+1$ (если $k = n+2$, то полагаем $k = 1$);
2. Выполняем отражение k -ой вершины симплекса $X_i^r, i \in [1, \dots, n+1]$;
3. Если $\Phi(X_k^{r+1}) > \Phi(X_k^r)$ и не все вершины перебраны, то переходим к п.1.
4. Иначе – продолжаем итерации по схеме простейшего симплекс-метода.

На рис.3.5.5 показана траектория поиска минимума функции Розенброка модифицированным симплекс-методом при «накрытии» дна оврага. Пунктиром показаны отвергнутые симплексы.

Циклическое движение

Ситуация, когда некоторая вершина симплекса не исключается на протяжении m итераций, интерпретируется как «зацикливание» алгоритма.

Простейший симплекс-метод модифицируется в этом случае следующим образом:

1. Находим вершину текущего симплекса $X_i^r, i \in [1, \dots, n+1]$, в которой функция $\Phi(X)$ принимает наименьшее значение $\min(\Phi(X_1^0), \Phi(X_2^r), \dots, \Phi(X_{n+1}^r)) = \Phi(X_k^r)$.
2. По формуле (3.5.6) выполняем редукцию симплекса $X_i^r, i \in [1, \dots, n+1]$ к вершине X_k^r .
3. Продолжаем итерации по схеме простейшего симплекс-метода.

Здесь количество итераций m рекомендуется находить из условия $m = \lceil 1.65n + 0.05n^2 \rceil$, где $\lceil * \rceil$ – символ ближайшего целого большего.

3.6 Метод деформируемого многогранника (Нелдера-Мида)

Рассмотрим следующую задачу локальной безусловной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного в n -мерном евклидовом пространстве R^n , $\min_{X \in R^n} \Phi(X) = \Phi(X^*) = \Phi^*$.

В случае если функция $\Phi(X)$ является овражной функцией, эффективность симплекс-метода при решении задачи значительно снижается в силу того, что регулярный симплекс нельзя «вытянуть» вдоль оврага.

Метод Нелдера-Мида (метод деформируемого многогранника) является развитием симплекс-метода и использует в процессе поиска деформацию (изменение размеров и формы) текущего симплекса (не обязательно регулярного).

Метод использует следующие операции над симплексами:

- отражение;
- редукция;
- сжатие;
- растяжение.

Отражение вершины симплекса относительно центра тяжести остальных вершин (см. рис. 3.6.1). В результате отражения k -й вершины симплекса с координатами вершин $X_i^r, i \in [1, \dots, n+1]$, получаем новый симплекс с координатами вершин

$$X_i^{r+1} = X_i^r, i \in [1, \dots, n+1], i \neq k, X_k^{r+1} = 2X_c^r - X_k^r, \quad (3.6.1)$$

где

$$X_c^r = \frac{1}{n} \sum_{i=1, i \neq k}^{n+1} X_i^r \quad (3.6.2)$$

– вектор координат центра тяжести остальных вершин симплекса (за исключением отраженной вершины k).

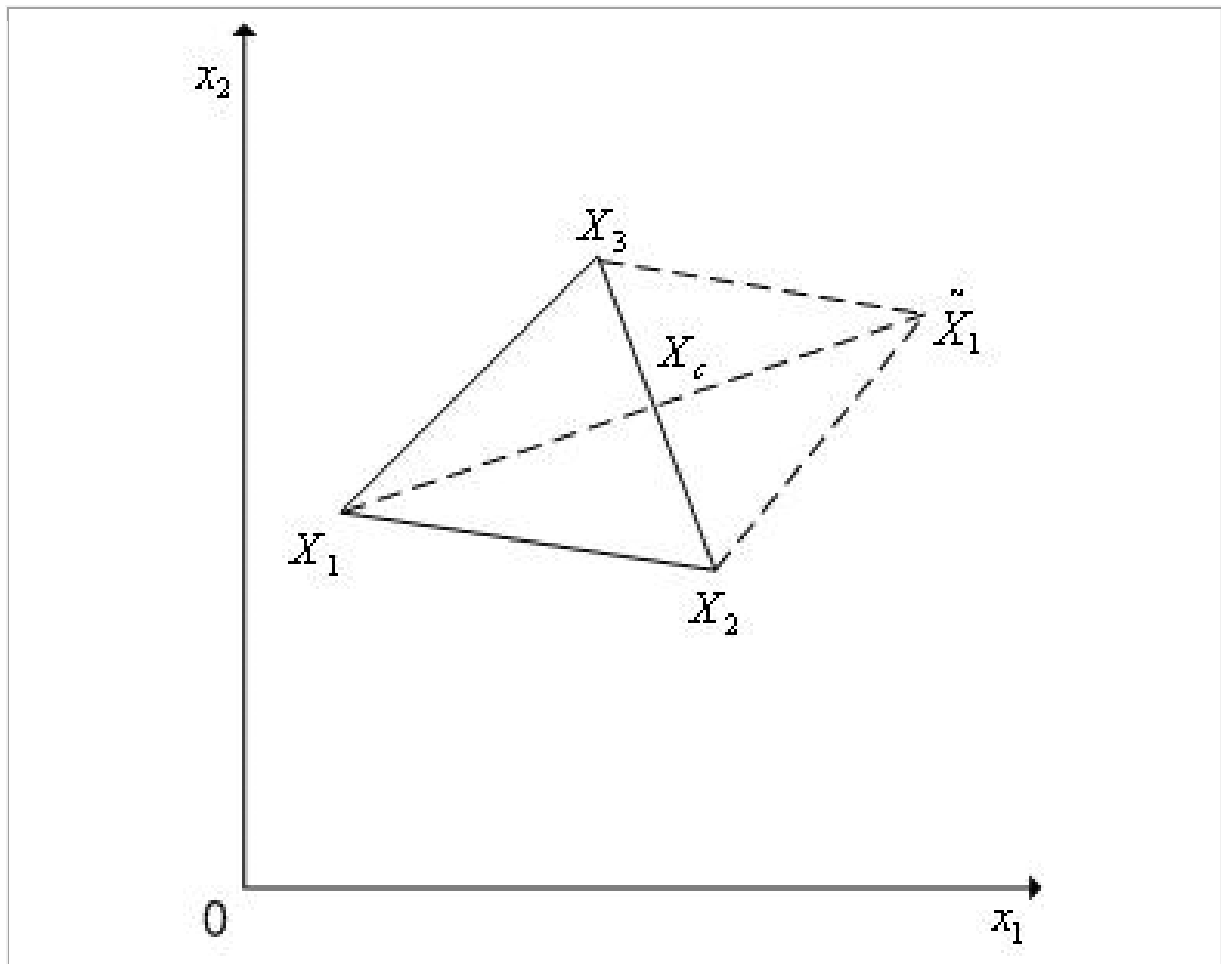


Рис. 3.6.1. Отражение вершины X_1 симплекса в пространстве R^2 относительно центра тяжести X_c остальных вершин

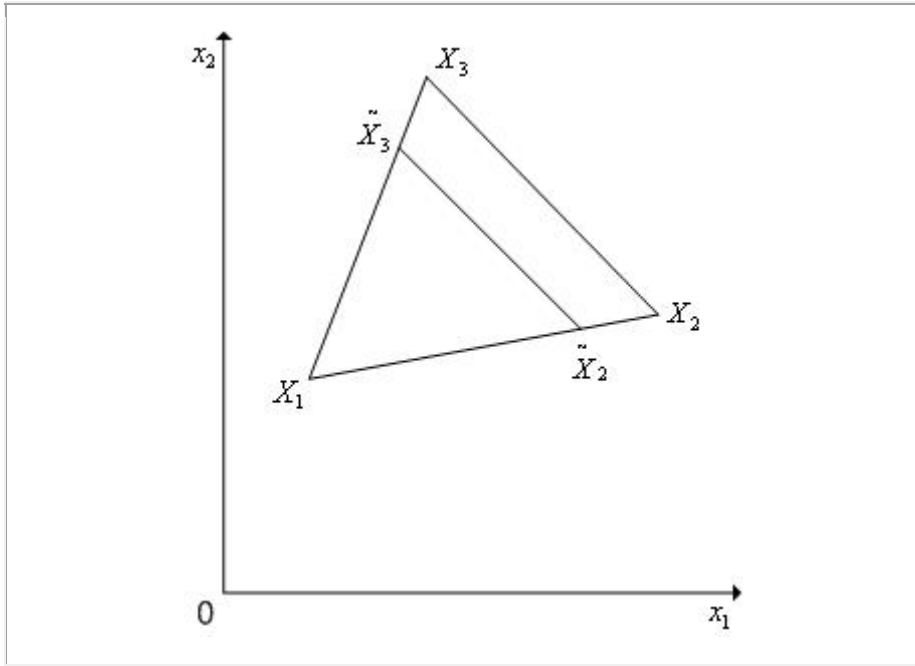


Рис. 3.6.2. Редукция вершин симплекса в пространстве R^2 к вершине X_1

Редукции симплекса (см. рис. 3.6.2) – уменьшение длин всех ребер симплекса в одно и то же количество раз. В результате выполнения редукции вершин симплекса $X_i^r, i \in [1, \dots, n+1]$ к вершине X_k получаем новый симплекс с координатами вершин

$$X_i^{r+1} = X_k^r + \gamma(X_i^r - X_k^r), i \in [1, \dots, n+1], i \neq k, \quad X_k^{r+1} = X_k^r, \quad (3.6.3)$$

где $\gamma \in (0,1)$, $\gamma \approx 0.5$ – коэффициент редукции.

Сжатие симплекса (см. рис. 3.6.3). В результате выполнения сжатия симплекса $X_i^r, i \in [1, \dots, n+1]$ в направлении $(X_k^r - X_c^r)$ получаем новый симплекс с координатами вершин

$$X_i^{r+1} = X_i^r, i \in [1, \dots, n+1], i \neq k, X_k^{r+1} = X_c^r + \beta(X_k^r - X_c^r), \quad (3.6.4)$$

где $\beta \in (0,1)$, $\beta \approx 0.4-0.6$ – коэффициент сжатия, X_c^r – вектор координат центра тяжести остальных вершин симплекса.

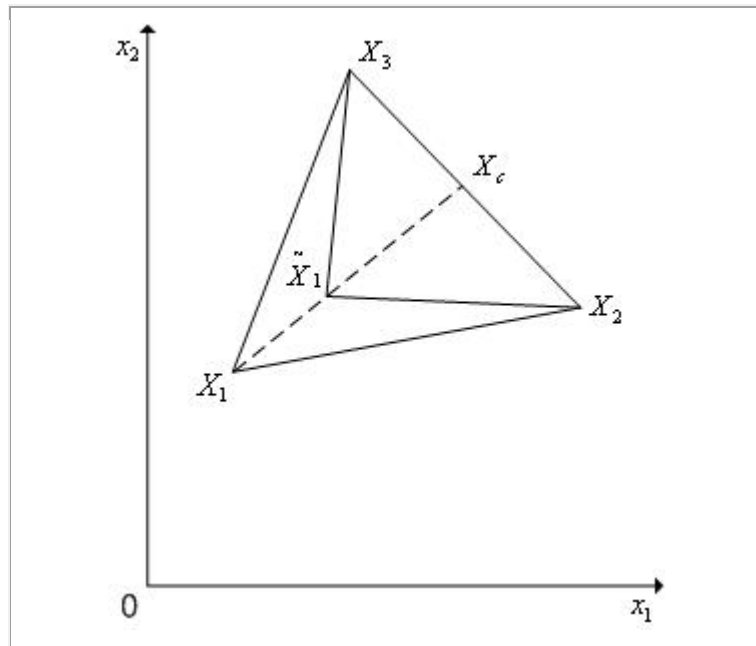


Рис. 3.6.3. Сжатие симплекса в пространстве R^2 в направлении $(X_1 - X_c)$

Растяжение симплекса (см. рис.3.6.4). В результате выполнения растяжения симплекса $X_i^r, i \in [1, \dots, n+1]$ в направлении $(X_k^r - X_c^r)$ получаем новый симплекс с координатами вершин

$$X_i^{r+1} = X_i^r, i \in [1, \dots, n+1], i \neq k, X_k^{r+1} = X_c^r + \alpha(X_k^r - X_c^r), \quad (3.6.5)$$

где $\alpha \approx 2.8 - 3.0$ – коэффициент растяжения, X_c^r – вектор координат центра тяжести остальных вершин симплекса.

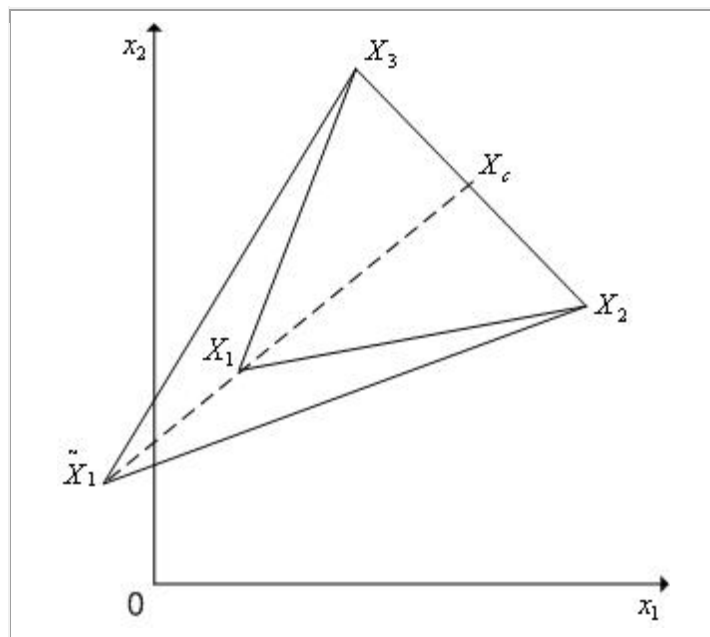


Рис. 3.6.4. Растяжение симплекса в пространстве R^2 в направлении $(X_1 - X_c)$

Схема метода Нелдера-Мида

Симплекс с вершинами $X_i^r, i \in [1, \dots, n+1]$ обозначим S^r .

1. Задаем начальную точку X^0 , длину ребра симплекса l и полагаем $r=0$.
2. Находим координаты $X_i^0, i \in [1, \dots, n+1]$ всех вершин регулярного симплекса S^0 с длиной ребра l . Вычисляем значения $\Phi(X_i^r)$ минимизируемой функции во всех вершинах симплекса.
3. Среди вершин симплекса S^r находим вершины $X_{l=k_l}^r, X_{h=k_h}^r, X_{s=k_s}^r$, в которых функция $\Phi(X)$ принимает, соответственно, наименьшее, наибольшее и следующее за максимальным значения, а также находим значения функции $\Phi(X)$ в этих точках:

$$\Phi(X_l^r) = \min_{i \in [1, \dots, n+1]} \Phi(X_i^r),$$

$$\Phi(X_h^r) = \max_{i \in [1, \dots, n+1]} \Phi(X_i^r),$$

$$\Phi(X_s^r) = \max_{i \in [1, \dots, n+1], i \neq h} \Phi(X_i^r).$$

4. По формулам (3.6.1), (3.6.2) выполняем отражение вершину симплекса X_h^r относительно центра тяжести остальных вершин симплекса – получаем новый симплекс S^{r+1} . Вычисляем значение $\Phi(X_{k_h}^{r+1})$ минимизируемой функции в новой вершине симплекса.
5. Если условие окончания итераций (см. ниже) выполнено, то в качестве приближенного значения точки минимума функции $\Phi(X)$ принимаем ту вершину симплекса S^{r+1} , в которой $\Phi(X)$ имеет минимальное значение, и заканчиваем вычисления.

6. Если $\Phi\left(X_{k_h}^{r+1}\right) \leq \Phi\left(X_h^r\right)$ и $\Phi\left(X_{k_h}^{r+1}\right) \leq \Phi\left(X_l^r\right)$, то переходим к п.7 (растяжению симплекса) – см. рис. 3.6.5. Если $\Phi\left(X_{k_h}^{r+1}\right) \leq \Phi\left(X_h^r\right)$, но $\Phi\left(X_{k_h}^{r+1}\right) \geq \Phi\left(X_l^r\right)$, то переходим к п.3 (отражению) – см. рис. 3.6.6. Если $\Phi\left(X_{k_h}^{r+1}\right) \geq \Phi\left(X_h^r\right)$, то переходим к п.8 (сжатию симплекса) – см. рис. 3.6.7, рис. 3.6.8.

7. Ситуация $\Phi\left(X_{k_h}^{r+1}\right) \leq \Phi\left(X_h^r\right)$ и $\Phi\left(X_{k_h}^{r+1}\right) \leq \Phi\left(X_l^r\right)$. По формуле (3.6.5) выполняем растяжение симплекса S^{r+1} в направлении $\left(X_{k_h}^{r+1} - X_c^{r+1}\right)$ – получаем новый симплекс S^{r+2} . Вычисляем значение минимизируемой функции в новой вершине симплекса $X_{k_h}^{r+2}$. Если $\Phi\left(X_{k_h}^{r+2}\right) \leq \Phi\left(X_l^r\right)$, то полагаем $r = r + 2$ и переходим к п.3 (отражению вершины симплекса). Иначе полагаем $r = r + 1$ и переходим к п.3 (отражению вершины симплекса) с симплексом S^r (т.е. не используем результаты растяжения).

8. Ситуация $\Phi\left(X_{k_h}^{r+1}\right) \geq \Phi\left(X_h^r\right)$. По формуле (3.6.4) выполняем сжатие симплекса S^{r+1} в направлении $\left(X_{k_h}^{r+1} - X_c^{r+1}\right)$ – получаем новый симплекс S^{r+2} . Вычисляем значение минимизируемой функции в новой вершине симплекса $X_{k_h}^{r+2}$.

Если $\Phi\left(X_{k_h}^{r+2}\right) \leq \Phi\left(X_{k_h}^r\right)$, то полагаем $r = r + 2$ и переходим к п.3 (отражению вершины симплекса). Иначе по формуле (3.6.3) выполняем редукцию симплекса S^r к вершине $X_{l=k_l}^r$ – получаем новый симплекс S^{r+1} .

Вычисляем значение минимизируемой функции во всех новых вершинах симплекса S^{r+1} .

Полагаем $r = r + 1$ и переходим к п.3 (отражению симплекса).

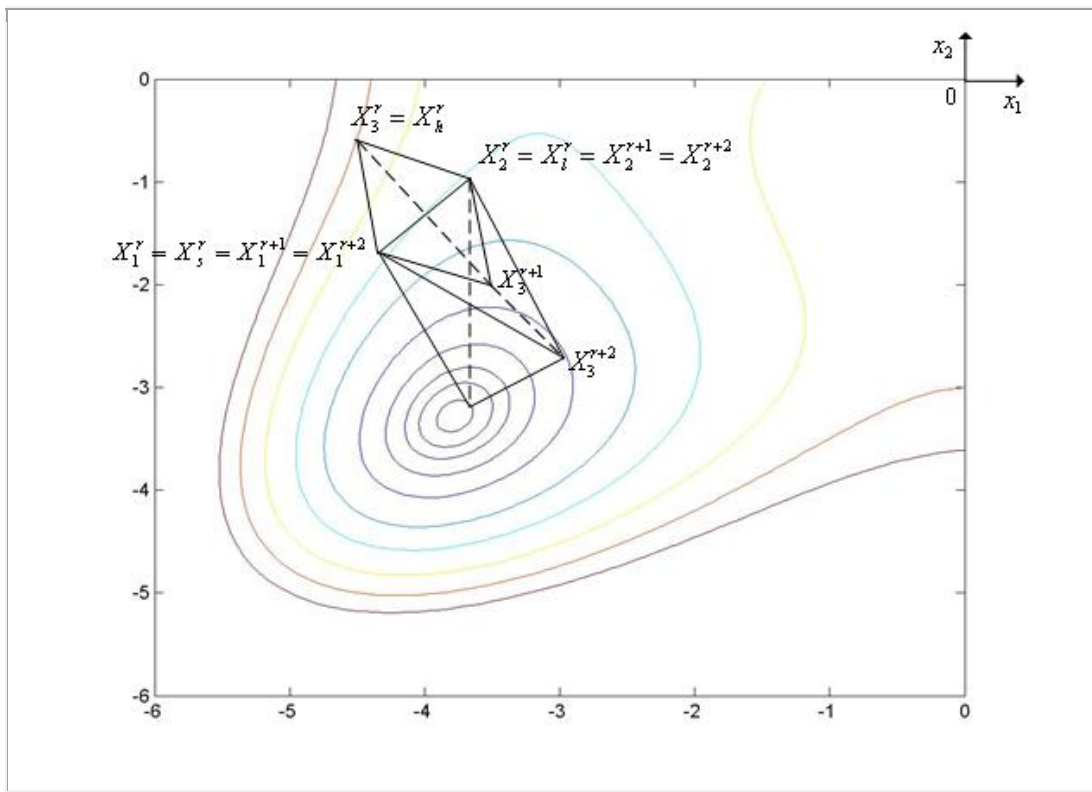


Рис. 3.6.5. Ситуация, когда метод Нелдера-Мида использует успешное растяжение симплекса

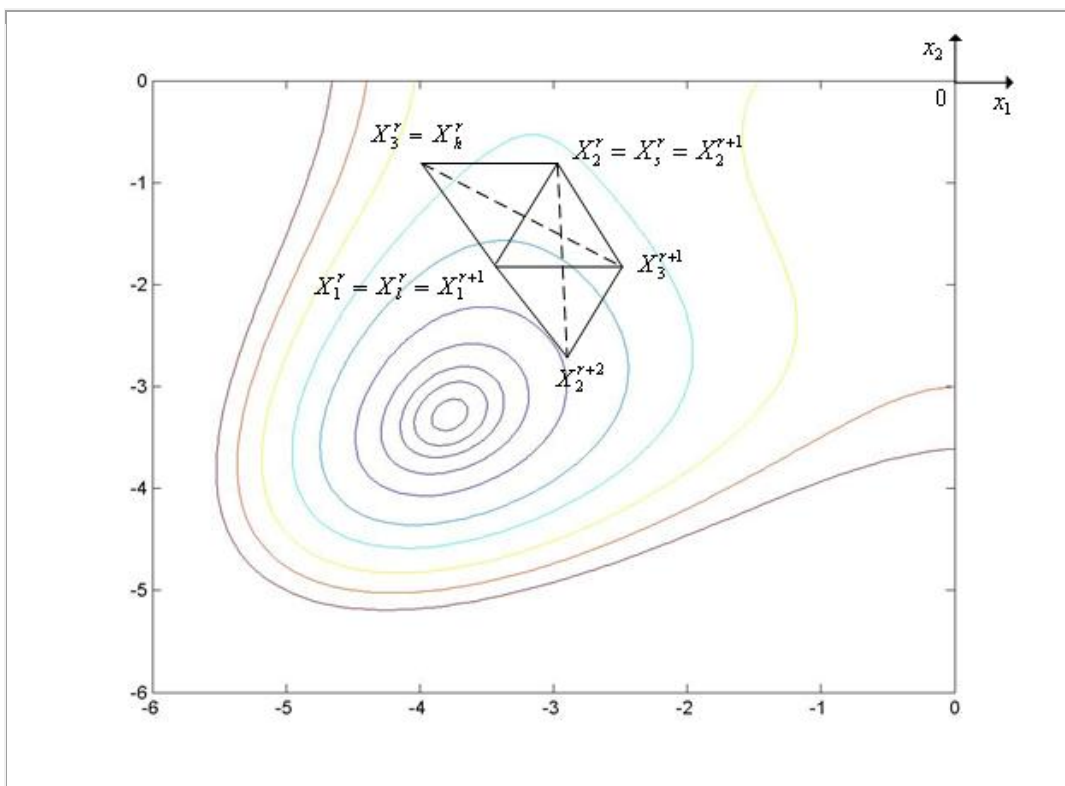


Рис. 3.6.6. Ситуация, когда метод Нелдера-Мида использует успешное отражение симплекса

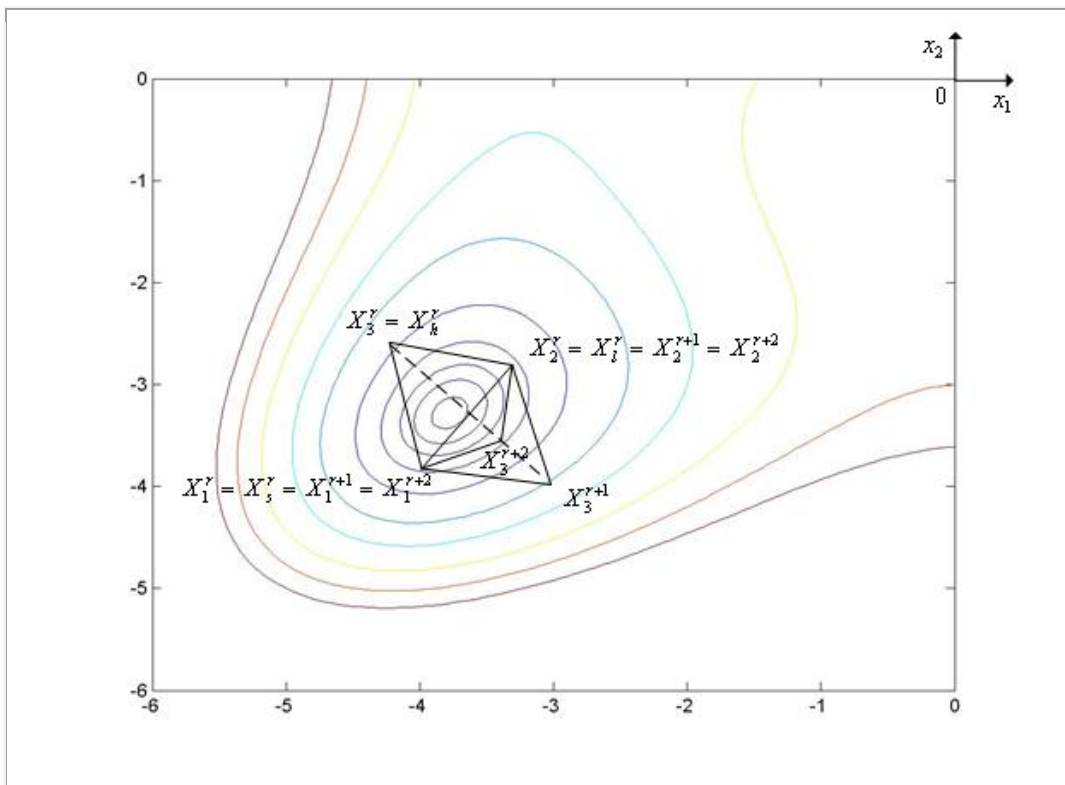


Рис. 3.6.7. Ситуация, когда метод Нелдера-Мида использует успешное сжатие симплекса

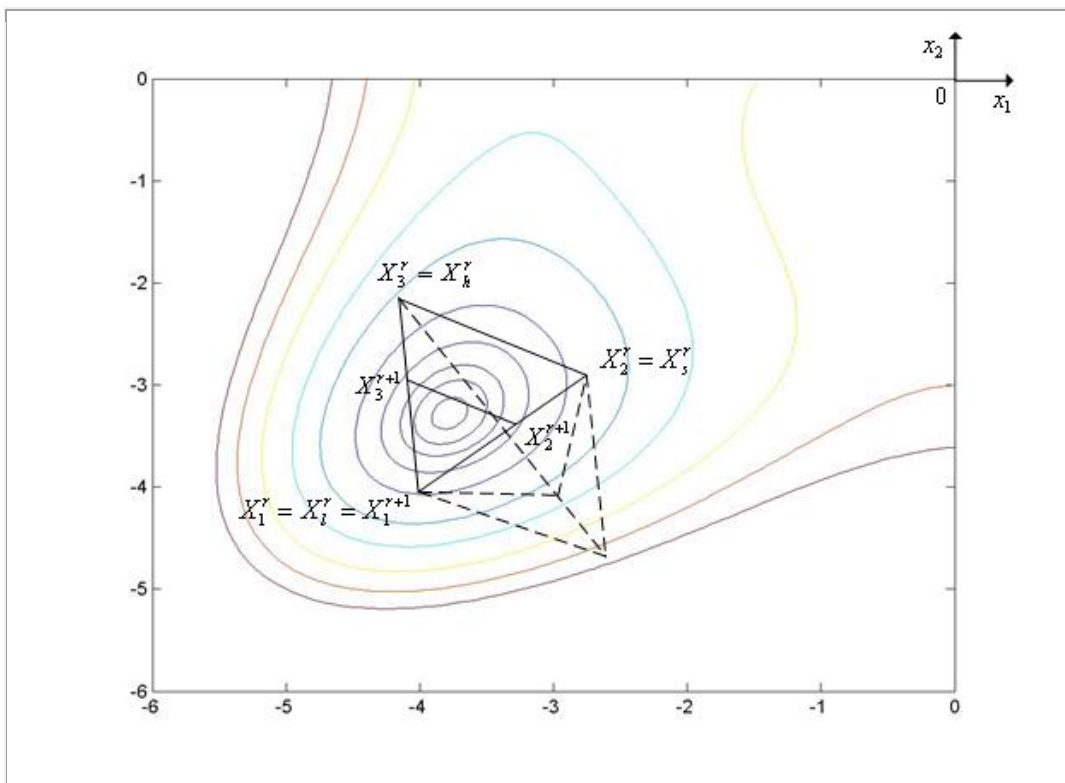


Рис. 3.6.8. Ситуация, когда метод Нелдера-Мида использует редукцию после неудачного сжатия симплекса. Пунктиром показаны отвергнутые (неудачные) итерации

На рис. 3.6.5 – рис. 3.6.8 минимизируемой функцией $\Phi(X)$ является функция Химмельблау.

В качестве условия окончания итераций в методе Нелдера-Мида можно использовать условие:

$$\max_{i \in [1, \dots, n+1], i \neq k} \left| \Phi(X_i^{r+1}) - \Phi(X_k^{r+1}) \right| \leq \varepsilon_\Phi, \quad (3.6.6)$$

где ε_Φ – требуемая точность решения по Φ , $k \in [1, \dots, n+1]$ – номер произвольной вершины симплекса. Можно также завершать итерации, когда длина максимального из ребер текущего симплекса станет меньше или равна ε_X – требуемой точности решения по X .

Изложенная схема метода Нелдера–Мида имеет тот недостаток, что для сильно овражных функций может происходить вырождение («сплющивание») симплекса. Поэтому к рассмотренной схеме метода Нелдера-Мида добавляется этап периодического (через N итераций) **восстановления симплекса**, который заключается в следующем:

- в текущем симплексе выбираются две «лучшие» вершины и определяется расстояние между ними \tilde{l} ;
- исходя из «лучшей» вершины текущего симплекса строится новый симплекс, длина ребра которого принимается равной \tilde{l} (см. параграф 3.5).

3.7 Метод наискорейшего спуска. Метод дробления шага

Рассматривается следующая многомерная задача локальной безусловной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного в n -мерном евклидовом пространстве R^n , $\min_{X \in R^n} \Phi(X) = \Phi(X^*) = \Phi^*$.

Положим, что функция $\Phi(X)$ всюду дифференцируема в n -мерном евклидовом пространстве R^n .

Направление спуска в градиентных методах оптимизации совпадает с направлением антиградиента минимизируемой функции $\Phi(X)$.

Итерационная формула градиентных методов оптимизации имеет вид

$$X^{r+1} = X^r + \lambda^r S^r. \quad (3.7.1)$$

Здесь λ^r – длина шага на r -ой итерации в направлении S^r , где

$$S^r = -\frac{\nabla \Phi^r}{\|\nabla \Phi^r\|} \quad (3.7.2)$$

– единичный вектор направления антиградиента функции $\Phi(X)$ в точке X^r , $\|\cdot\|$ – некоторая векторная норма, например, евклидова. Напомним, что градиент функции $\Phi(X)$ в точке X^r есть значение вектора частных производных этой функции в точке X^r :

$$\nabla \Phi^r = \nabla \Phi(X^r) = \left(\begin{array}{c} \frac{\partial \Phi(X)}{\partial x_1} \\ \frac{\partial \Phi(X)}{\partial x_2} \\ \dots \\ \frac{\partial \Phi(X)}{\partial x_n} \end{array} \right)_{X=X^r}.$$

Различные градиентные методы оптимизации отличаются между собой правилами выбора длины шага λ^r .

Градиентный метод наискорейшего спуска

Градиентный метод наискорейшего спуска в качестве длины шага λ^r использует величину, при которой достигается минимум функции $\Phi(X)$ в направлении S^r :

$$\Phi(X^{r+1}) = \Phi(X^r + \lambda^r S^r) = \min_{\lambda \in R^1} \Phi(X^r + \lambda S^r). \quad (3.7.3)$$

Задача (3.7.1) есть одномерная задача локальной безусловной оптимизации, которая может быть решена рассмотренными в разделе 2 методами.

Схема метода

1. Задаем начальную точку X^0 и полагаем счетчик числа итераций $r=0$.
2. По формуле (3.7.2) вычисляем компоненты вектора S^r .

3. Каким-либо методом решаем одномерную задачу безусловной оптимизации (3.7.3) – определяем точку X^{r+1} .
4. Вычисляем величину $\Phi(X^{r+1})$ – значение функции $\Phi(X)$ в точке X^{r+1} .
5. Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем $X^* \approx X^{r+1}$ и завершаем итерации. Иначе – полагаем $r = r + 1$, переходим к п. 2.

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

$$\|X^{r+1} - X^r\| = \lambda^r \leq \varepsilon_X, \quad (3.7.4)$$

$$|\Phi(X^{r+1}) - \Phi(X^r)| \leq \varepsilon_\Phi. \quad (3.7.5)$$

В качестве критерия окончания поиска можно использоваться также условие

$$\|\nabla \Phi^r\| \leq \varepsilon_{\nabla}, \quad (3.7.6)$$

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

Градиентный метод наискорейшего спуска иллюстрирует рис. 3.7.1, на котором показан фрагмент линий уровня функции Химмельблау.

Градиентный метод с дроблением шага

В градиентном методе с дроблением шага точка X^{r+1} определяется по формуле:

$$X^{r+1} = X^r + \lambda^r S^r, \quad (3.7.7)$$

где величина шага λ^r находится из условия

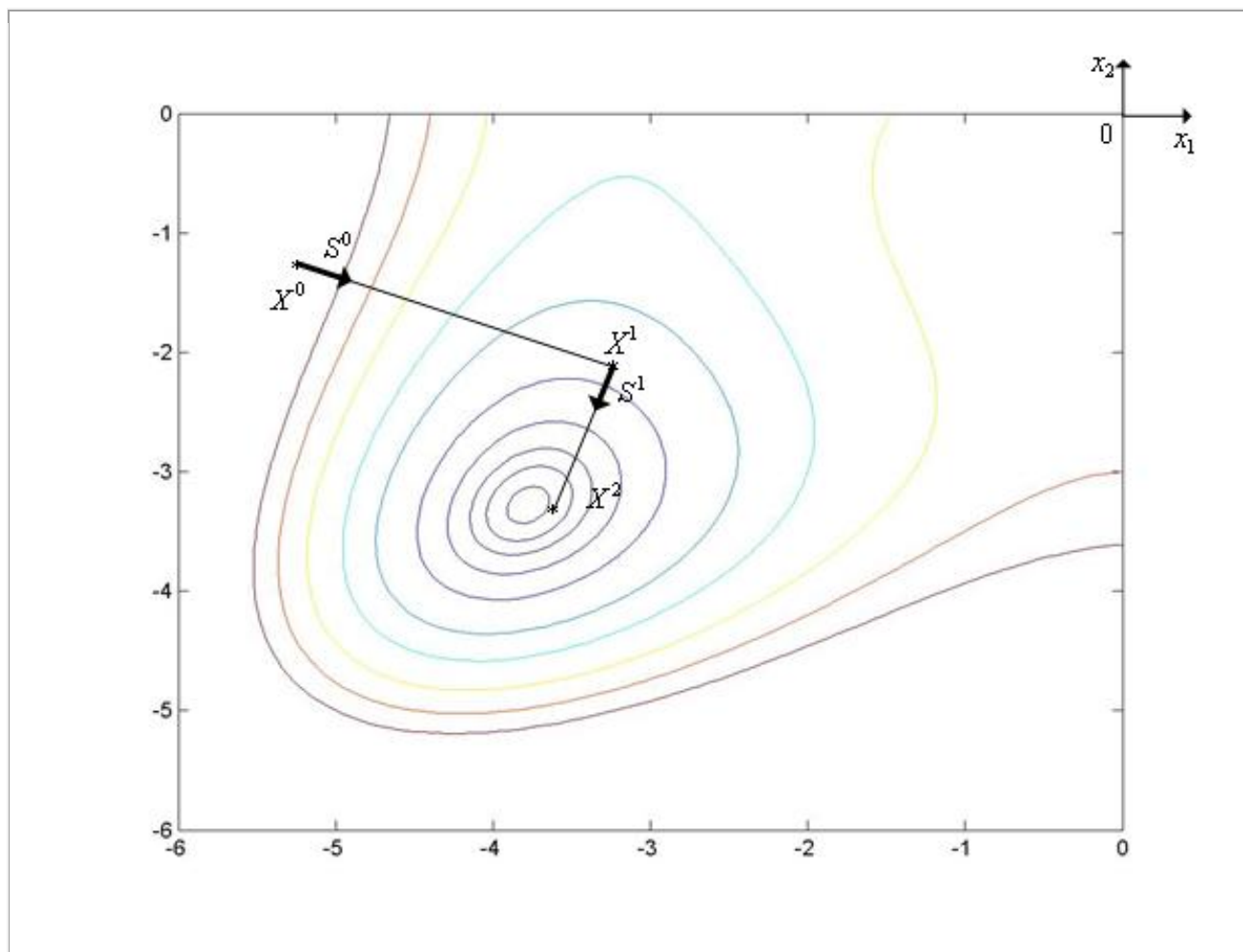


Рис. 3.7.1. Траектория поиска минимума функции Химмельблау градиентным методом наискорейшего спуска

$$\Phi(X^r) - \Phi(X^{r+1}) \geq 0.5\lambda^r \|\nabla\Phi^r\|. \quad (3.7.8)$$

Схема метода

1. Задаем начальную точку X^0 , начальную величину шага λ^0 и коэффициент дробления шага $\nu \in (0, 1]$. Полагаем счетчик числа итераций $r = 0$.
2. По формуле (3.7.7) вычисляем компоненты вектора X^{r+1} .
3. Вычисляем величину $\Phi(X^{r+1})$ – значение функции $\Phi(X)$ в точке X^{r+1} .
4. Если условие (3.7.8) выполнено, то переходим к следующему пункту. Иначе – переходим к пункту 6.
5. Полагаем $\lambda^r = \nu\lambda^r$ и переходим к пункту 2.
6. Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем $X^* \approx X^{r+1}$, и завершаем итерации. Иначе – полагаем $r = r + 1$ переходим к п. 2.

В качестве критерия окончания поиска можно использовать условия (3.7.4) – (3.7.6).

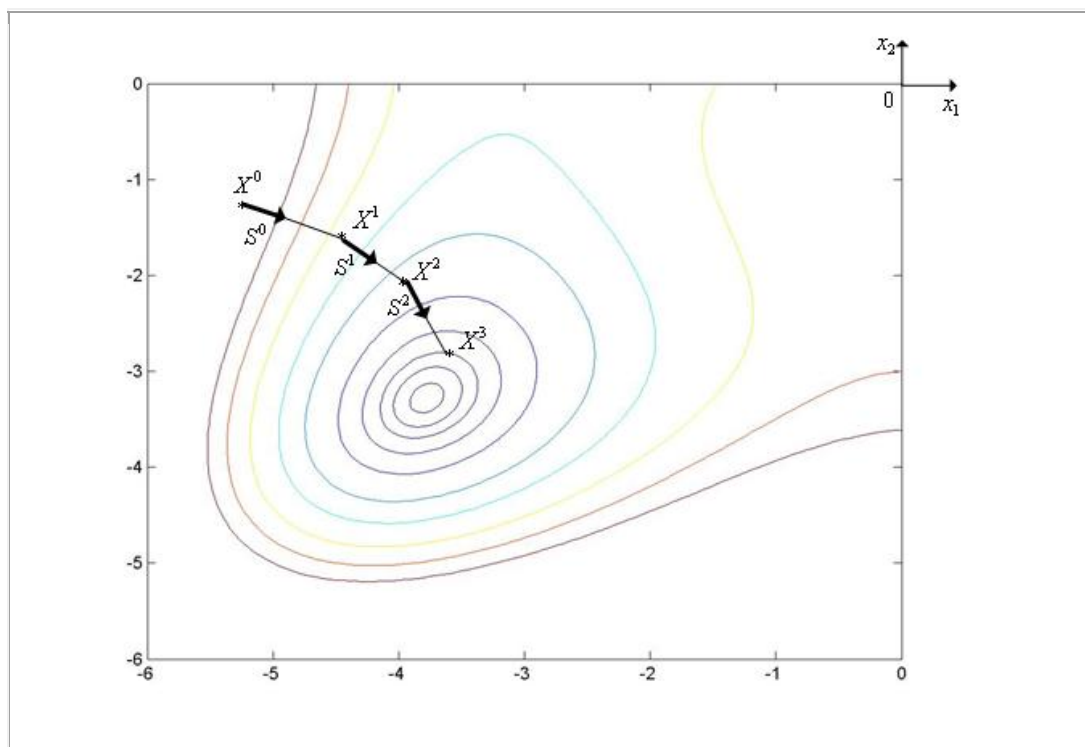


Рис. 3.7.2. Траектория поиска минимума функции Химмельблау градиентным методом дробления шага

Градиентный метод дробления шага иллюстрирует рис. 3.7.2, на котором показан фрагмент линий уровня функции Химмельблау.

Пример 3.7.1. Выполнить несколько итераций решения двумерной задачи локальной безусловной оптимизации $\min_{X \in R^2} \Phi(X) = \Phi(X^*) = \Phi^*$, где $\Phi(X) = \Phi(x, y) = x^2 + y^2 + 3(x + y)^2 = 4x^2 + 4y^2 + 6xy$, градиентным методом с дроблением шага, исходя из точки $X^0 = (x^0, y^0) = (-2.0, 1.0)$. Принять $\lambda^0 = 1.0$, $\nu = 0.7$, в качестве нормы вектора градиента использовать евклидову норму.

Решение. Итерация градиентного метода с дроблением шага для задачи имеет вид:

$$X^{r+1} = \begin{pmatrix} x^{r+1} \\ y^{r+1} \end{pmatrix} = \begin{pmatrix} x^r \\ y^r \end{pmatrix} + \lambda^r \begin{pmatrix} S_x^r \\ S_y^r \end{pmatrix}, \quad (3.7.9)$$

где

$$\begin{pmatrix} S_x^r \\ S_y^r \end{pmatrix} = S^r = -\frac{\nabla \Phi^r}{\|\nabla \Phi^r\|}, \quad \nabla \Phi^r = \begin{pmatrix} \frac{\partial \Phi(x^r, y^r)}{\partial x} \\ \frac{\partial \Phi(x^r, y^r)}{\partial y} \end{pmatrix}, \quad (3.7.10)$$

а величина шага λ^r находится из условия

$$\Phi(X^r) - \Phi(X^{r+1}) \geq 0.5\lambda^r \|\nabla \Phi^r\|. \quad (3.7.11)$$

Найдем явные выражения для частных производных нашей функции:

$$\frac{\partial \Phi(x, y)}{\partial x} = 8x + 6y; \quad \frac{\partial \Phi(x, y)}{\partial y} = 8y + 6x. \quad (3.7.12)$$

Таким образом, из (3.7.9), (3.7.10), (3.7.12) имеем искомую итерационную формулу градиентного метода с дроблением шага для нашей задачи

$$X^{r+1} = \begin{pmatrix} x^{r+1} \\ y^{r+1} \end{pmatrix} = \begin{pmatrix} x^r \\ y^r \end{pmatrix} + \lambda^r \begin{pmatrix} S_x^r \\ S_y^r \end{pmatrix}, \quad S_x^r = -\frac{8x^r + 6y^r}{\|\nabla \Phi^r\|}, \quad S_y^r = -\frac{8y^r + 6x^r}{\|\nabla \Phi^r\|},$$

$$\|\nabla \Phi(X)\| = \sqrt{(8x + 6y)^2 + (8y + 6x)^2}. \quad (3.7.13)$$

Первая итерация ($r=0$).

Из формул (3.7.12), (3.7.13) последовательно имеем:

$$X^1 = \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} = \begin{pmatrix} x^0 \\ y^0 \end{pmatrix} + \lambda^0 \begin{pmatrix} S_x^0 \\ S_y^0 \end{pmatrix},$$

$$\frac{\partial \Phi(x^0, y^0)}{\partial x} = 8x^0 + 6y^0 = 8 \cdot (-2) + 6 = -10,$$

$$\frac{\partial \Phi(x^0, y^0)}{\partial y} = 8y^0 + 6x^0 = 8 + 6 \cdot (-2) = -4,$$

$$\|\nabla\Phi^0\| = \sqrt{(-10)^2 + (-4)^2} \approx 10.77, \quad S_x^0 \approx \frac{10}{10.77} \approx 0.93, \quad S_y^0 \approx \frac{4}{10.77} \approx 0.37,$$

$$X^1 = \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} = \begin{pmatrix} x^0 \\ y^0 \end{pmatrix} + \lambda^0 \begin{pmatrix} S_x^0 \\ S_y^0 \end{pmatrix} \approx \begin{pmatrix} -2 \\ 1 \end{pmatrix} + 0.5 \begin{pmatrix} 0.93 \\ 0.37 \end{pmatrix} \approx \begin{pmatrix} -1.54 \\ 1.18 \end{pmatrix}.$$

Таким образом, $X^1 = \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} \approx \begin{pmatrix} -1.54 \\ 1.18 \end{pmatrix}$ (см. рис. 3.7.3).

Условие (3.7.11) на первой итерации имеет вид

$$\Phi(X^0) - \Phi(X^1) \geq 0.5\lambda^0 \|\nabla\Phi^0\|.$$

Поскольку

$$\Phi(X^0) = \Phi(x^0, y^0) = 4(x^0)^2 + 4(y^0)^2 + 6x^0y^0 = 4(-2)^2 + 4 + 6(-2) = 8,$$

$$\Phi(X^1) = \Phi(x^1, y^1) = 4(x^1)^2 + 4(y^1)^2 + 6x^1y^1 \approx 4(-1.54)^2 + 4(1.18)^2 + 6(-1.54)(1.18) \approx 4.16,$$

левая часть этого неравенства равна $8.0 - 4.16 = 3.84$. Его правая часть, легко видеть, равна $(0.5)^2 \cdot 10.77 \approx 2.69$.

Таким образом, на первой итерации условие (3.7.11) выполняется и величина шага λ должна быть изменена: $\lambda^1 = 0.7 \cdot 0.5 = 0.35$.

Вторая итерация ($r = 1$).

Аналогично первой итерации последовательно имеем:

$$X^2 = \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} = \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} + \lambda^1 \begin{pmatrix} S_x^1 \\ S_y^1 \end{pmatrix},$$

$$\frac{\partial\Phi(x^1, y^1)}{\partial x} = 8x^1 + 6y^1 = 8 \cdot (-1.54) + 6 \cdot 1.18 = -5.24,$$

$$\frac{\partial \Phi(x^1, y^1)}{\partial y} = 8y^1 + 6x^1 = 8 \cdot 1.18 + 6 \cdot (-1.54) = 0.20,$$

$$\|\nabla \Phi^1\| = \sqrt{(-5.24)^2 + (0.20)^2} \approx 5.24, \quad S_x^1 \approx \frac{5.24}{5.24} \approx 1.0, \quad S_y^1 \approx -\frac{0.2}{5.24} \approx -0.04,$$

$$X^2 = \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} = \begin{pmatrix} x^1 \\ y^1 \end{pmatrix} + \lambda^1 \begin{pmatrix} S_x^1 \\ S_y^1 \end{pmatrix} \approx \begin{pmatrix} -1.54 \\ 1.18 \end{pmatrix} + 0.35 \begin{pmatrix} 1.0 \\ 0.04 \end{pmatrix} \approx \begin{pmatrix} -1.19 \\ 1.22 \end{pmatrix}.$$

Таким образом, $X^2 = \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} \approx \begin{pmatrix} -1.19 \\ 1.22 \end{pmatrix}$ (см. рис. 3.7.3).

Условие (3.7.11) на второй итерации имеет вид

$$\Phi(X^1) - \Phi(X^2) \geq 0.5 \lambda^1 \|\nabla \Phi^1\|.$$

Поскольку $\Phi(X^1) \approx 4.16$,

$$\begin{aligned} \Phi(X^2) = \Phi(x^2, y^2) &= 4(x^2)^2 + 4(y^2)^2 + 6x^2y^2 \approx 4(-1.19)^2 + 4(1.22)^2 + \\ &+ 6(-1.19)(1.22) \approx 2.91, \end{aligned}$$

левая часть этого неравенства равна $4.16 - 2.91 = 1.25$. Его правая часть, легко видеть, равна $0.5 \cdot 0.35 \cdot 5.24 \approx 0.92$.

Таким образом, на второй итерации условие (3.7.11.) выполняется и величина шага λ должна быть изменена: $\lambda^2 = 0.7 \cdot 0.35 \approx 0.24$.

Третья итерация ($r = 2$).

Аналогично первой итерации последовательно имеем

$$X^3 = \begin{pmatrix} x^3 \\ y^3 \end{pmatrix} = \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} + \lambda^2 \begin{pmatrix} S_x^2 \\ S_y^2 \end{pmatrix},$$

$$\frac{\partial \Phi(x^2, y^2)}{\partial x} = 8x^2 + 6y^2 = 8 \cdot (-1.19) + 6 \cdot 1.22 = -2.20,$$

$$\frac{\partial \Phi(x^2, y^2)}{\partial y} = 8y^2 + 6x^2 = 8 \cdot 1.22 + 6 \cdot (-1.19) = 2.62,$$

$$\|\nabla \Phi^2\| = \sqrt{(-2.20)^2 + (2.62)^2} \approx 3.42, \quad S_x^2 \approx \frac{2.20}{3.42} \approx 0.64, \quad S_y^2 \approx -\frac{2.62}{3.42} \approx -0.77,$$

$$X^3 = \begin{pmatrix} x^3 \\ y^3 \end{pmatrix} = \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} + \lambda^2 \begin{pmatrix} S_x^2 \\ S_y^2 \end{pmatrix} \approx \begin{pmatrix} -1.19 \\ 1.22 \end{pmatrix} + 0.24 \begin{pmatrix} 0.64 \\ -0.77 \end{pmatrix} \approx \begin{pmatrix} -1.04 \\ 1.03 \end{pmatrix}.$$

Таким образом, $X^3 = \begin{pmatrix} x^3 \\ y^3 \end{pmatrix} \approx \begin{pmatrix} -1.04 \\ 1.03 \end{pmatrix}$ (см. рис. 3.7.3).

Условие (3.7.11) на третьей итерации имеет вид

$$\Phi(X^2) - \Phi(X^3) \geq 0.5\lambda^2 \|\nabla \Phi^2\|.$$

Поскольку $\Phi(X^2) \approx 2.91$,

$$\begin{aligned} \Phi(X^3) = \Phi(x^3, y^3) &= 4(x^3)^2 + 4(y^3)^2 + 6x^3y^3 \approx 4(-1.04)^2 + 4(1.03)^2 + \\ &+ 6(-1.04)(1.03) \approx 2.14, \end{aligned}$$

левая часть этого неравенства равна $2.91 - 2.14 = 0.77$. Его правая часть, легко видеть, равна $0.5 \cdot 0.24 \cdot 3.42 \approx 0.41$.

Таким образом, на третьей итерации условие (3.7.11) выполняется и величина шага λ должна быть изменена: $\lambda^3 = 0.7 \cdot 0.24 \approx 0.17$.

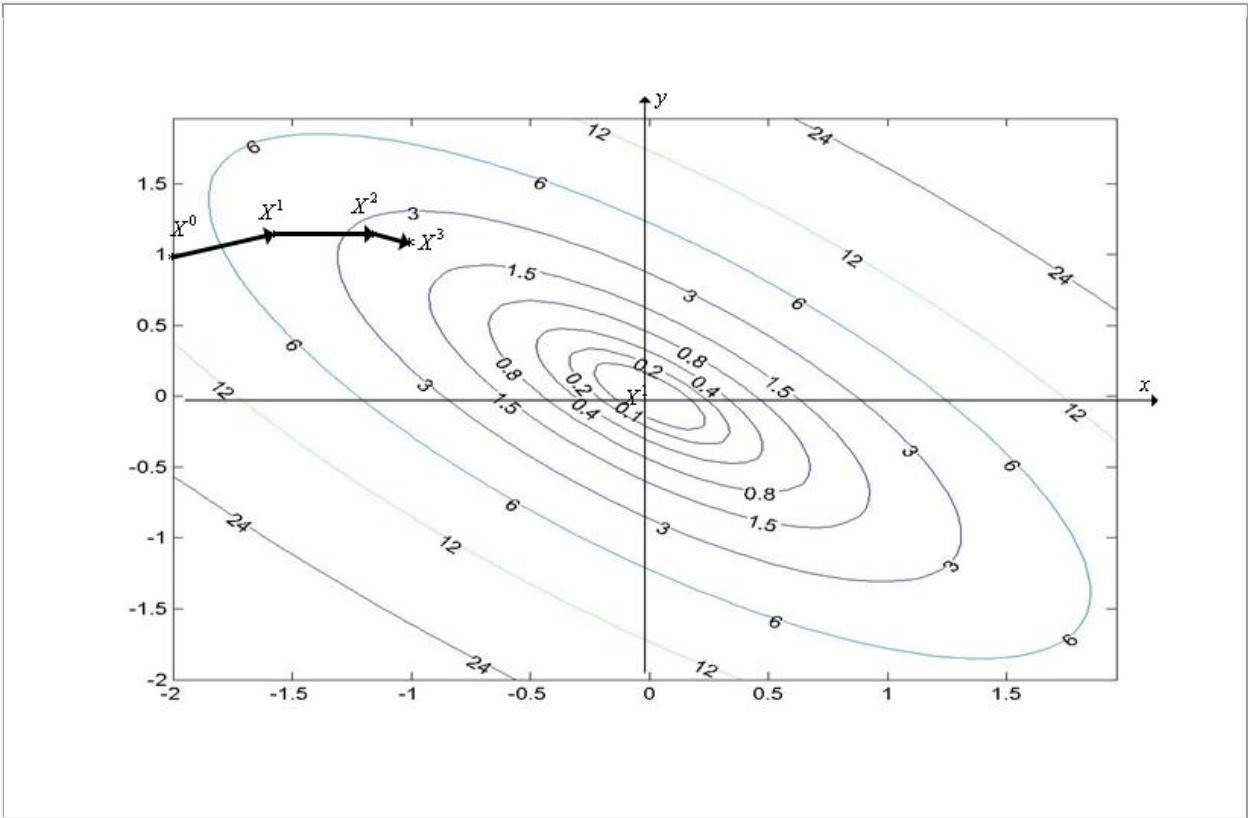


Рис. 3.7.3. Фрагмент (три итерации) траектории поиска минимума функции $\Phi(X) = \Phi(x, y) = x^2 + y^2 + 3(x + y)^2 = 4x^2 + 4y^2 + 6xy$ градиентным методом с дроблением шага, исходя из точки $X^0 = (x^0, y^0) = (-2.0, 1.0)$

3.8 Метод оптимизации Ньютона

Положим, что функция $\Phi(X)$ всюду дважды дифференцируема в n -мерном евклидовом пространстве R^n . Рассмотрим следующую многомерную задачу локальной безусловной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного в n -мерном евклидовом пространстве R^n ,

$$\min_{X \in R^n} \Phi(X) = \Phi(X^*) = \Phi^* .$$

Рассмотрим первые три члена разложения функции $\Phi(X)$ в ряд Тейлора в окрестности точки X^r :

$$\begin{aligned} \Phi(X) \approx \tilde{\Phi}^r(X) = & \Phi(X^r) + (\nabla \Phi(X^r), (X - X^r)) + \\ & + \frac{1}{2} (H(X^r)(X - X^r), (X - X^r)). \end{aligned} \quad (3.8.1)$$

Здесь $H(X)$ – матрица Гессе функции $\Phi(X)$. Из (3.8.1) следует, что градиент функции $\tilde{\Phi}^r(X)$ равен

$$\nabla \tilde{\Phi}^r(X) = \nabla \Phi(X^r) + H(X^r)(X - X^r). \quad (3.8.2)$$

Если матрица Гессе $H(X^r)$ положительно определена, то функция $\tilde{\Phi}^r(X)$ достигает минимума в точке, в которой градиент этой функции равен нулевому вектору.

Таким образом, в точке X^{r+1} минимума функции $\tilde{\Phi}^r(X)$ справедливо равенство:

$$\nabla \Phi(X^r) + H(X^r)(X^{r+1} - X^r) = 0, \quad (3.8.3)$$

где 0 – n -мерный вектор нулей.

Отсюда получаем итерационную формулу

$$X^{r+1} = X^r - H^{-1}(X^r) \nabla \Phi(X^r) = X^r + \Delta^r \quad (3.8.4)$$

для отыскания очередного приближения к точке минимума функции $\Phi(X)$.

Здесь

$$\Delta^r = -H^{-1}(X^r) \nabla \Phi(X^r). \quad (3.8.5)$$

Выражение (3.8.4) представляет собой итерационную формулу решения системы уравнений (3.8.3) широко известным методом касательных (методом Ньютона) – см. параграф 2.6. Этим фактом объясняется название рассматриваемого метода оптимизации.

Найдем скалярное произведение градиента функции $\Phi(X)$ в точке X^r и вектора Δ^r :

$$(\nabla \Phi(X^r), \Delta^r) = (\nabla \Phi(X^r), H^{-1}(X^r) \nabla \Phi(X^r)) < 0. \quad (3.8.6)$$

Последнее неравенство справедливо в силу постулируемой положительной определенности матрицы Гессе в точке X^r . Геометрически неравенство (3.8.6)

означает, что вектор Δ^r образует тупой угол с градиентом целевой функции $\Phi(X)$ в точке X^r (см. рис. 3.8.1).

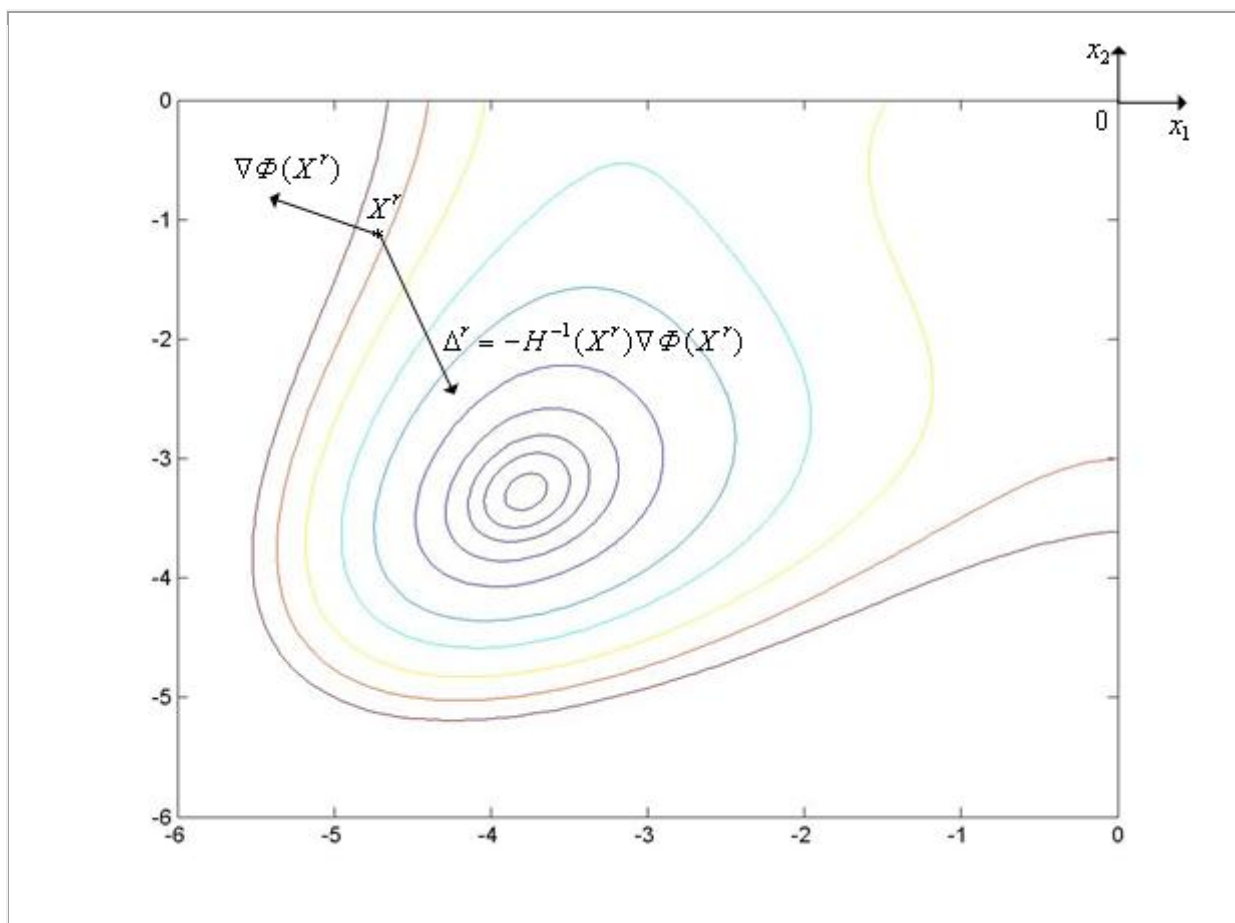


Рис. 3.8.1. К обоснованию многомерной оптимизации Ньютона

Таким образом, при минимизации овражных функций вектор Δ^r может составлять с осью оврага меньший угол, чем вектор антиградиента. Эта особенность делает метод оптимизации Ньютона, вообще говоря, более эффективным, чем градиентный метод наискорейшего спуска.

Отметим трудности, которые могут возникать при использовании итерационной формулы (3.8.4):

- Если размерность пространства R^n велика, то обращение на каждой итерации матрицы Гессе $H(X^r)$ может потребовать значительных вычислительных ресурсов;
- Значение минимизируемой функции $\Phi(X)$ в точке $X^{r+1} = X^r + \Delta^r$ может превышать значение функции в предыдущей точке X^r вследствие того, что направление Δ^r ведет к уменьшению $\Phi(X)$, но величина шага слишком велика;

- Направление спуска, определяемое вектором $\Delta^r = -H^{-1}(X^r)\nabla\Phi(X^r)$, ведет к убыванию целевой функции только при положительной определенности матрицы Гессе $H(X^r)$. Это приводит к тому, что на каждой итерации необходимы вычислительные затраты на проверку обусловленности этой матрицы. Указанная матрица может быть плохо обусловленной. Более того, указанная матрица может быть вырожденной и, поэтому, не иметь обратной матрицы.

Вследствие этих трудностей итерационная формула (3.8.4) в «чистом» виде не используется в вычислительной практике.

Для того чтобы избежать обращения матрицы Гессе, на практике вектор Δ^r находят обычно из следующей системы линейных алгебраических уравнений (СЛАУ), вытекающей из равенства (3.8.5):

$$H(X^r)\Delta^r = -\nabla\Phi(X^r). \quad (3.8.7)$$

СЛАУ (3.8.7) может быть решена различными численными методами (например, прямыми методами, итерационными методами).

Величина шага в направлении Δ^r , которая приводит к убыванию функции $\Phi(X)$, может быть обеспечена путем добавления в итерационную формулу (3.8.4) коэффициента $\lambda^r \in (0,1]$, т.е. путем использования вместо формулы (3.8.4) итерационной формулы

$$X^{r+1} = X^r - \lambda^r H^{-1}(X^r)\nabla\Phi(X^r) = X^r + \lambda^r \Delta^r, \quad (3.8.8)$$

где коэффициент λ^r выбирают тем или иным способом так, чтобы обеспечить условие $\Phi(X^{r+1}) < \Phi(X^r)$.

Для того, чтобы направление спуска независимо от определенности матрицы Гессе $H(X^r)$ вело к убыванию функции $\Phi(X)$, в качестве вектора Δ^r можно использовать вектор

$$\Delta^r = -\left(\mu^r I + H(X^r)\right)^{-1} \nabla\Phi(X^r), \quad (3.8.9)$$

где $I - (n \times n)$ – единичная матрица, а $\mu^r > 0$ – параметр, выбираемый так, чтобы матрица $\mu^r I + H(X^r)$ являлась положительно определенной.

Схема метода

Рассмотрим схему одной из модификаций метода оптимизации Ньютона, в которой используется итерационная формула (3.8.8) и вектор Δ^r находят путем решения на каждой итерации СЛАУ (3.8.7).

1. Задаем начальную точку X^0 , начальную величину шага $\lambda^0 = 1$ и коэффициент дробления шага $\nu \in (0, 1]$. Полагаем счетчик числа итераций $r = 0$.
2. Вычисляем в точке X^r вектор градиента $\nabla \Phi(X^r)$ и матрицу Гессе $H(X^r)$.
3. Решаем СЛАУ (3.8.7) и находим вектор Δ^r .
4. По формуле (3.8.8) вычисляем компоненты вектора X^{r+1} .
5. Вычисляем величину $\Phi(X^{r+1})$ – значение функции $\Phi(X)$ в точке X^{r+1} .
6. Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем $X^* \approx X^{r+1}$, и завершаем итерации. Иначе – переходим к следующему пункту.
7. Если $\Phi(X^{r+1}) < \Phi(X^r)$, то полагаем $r = r + 1$ и переходим к п.2. Иначе – фиксированное число раз полагаем $\lambda^r = \nu \lambda^r$ и переходим к пункту 4.

В качестве условия окончания поиска можно использоваться одно из стандартных условий окончания итераций:

$$\|X^{r+1} - X^r\| = \lambda^r \leq \varepsilon_X, \quad (3.8.10)$$

$$|\Phi(X^{r+1}) - \Phi(X^r)| \leq \varepsilon_\Phi \quad (3.8.11)$$

или условие

$$\|\nabla \Phi^r\| \leq \varepsilon_\nabla, \quad (3.8.12)$$

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

3.9 Метод с возвратом при неудачном шаге. Метод наилучшей пробы

Рассматривается следующая многомерная задача локальной безусловной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного в n -мерном евклидовом пространстве R^n , $\min_{X \in R^n} \Phi(X) = \Phi(X^*) = \Phi^*$.

При решении задачи методом с возвратом при неудачном шаге (одношаговый метод оптимизации) используется итерационная формула:

$$X^{r+1} = X^r + \lambda^r \frac{\psi^r}{\|\psi^r\|}, \quad (3.9.1)$$

где λ^r – величина шага на r -ой итерации, $\psi^r = (\psi_1^r, \psi_2^r, \dots, \psi_n^r)$ – реализация n -мерного случайного вектора, $\|\cdot\|$ – некоторая векторная норма.

Обычно в качестве координат вектора ψ^r используют независимые случайные величины, равномерно распределенные в интервале $[-1, 1]$.

Схема метода с возвратом при неудачном шаге

1. Задаем начальную точку X^0 , начальную длину шага λ^0 и полагаем счетчик числа итераций $r = 0$.
2. Задаем начальное значение счетчика числа неудачных попыток $k = 1$.
3. Получаем реализацию случайных чисел $\psi_1^r, \psi_2^r, \dots, \psi_n^r$ – компонент вектора ψ^r и по формуле (3.9.1) находим пробную точку X^{r+1} .
4. Вычисляем значение $\Phi(X^{r+1})$ функции $\Phi(X)$ в точке X^{r+1} .
5. Если $\Phi(X^{r+1}) < \Phi(X^r)$, то полагаем $r = r + 1$ и переходим к п.3. Иначе – переходим к п.6.
6. Полагаем $k = k + 1$. Если $k < K$, то переходим к п.3. Иначе – переходим к п.7. Здесь K – предельное количество неудачных попыток (свободный параметр метода). Рекомендуется $K = 3n$.
7. Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то полагаем $X^* \approx X^{r+1}$ и завершаем итерации. Иначе – по-

лагаем $r=r+1$, $\lambda^r = \alpha\lambda^{r-1}$ и переходим к п.2. Здесь $\alpha \in (0,1)$ – коэффициент уменьшения шага (свободный параметр метода).

В качестве условия окончания поиска можно использоваться одно из стандартных условий окончания итераций:

$$\|X^{r+1} - X^r\| = \lambda^r \leq \varepsilon_X, \quad (3.9.2)$$

где ε_X – константа, определяющая требуемую точность решения по X ;

$$|\Phi(X^{r+1}) - \Phi(X^r)| \leq \varepsilon_\Phi, \quad (3.9.3)$$

где ε_Φ – константа, определяющая требуемую точность решения по Φ .

Метод с возвратом при неудачном шаге иллюстрирует рис. 3.9.1, на котором показан фрагмент линий уровня функции Химмельблау.

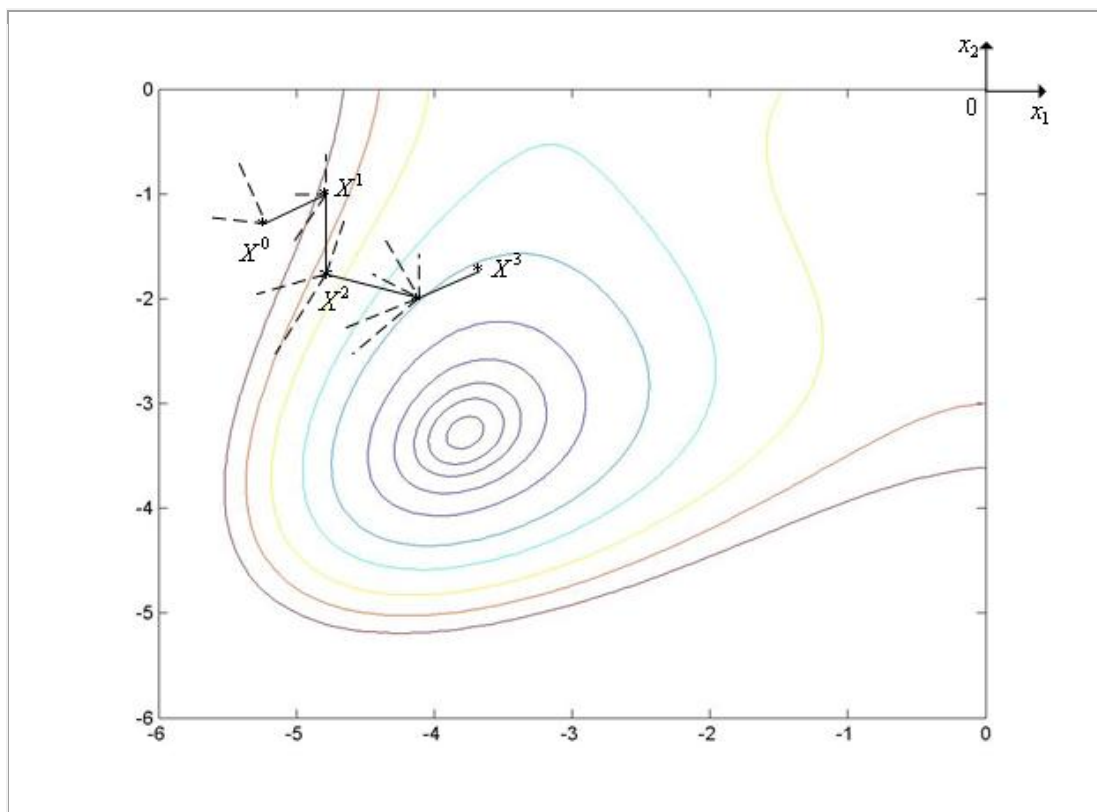


Рис. 3.9.1. Траектория поиска минимума функции Химмельблау методом с возвратом при неудачном шаге. Пунктиром на рисунке показаны неудачные шаги

Модификацией метода с возвратом при неудачном шаге является метод наилучшей пробы (также одношаговый метод оптимизации).

Схема метода наилучшей пробы

1. Задаем начальную точку X^0 , начальную длину шага λ^0 и полагаем счетчик числа итераций $r=0$.
2. Генерируем M случайных векторов ψ_i^r , $i \in [1, \dots, M]$ и по формуле (3.9.1) находим пробные точки X_i^{r+1} , $i \in [1, \dots, M]$.
3. Вычисляем значения $\Phi(X_i^{r+1})$ функции $\Phi(X)$ в пробных точках X_i^{r+1} , $i \in [1, \dots, M]$ и находим минимальное из этих значений $\Phi(X^{r+1}) = \Phi(X_k^{r+1}) = \min_{i \in [1, \dots, M]} \Phi(X_i^{r+1})$.
4. Если $\Phi(X^{r+1}) < \Phi(X^r)$, то полагаем $r=r+1$ и переходим к п.2. Иначе – переходим к п.5.
5. Проверяем условие окончания поиска (см. (3.9.2), (3.9.3)). Если условие окончания поиска выполнено, то полагаем $X^* \approx X^{r+1}$ и завершаем итерации. Иначе – полагаем $r=r+1$, $\lambda^r = \alpha \lambda^r$ и переходим к п.2. Здесь $\alpha \in (0, 1)$ – коэффициент уменьшения шага (свободный параметр метода).

Метод наилучшей пробы иллюстрирует рис. 3.9.2, на котором показан фрагмент линий уровня функции Химмельблау.

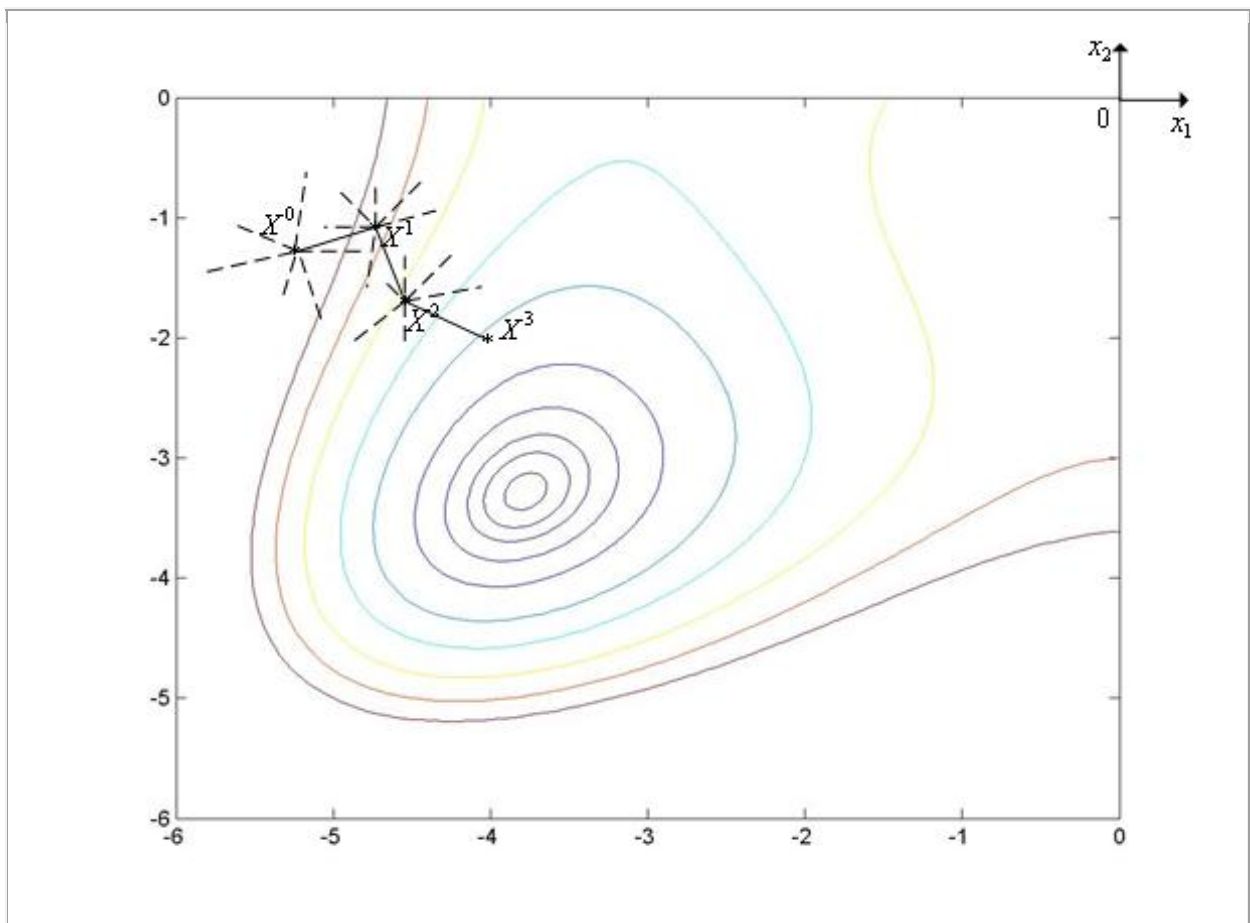


Рис. 3.9.2. Траектория поиска минимума функции Химмельблау методом наилучшей пробы. Пунктиром на рисунке показаны неудачные пробы

3.10 Метод комплексов

Рассматривается следующая многомерная задача локальной безусловной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного в n -мерном евклидовом пространстве R^n , $\min_{X \in R^n} \Phi(X) = \Phi(X^*) = \Phi^*$.

Комплексом называется многогранник с $N > n+1$ вершинами (не обязательно выпуклый). Рекомендуются $N = 2n$. Вообще говоря, комплексом в комбинаторной топологии называется геометрическая фигура, которая может быть разбита на более элементарные фигуры. В нашем случае такими элементарными фигурами являются симплексы. Поэтому, говоря более строго, в данном параграфе рассматриваются **симплициальные комплексы**.

При решении **методом комплексов** используются следующие операции:

- генерация случайного комплекса;
- отражение вершины комплекса с растяжением;
- сжатие комплекса.

Генерация случайного комплекса

В пространстве R^n координаты вершин случайного комплекса с N вершинами могут быть найдены по формуле

$$X_i = X_0 + l \frac{\psi_i}{\|\psi_i\|}, \quad i \in [0, \dots, N], \quad (3.10.1)$$

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

Отражение вершины комплекса с растяжением

Положим, что в пространстве R^n тем или иным способом задан комплекс C^r с N вершинами X_i^r , $i \in [1, \dots, N]$, и его вершину X_k^r необходимо отра-

зять через центр тяжести комплекса с растяжением. В новом комплексе C^{r+1} все вершины, кроме k -ой, совпадают с соответствующими вершинами исходного комплекса C^r , а k -я вершина находится на прямой, проходящей через центр тяжести этого комплекса и его вершину X_k^r (см. рис. 3.10.1).

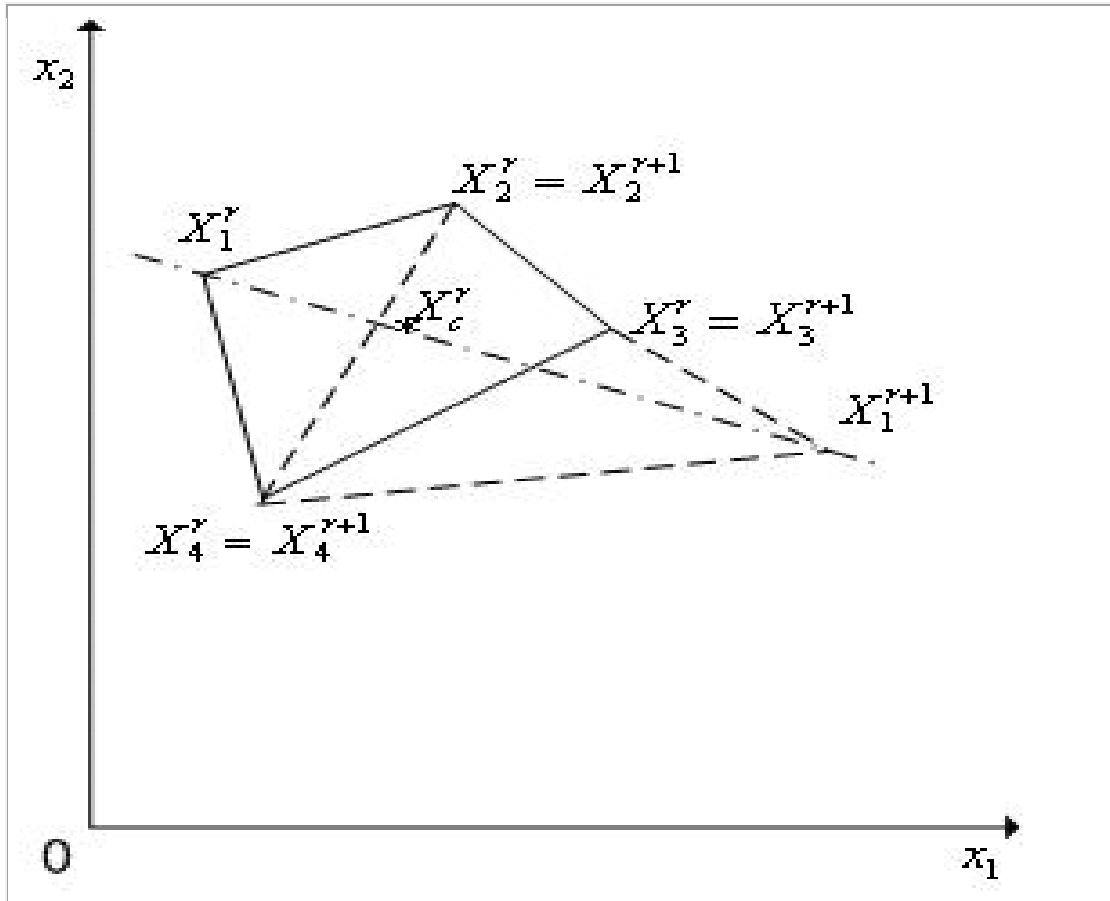


Рис. 3.10.1. Отражение вершины X_1^r комплекса C^r через центр его тяжести с растяжением. Пунктиром показан новый комплекс C^{r+1}

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

Тогда имеем

$$X_i^{r+1} = X_i^r, \quad i \in [1, \dots, N], \quad i \neq k; \quad X_k^{r+1} = X_c^r + \alpha (X_c^r - X_k^r), \quad (3.10.2)$$

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

$$X_c^r = \frac{1}{N} \sum_{i=1}^N X_i^r. \quad (3.10.3)$$

Сжатие комплекса

Положим, что в пространстве R^n тем или иным способом задан комплекс C^r с N вершинами X_i^r , $i \in [1, \dots, N]$, и его вершину X_k^r необходимо переместить ближе к центру тяжести комплекса C^r – выполнить сжатие комплекса. В новом комплексе C^{r+1} все вершины, кроме k -ой, совпадают с соответствующими вершинами исходного комплекса C^r , а k -я вершина находится на прямой, проходящей через центр тяжести этого комплекса и его вершину X_k^r (см. рис. 3.10.2).

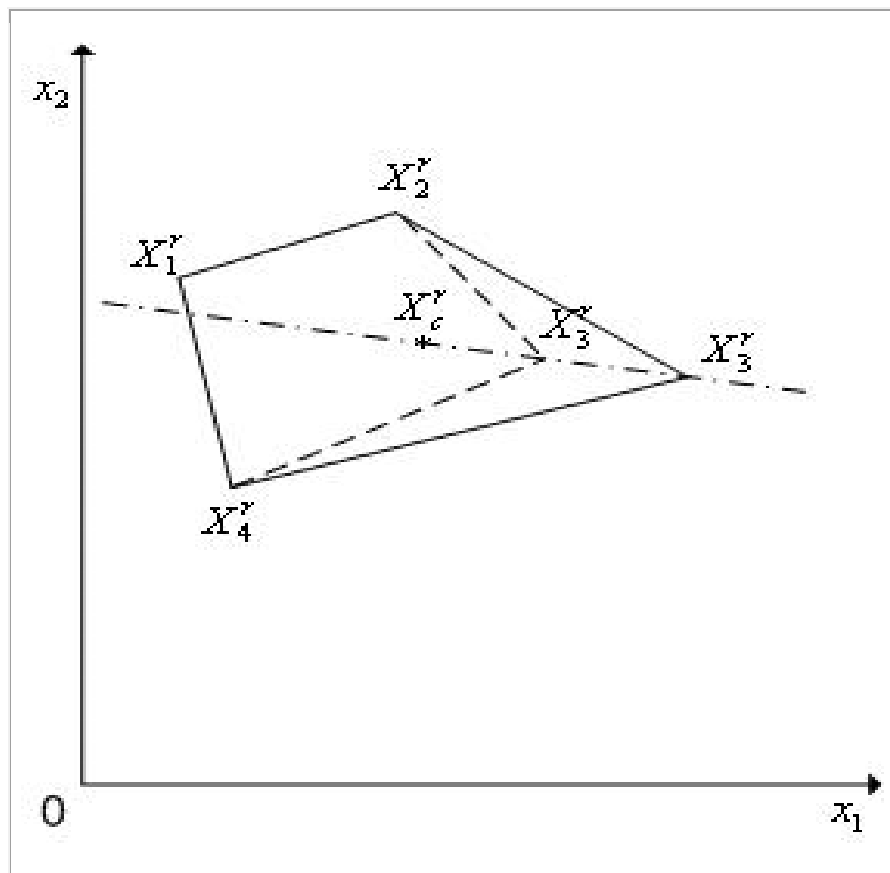


Рис. 3.10.2. Сжатие комплекса C^r . Пунктиром показан новый комплекс C^r

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

Тогда имеем:

$$X_i^{r+1} = X_i^r, \quad i \in [1, \dots, N], \quad i \neq k; \quad X_k^{r+1} = X_c^r + \beta(X_k^r - X_c^r), \quad (3.10.4)$$

где β – коэффициент сжатия комплекса (рекомендуемое значение – 2), X_c^r – вектор координат центра тяжести комплекса C^r (см. (3.10.3)).

Схема простейшего варианта метода комплексов

1. Задаем начальную точку X_0^0 , исходя из которой должен быть построен комплекс C^0 , начальное значение величины $l=l^0$ и полагаем счетчик числа итераций $r=0$.
2. Генерируем N случайных векторов ψ_i^r , $i \in [1, \dots, N]$ и по формуле (3.10.1) находим координаты X_i^r , $i \in [1, \dots, N]$ вершин комплекса C^r .
3. Вычисляем значения $\Phi(X_i^r)$ функции $\Phi(X)$ во всех вершинах комплекса C^r .
4. Находим максимальное из значений $\Phi(X)$ в вершинах комплекса C^r

$$\Phi(X_k^r) = \max_{i \in [1, \dots, N]} \Phi(X_i^r).$$

5. По формулам (3.10.2), (3.10.3) для комплекса C^r выполняем отражение вершины комплекса с растяжением X_k^r – получаем вершину X_k^{r+1} и новый комплекс C^{r+1} .
6. Вычисляем значение $\Phi(X_k^{r+1})$ функции $\Phi(X)$ в вершине X_k^{r+1} .
7. Если $\Phi(X_k^{r+1}) < \Phi(X_k^r)$, то полагаем $r=r+1$ и переходим к п.8. Иначе – по формулам (3.10.4), (3.10.3) выполняем сжатие симплекса C^{r+1} в направлении $X_k^{r+1} - X_c^{r+1}$, полагаем $r=r+2$ и переходим к п.6.
8. Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то в качестве точки X^* полагаем вершину комплекса C^r , к которой функция $\Phi(X)$ имеет наименьшее значение и завершаем итерации. Иначе – переходим к п. 4.

В качестве критерия окончания поиска может использоваться следующее условие: максимальная длина ребра комплекса C^r не превышает ε_X – требуемую точность решения по X .

Может использоваться также следующее аналогичное условие: максимальная разность значений функции $\Phi(X)$ в двух вершинах комплекса C^r не превышает ε_Φ – требуемую точность решения по Φ .

Могут использоваться также более сложные условия окончания поиска:

$$\frac{1}{N} \sqrt{\sum_{i=1}^N \|X_i^r - X_i^r\|} \leq \varepsilon_X, \quad (3.10.5)$$

$$\frac{1}{N} \sqrt{\sum_{i=1}^N \|\Phi(X_i^r) - \Phi_c^r\|} \leq \varepsilon_\Phi. \quad (3.10.6)$$

В формуле (3.10.5) векторная норма означает расстояние вершины X_i^r до центра тяжести комплекса C^r , а сама формула (3.10.5) определяет среднее расстояние вершин комплекса C^r до его центра тяжести.

В формуле (3.10.6) $\Phi_c^r = \frac{1}{N} \sum_{i=1}^N \Phi(X_i^r)$ есть среднее значение функции $\Phi(X)$ в вершинах комплекса C^r , а сама формула (3.10.6) определяет среднее отклонение значений функции $\Phi(X)$ в вершинах комплекса C^r от этого среднего значения.

Метод комплексов иллюстрирует рис. 3.10.3, на котором показан фрагмент линий уровня функции Химмельблау. На рисунке исходный комплекс C^r имеет вершины X_i^r , $i \in [1, \dots, 4]$. После отражения с растяжением вершины X_1^r этого комплекса, в которой функция имеет максимальное значение, получаем комплекс C^{r+1} с вершинами X_i^{r+1} , $i \in [1, \dots, 4]$. После отражения с растяжением вершины X_2^r комплекса C^{r+1} , в которой функция имеет максимальное значение, получаем комплекс C^{r+2} с вершинами X_i^{r+2} , $i \in [1, \dots, 4]$.

Известно множество модификаций рассмотренного метода комплексов, направленных, в частности, на преодоление «уплощения» комплекса в процессе поиска.

С этой целью через фиксированное количество итераций находят максимальная и минимальная диагонали комплекса и, если их отношение превышает заданное, то по рассмотренной схеме производится построение нового комплекса.

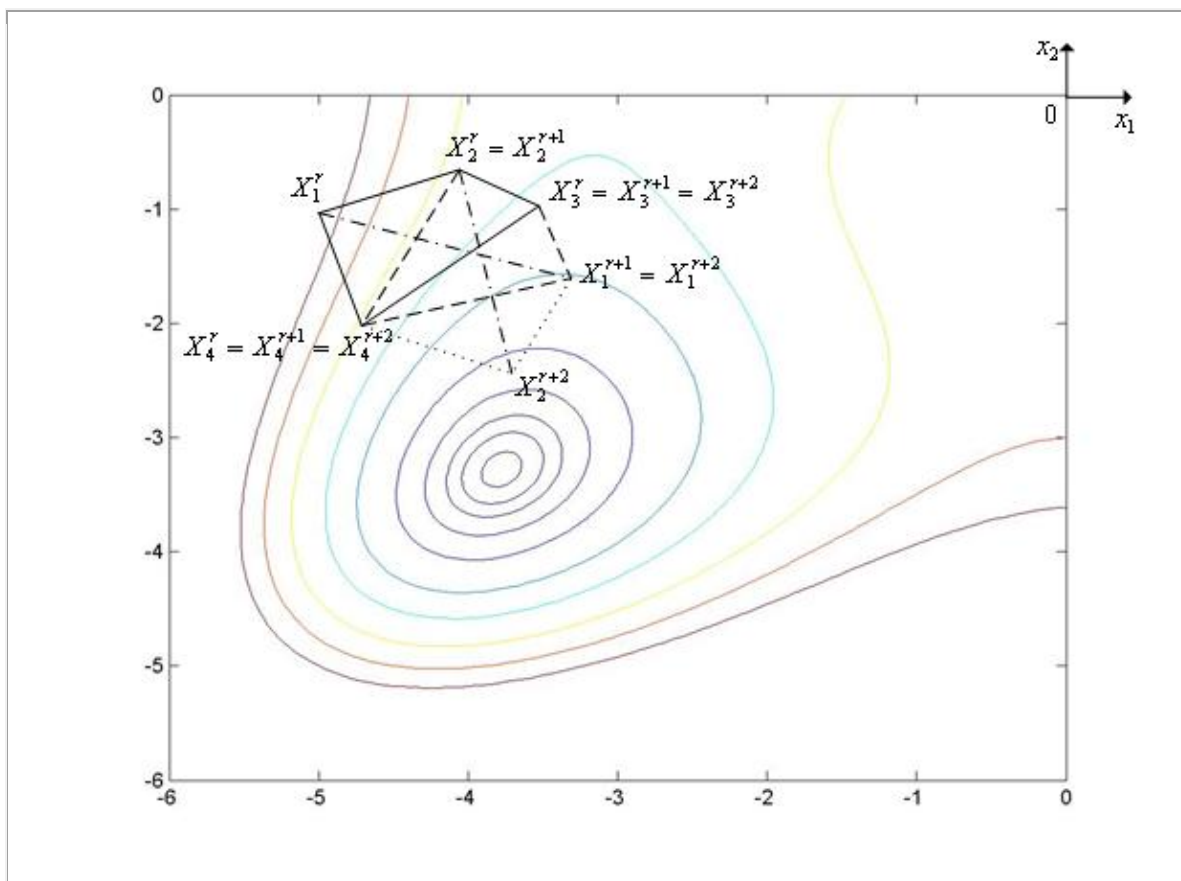


Рис. 3.10.3. Траектория поиска минимума функции Химмельблау методом комплексов

3.11 Метод повторяющегося случайного поиска

Рассматривается следующая многомерная задача локальной безусловной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного в n -мерном евклидовом пространстве R^n , $\min_{X \in R^n} \Phi(X) = \Phi(X^*) = \Phi^*$.

В методе повторяющегося случайного поиска (3-шаговый метод) используется итерационная схема (см. рис. 3.11.1)

$$X^{r+1} = X^r + \lambda^r \Delta^r, \quad (3.11.1)$$

где λ^r – величина шага (скаляр) на r -ой итерации, Δ^r – $(n \times 1)$ – вектор, определяющий направление шага на r -ой итерации:

$$\Delta^r = \left[\beta \frac{S^r}{\|S^r\|} + (1-\beta) \frac{P^r}{\|P^r\|} \right]. \quad (3.11.2)$$

Здесь $S^r = \gamma S^{r-1} + (1-\gamma) S^{r-2}$ – вектор «предыстории», определяющий среднее направление поиска на двух предыдущих шагах; $\|*\|$ – некоторая векторная норма; P^r – n -мерный вектор псевдослучайных чисел, равномерно распределенных в $[0,1]$; скаляр $\beta \in [0,1]$ – коэффициент, задающий относительные веса детерминированной и случайной компонент в векторе Δ^r (свободный параметр метода); скаляр $\gamma \in [0,1]$ – коэффициент, задающий относительные веса векторов S^{r-1}, S^{r-2} в векторе S^r (свободный параметр метода).

Заметим, что отношение $\frac{S^r}{\|S^r\|}$ представляет собой единичный вектор направления S^r , а отношение $\frac{P^r}{\|P^r\|}$ – единичный вектор направления P^r .

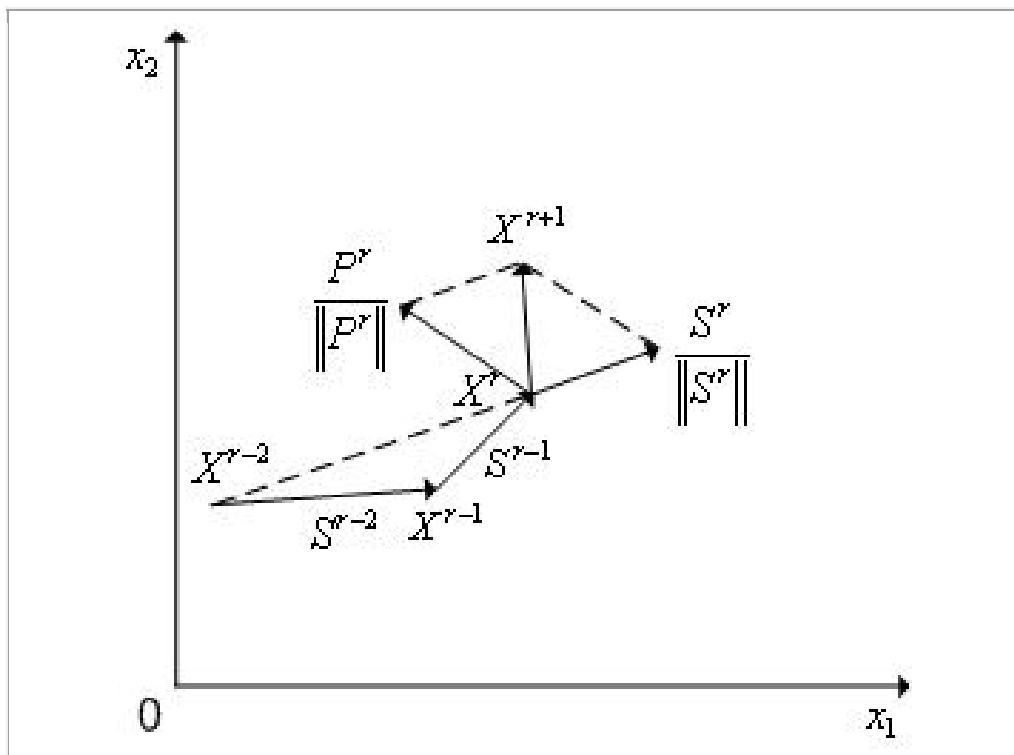


Рис. 3.11.1. К итерационной схеме метода повторяющегося случайного поиска

Принято $\gamma = 0.5$, $\beta = 0.5$, $\lambda^r = 2$, так что $S^r = 0.5(S^{r-1} + S^{r-2})$ и

$$X^{r+1} = X^r + \left[\frac{S^r}{\|S^r\|} + \frac{P^r}{\|P^r\|} \right].$$

Упрощенная схема метода повторяющегося случайного поиска

1. Задаем начальную точку X^0 , начальный шаг λ^2 , значения коэффициентов β, γ и полагаем счетчик числа итераций $r = 2$.
2. Тем или иным способом, например, с помощью одношагового метода наилучшей пробы определяем точки X^1, X^2 – этап «разгона» метода.
3. Генерируем n -мерный случайный вектор P^r и по формулам (3.11.1), (3.11.2) вычисляем координаты точки X^{r+1} и значение $\Phi(X^{r+1})$ функции $\Phi(X)$ в этой точке.
4. Если $\Phi(X^{r+1}) < \Phi(X^r)$, то проверяем условие окончания итераций (см. ниже). Если условие окончания выполнено, то полагаем $X^* \approx X^{r+1}$ и завершаем итерации. Если условие окончания итераций не выполнено, то некоторому правилу увеличиваем длину шага λ^r , например, полагая $\lambda^{r+1} = 2\lambda^r$, принимаем $r = r + 1$ и переходим к п.3. Если $\Phi(X^{r+1}) \geq \Phi(X^r)$, то переходим к п. 5.
5. Некоторое фиксированное количество раз делаем попытку, исходя из той же точки X^r , не меняя длины шага λ^r , добиться уменьшения значения функции $\Phi(X)$ путем только изменения вектора P^r , т.е., не меняя X^r и λ^r , переходим на п. 3. Если это фиксированное количество попыток не привело к успеху, то, исходя из той же точки X^r , по некоторому правилу уменьшаем длину шага λ^r , например, полагая $\lambda^r = \frac{\lambda^r}{2}$, и переходим к п.3.

В качестве условия окончания поиска можно использоваться одно из стандартных условий окончания итераций:

$$\|X^{r+1} - X^r\| \leq \varepsilon_X, \quad (3.11.3)$$

где ε_X – константа, определяющая требуемую точность решения по X ;

$$\left| \Phi(X^{r+1}) - \Phi(X^r) \right| \leq \varepsilon_\Phi, \quad (3.11.4)$$

где ε_Φ – константа, определяющая требуемую точность решения по Φ .

Известно множество модификаций рассмотренной простейшей схемы метода повторяющегося случайного поиска. Например, в процессе поиска могут изменяться по некоторым правилам не только длина шага λ^r , но и коэффициенты β, γ .

Метод повторяющегося случайного поиска иллюстрирует рис. 3.11.2, на котором показан фрагмент линий уровня функции Химмельблау.

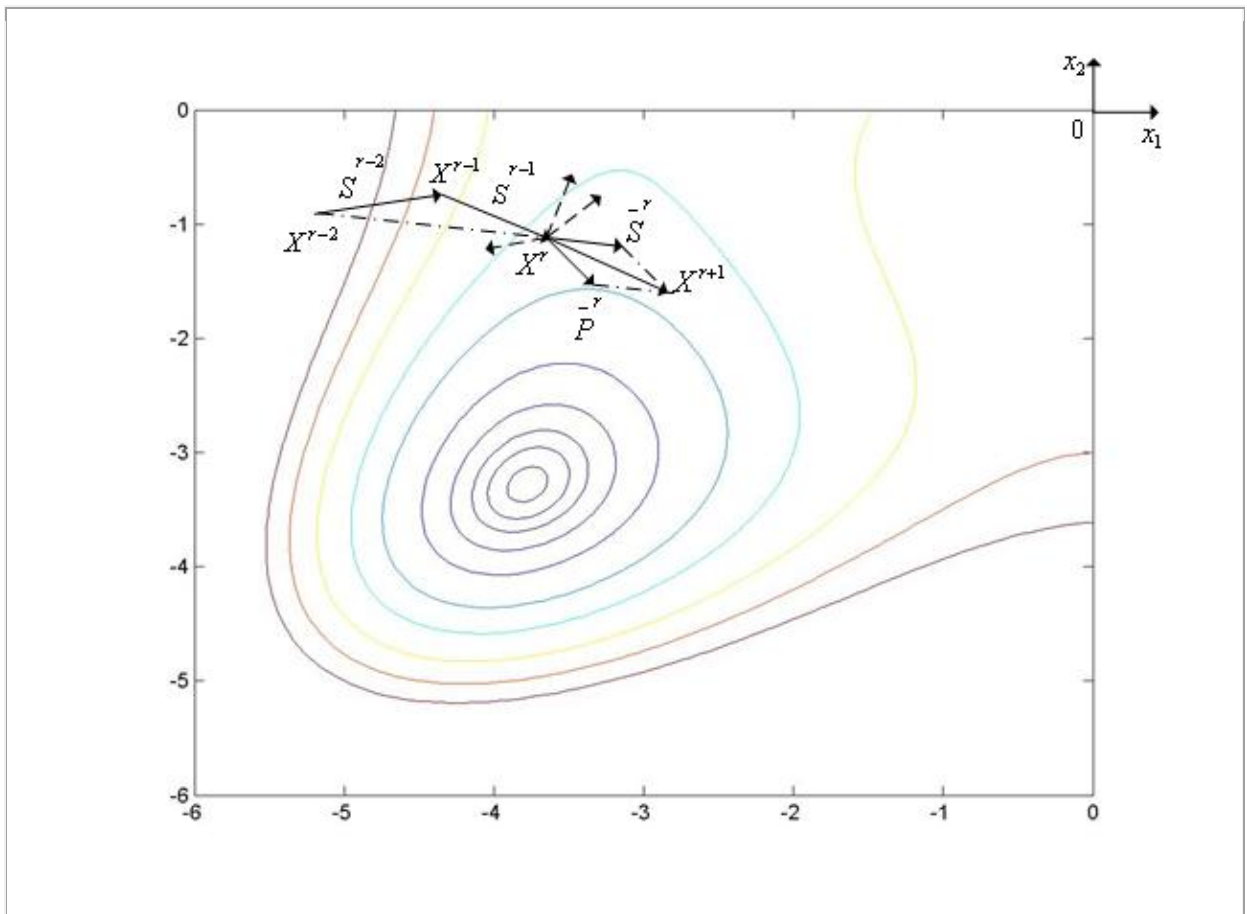


Рис. 3.11.2. Траектория поиска минимума функции Химмельблау методом повторяющегося случайного поиска

На рисунке принято $\tilde{S}^r = \frac{S^r}{\|S^r\|}$, $\tilde{P}^r = \frac{P^r}{\|P^r\|}$, $\gamma = 0.5$, $\beta = 0.5$, $\lambda^r = 2$, так что $S^r = 0.5(S^{r-1} + S^{r-2})$ и $X^{r+1} = X^r + [\tilde{S}^r + \tilde{P}^r]$. Пунктиром показаны отвергнутые векторы.

3.12 Метод случайного поиска с постоянным радиусом поиска и случайными направлениями

Рассматривается следующая многомерная задача локальной безусловной оптимизации: найти минимум критерия оптимальности $\Phi(X)$, определенного в n -мерном евклидовом пространстве R^n , $\min_{X \in R^n} \Phi(X) = \Phi(X^*) = \Phi^*$.

Метод случайного поиска с постоянным радиусом поиска и случайными направлениями использует процедуру генерации случайных точек, равномерно распределенных по поверхности гиперсферы в пространстве R^n . Пусть $X^c = (x_1^c, x_2^c, \dots, x_n^c)$ – вектор координат центра гиперсферы, ρ – радиус гиперсферы, R – вектор с началом в точке X^c и концом в искомой точке на поверхности гиперсферы, $\alpha_i, i \in [1, \dots, n]$ – углы между вектором R и осями координатных осей $Ox_i, i \in [1, \dots, n]$.

Во введенных обозначениях схема алгоритма генерации случайных точек, равномерно распределенных по поверхности гиперсферы радиуса ρ , может быть представлена в следующем виде:

- генерируем n случайных чисел, равномерно распределенных в интервале $[0, 2\pi)$;
- вычисляем направляющие косинусы $\cos \alpha_i, i \in [1, \dots, n]$ вектора R ;
- находим координаты искомой точки $x_i = x_i^c + \rho \cos \alpha_i, i \in [1, \dots, n]$.

Упрощенная схема метода случайного поиска с постоянным радиусом поиска и случайными направлениями

1. Задаем начальную точку X^0 , начальный радиус гиперсферы ρ^0 , и полагаем счетчик числа итераций $r = 0$.

2. Генерируем случайные точки X_i^r , $i \in [1, \dots, N]$ равномерно распределенные по поверхности гиперсферы радиуса ρ^r с центром в точке X^r . Здесь N – количество точек – свободный параметр метода.
3. Вычисляем значения минимизируемой функции $\Phi(X_i^r)$ в полученных точках и находим точку, в которой достигается минимальное значение функции $\Phi(X)$: $\Phi(X_k^r) = \min_{i \in [1, \dots, N]} \Phi(X_i^r)$.
4. Каким-либо из рассмотренных в главе 2 одномерных методов оптимизации (например, методом Пауллы) находим минимум функции $\Phi(X)$ в направлении $(X_k^r - X^r)$: $\Phi(X^{r+1}) = \min_{\lambda \in R^1} \Phi(X^r + \lambda(X_k^r - X^r))$.
5. Проверяем условие окончания итераций (см. ниже). Если условие окончания выполнено, то полагаем $X^* \approx X^{r+1}$ и завершаем итерации. Иначе – принимаем $r = r + 1$ и переходим к п.2.

В качестве условия окончания поиска можно использовать одно из стандартных условий окончания итераций:

$$\|X^{r+1} - X^r\| \leq \varepsilon_X, \quad (3.12.1)$$

где ε_X – константа, определяющая требуемую точность решения по X ;

$$|\Phi(X^{r+1}) - \Phi(X^r)| \leq \varepsilon_\Phi, \quad (3.12.2)$$

где ε_Φ – константа, определяющая требуемую точность решения по Φ .

Могут быть использованы также другие критерии окончания поиска, например, условие не превышения текущим радиусом гиперсферы величины ε_X :

$$\rho^r \leq \varepsilon_X. \quad (3.12.3)$$

В процессе поиска радиус гиперсферы может меняться, увеличиваясь при удачных шагах (вдали от точки X^*) и уменьшаясь при неудачных шагах (вблизи от точки X^*).

Поиск может быть ускорен, если точки на гиперсфере выбирать (случайным образом) в некотором секторе по отношению к предыдущему направлению. Угол раскрыва этого сектора может меняться в процессе поиска.

Метод случайного поиска с постоянным радиусом поиска и случайными направлениями иллюстрирует рис. 3.12.1, на котором показан фрагмент линий уровня функции Химмельблау.

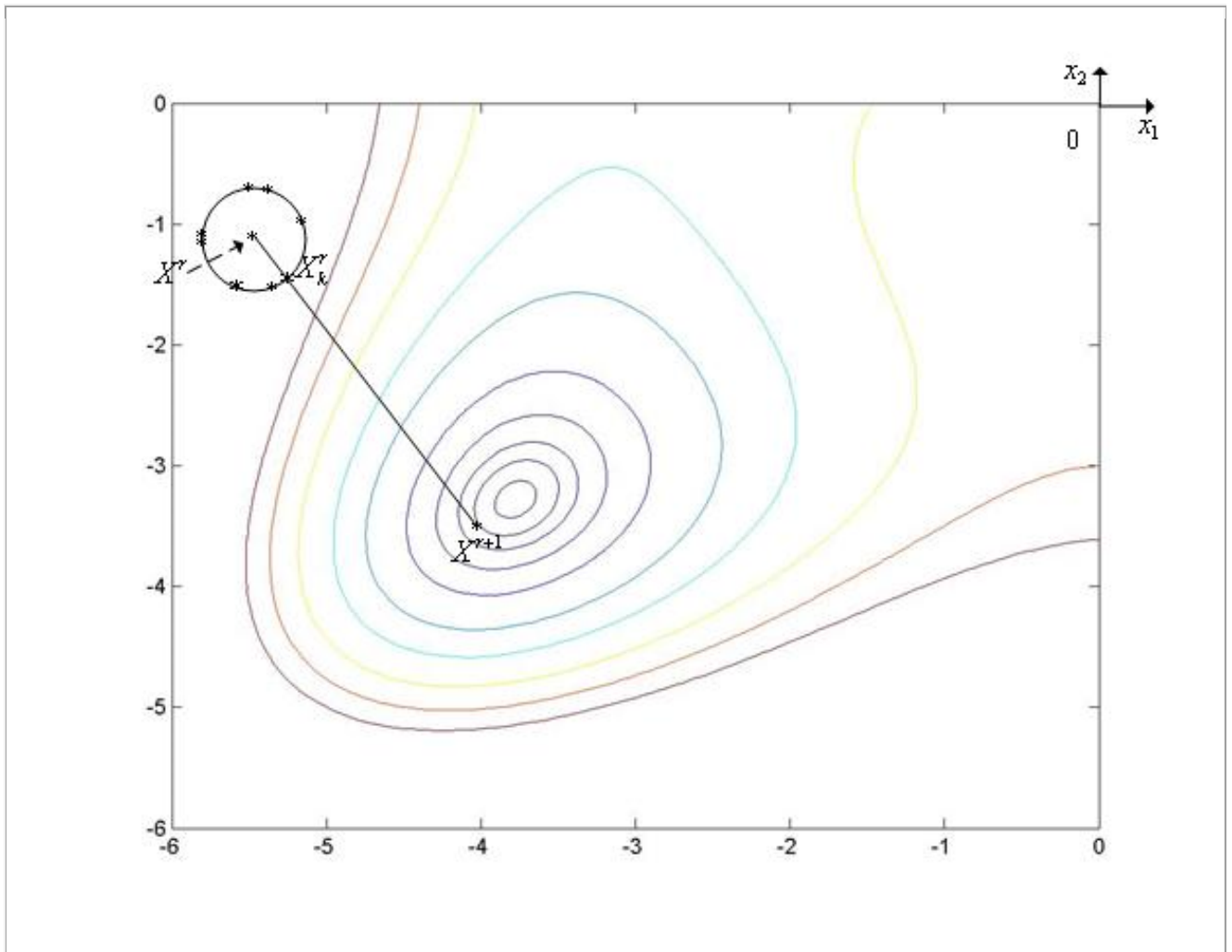


Рис. 3.12.1. Траектория поиска минимума функции Химмельблау методом случайного поиска с постоянным радиусом поиска и случайными направлениями ($n=2$)

На рисунке точки, лежащие на окружности с центром в точке X^r , соответствуют случайным точкам X_i^r , $i \in [1, \dots, N]$.

Примечание 3.12.1. Одна итерация по методу случайного поиска с постоянным радиусом поиска и случайными направлениями может привести к уменьшению минимизируемой функции в большей степени, чем один шаг поиска в направлении антиградиента этой функции.

Данное утверждение иллюстрирует рис. 3.12.2, на котором показаны линии уровня двумерной квадратичной функции (см. параграф 3.4).

$$\Phi(x) = \Phi(x_1, x_2) = x_1^2 + x_2^2 + 3(x_1 + x_2)^2 = 4x_1^2 + 4x_2^2 + 6x_1x_2.$$

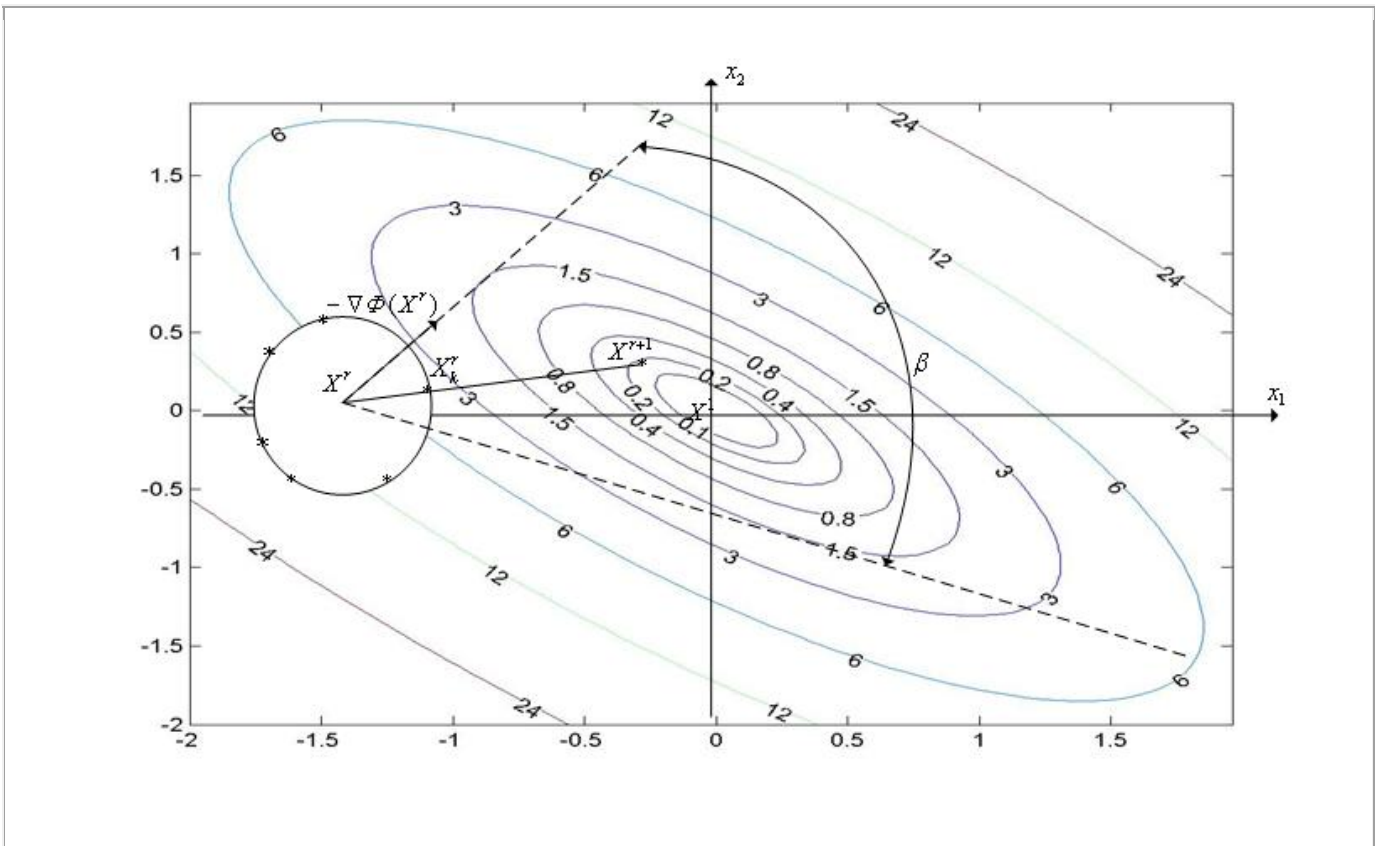


Рис. 3.12.2. Один шаг поиска в направлении антиградиента минимизируемой функции ($\Phi(X^r)$) приводит на линию уровня (1.5). В то же время одна итерация по методу случайного поиска с постоянным радиусом поиска и случайными направлениями – на линию уровня ~ 0.2 . Точки на окружности с центром в точке X^r соответствуют случайным точкам. Любое направление поиска в секторе β лучше, чем направление антиградиента

РАЗДЕЛ 4 НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ. МНОГОМЕРНАЯ УСЛОВНАЯ ОПТИМИЗАЦИЯ

4.1 Методы последовательной безусловной оптимизации

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

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

где множество допустимых значений:

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

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

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

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

В качестве приближенного решения задачи (4.1.1), (4.1.2) принимается решение \tilde{X} вспомогательной задачи (4.1.3) при достаточно большом α . Поясним идею методов последовательной безусловной оптимизации примером.

Пример 4.1.1. Пусть

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

и имеется одно ограничение типа равенств с ограничивающей функцией

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

Положим

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

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

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

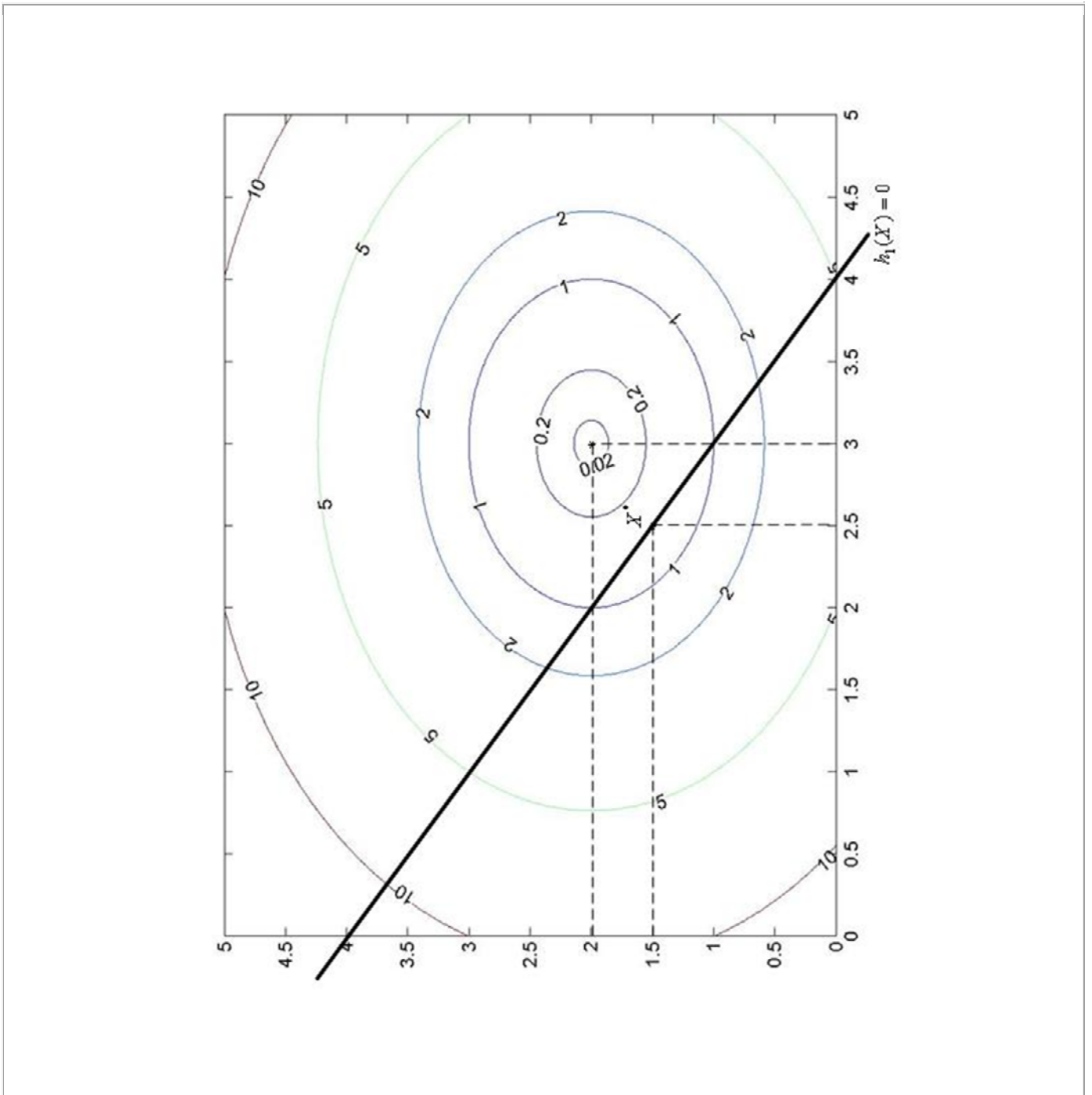


Рис. 4.1.1. К примеру 4.1.1. Точка минимума функции $Q^\alpha(X)$ при $\alpha = 0$ имеет координаты $(3, 2)$. Решением задачи (4.1.3) – (4.1.6) является точка X^* с координатами $(2.5, 1.5)$

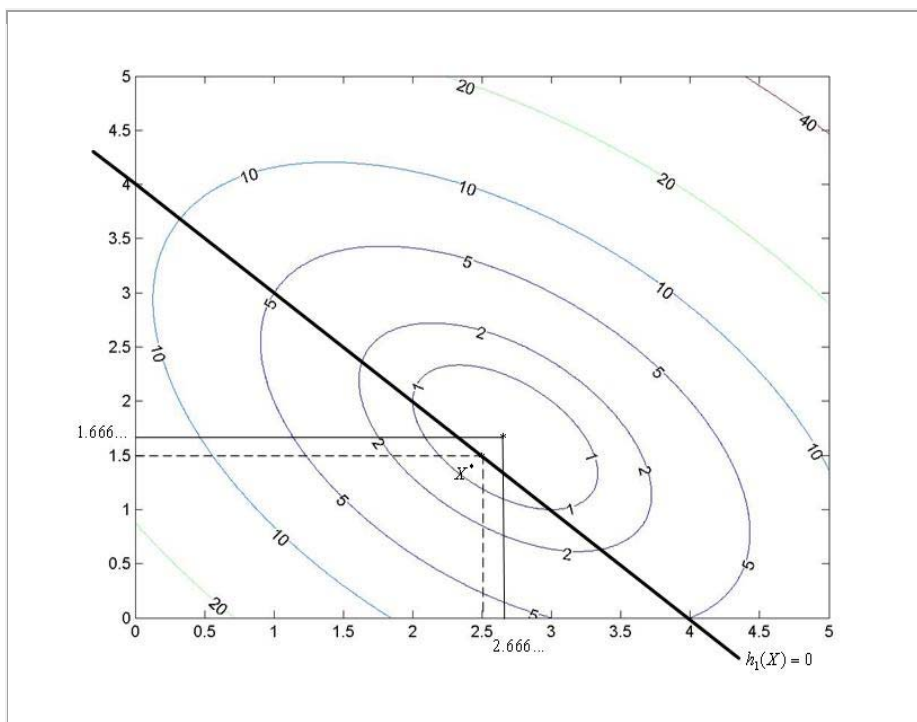


Рис. 4.1.2. К примеру 4.1.1. Точка минимума функции $Q^\alpha(X)$ при $\alpha=1$ имеет координаты $(2.666..., 1.666...)$. Решением задачи (4.1.3) – (4.1.6) является точка X^* с координатами $(2.5, 1.5)$

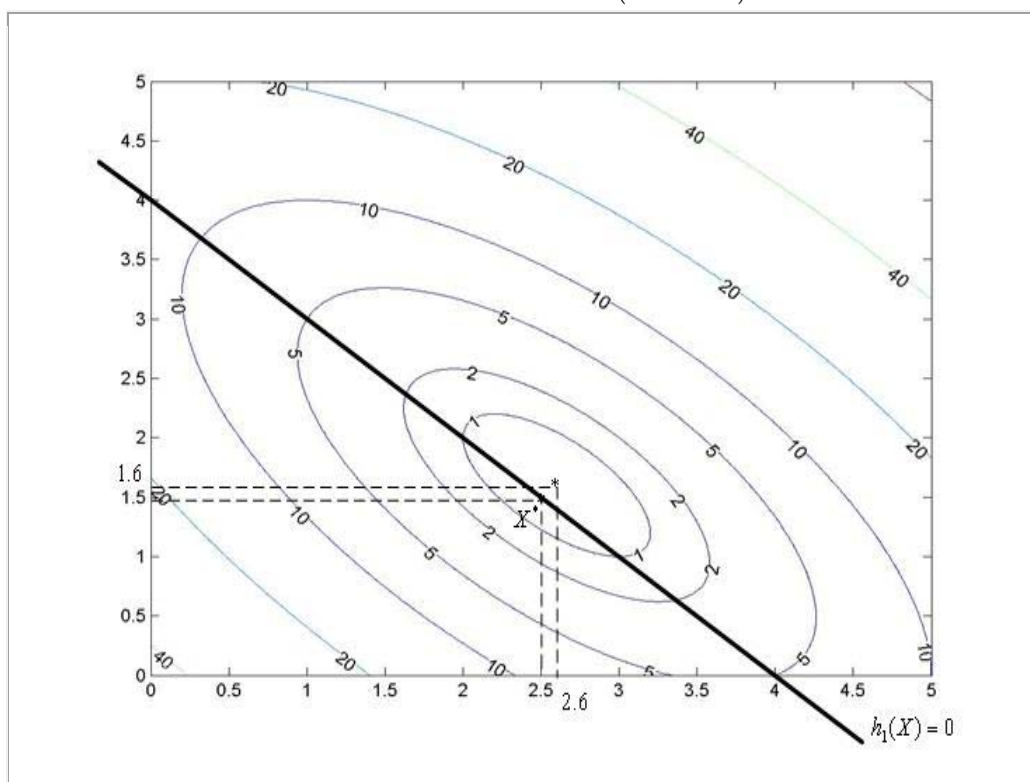


Рис. 4.1.3. К примеру 4.1.1. Точка минимума функции $Q^\alpha(X)$ при $\alpha=2$ имеет координаты $(2.6, 1.6)$. Решением задачи (4.1.3) – (4.1.6) является точка X^* с координатами $(2.5, 1.5)$

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

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

В методе штрафных функций функцию $P^\alpha(X)$, которая в этом случае называется **штрафной функцией**, подбирают таким образом, чтобы при больших α функция $Q^\alpha(X)$ мало отличалась от функции $\Phi(X)$ при $X \in D$ и быстро возрастала при удалении точки $X \notin D$ от границы области допустимых значений D . В методе штрафных функций точка X в процессе поиска может выходить за границы области D (см. рис. 4.1.4). Т.е. метод штрафных функций относится к классу **методов внешней точки**. Рассмотренный выше пример 4.1.1 также иллюстрирует метод штрафных функций.

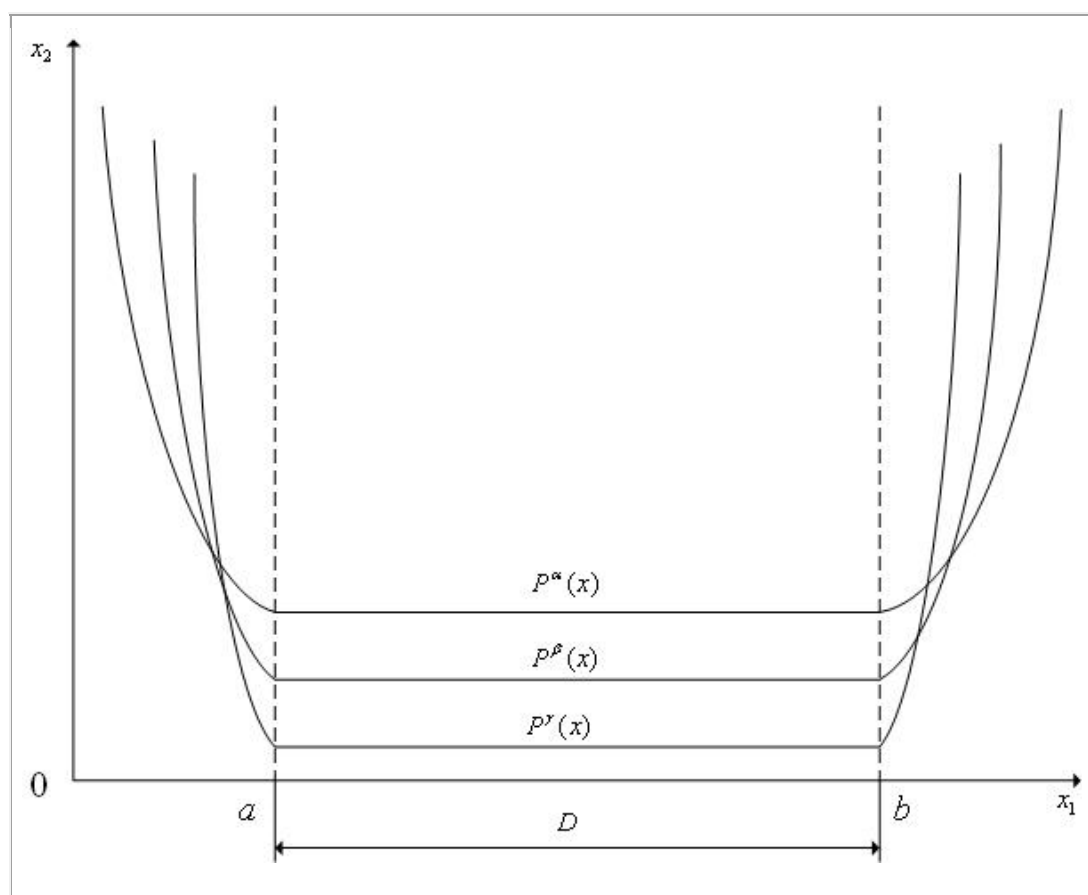


Рис. 4.1.4. К методу штрафных функций ($n=1$). Интервал $[a, b]$ – область допустимых значений D ; $\gamma > \beta > \alpha$

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

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

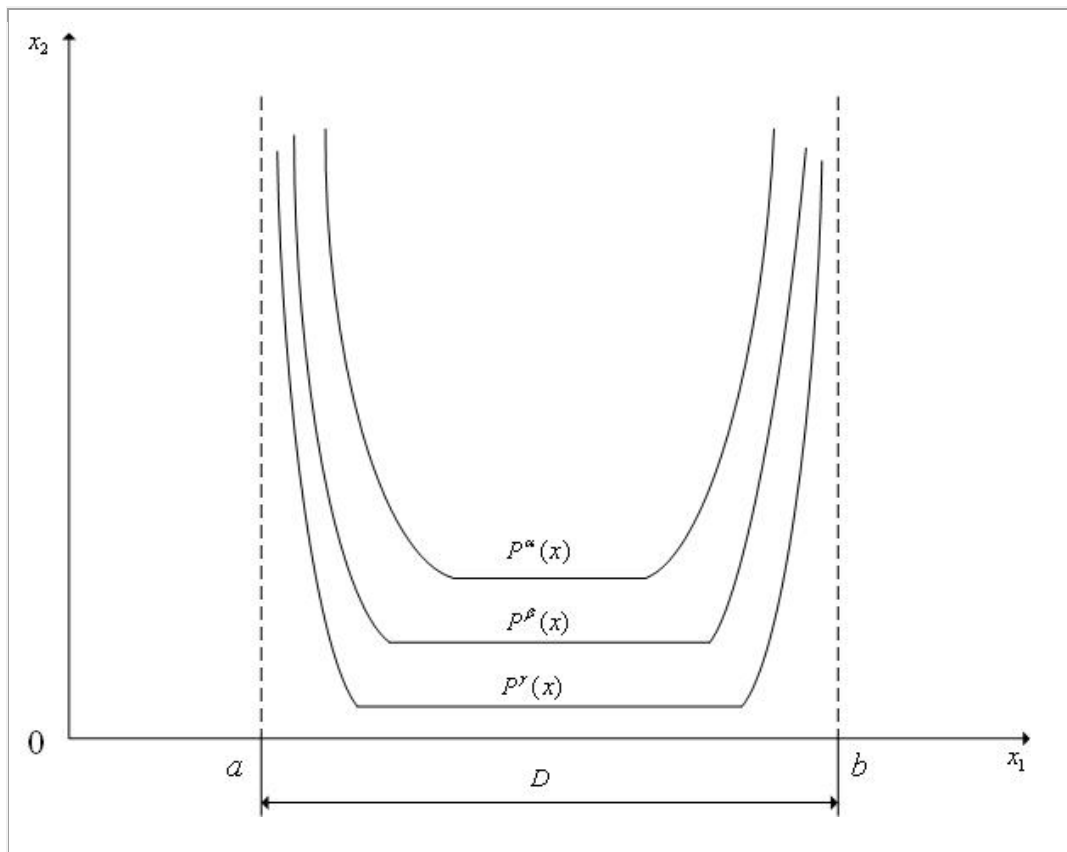


Рис. 4.1.5. К методу барьерных функций ($n = 1$). Интервал $[a, b]$ – область допустимых значений D ; $\gamma > \beta > \alpha$

Штрафная функция в общем случае имеет вид:

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

где $\alpha^r = (\alpha_h^r, \alpha_g^r)$ – двумерный вектор параметров штрафной функции; μ_i, λ_j – весовые коэффициенты, могущие изменяться в процессе итераций, $H(h_i(X)), G(g_j(X))$ – функционалы над функциями $h_i(X) \in [1, \dots, m]$, $g_j(X) \in [1, \dots, l]$, соответственно.

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

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

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

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

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

В качестве функционалов $H(h_i(X)), G(g_j(X))$ можно взять расстояния в какой-либо метрике от точки X до соответствующей границы множества D . Однако, вычисление этих расстояний, а значит и значений штрафной функции, может быть затруднительным. Поэтому обычно применяют штрафные функции более удобного вида.

Так в качестве функционалов $H(h_i(X))$ обычно используют функционалы $H(h_i(X)) = h_i^2(X)$, в качестве функционалов $G(g_j(X))$ – функционалы

$$G(g_j(X)) = (g_j^+(X))^2, \text{ где } g_j^+(X) = \begin{cases} 0, & g_j(X) \geq 0, \\ g_j(X), & g_j(X) < 0. \end{cases}$$

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

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

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

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

Схема метода штрафных функций

1. Задаем начальную точку X^0 и полагаем счетчик числа итераций $r = 0$.
2. Исходя из точки X^r , одним из методов локальной безусловной оптимизации решаем задачу – находим точку X^{r+1} .
3. Проверяем условие окончания поиска (4.1.8). Если условие окончания поиска выполнено, то полагаем $X^* = X^{r+1}$ и завершаем итерации. Иначе – по некоторому правилу увеличиваем значения параметров α_h^r, α_g^r , полагаем $r = r + 1$ и переходим к п.3.

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

4.2 Метод скользящего допуска

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

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

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

Основы метода скользящего допуска

Метод скользящего допуска существенно использует множество

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

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

При этом функционал $T(X)$ должен быть сконструирован таким образом, чтобы $T(X) = 0$ при $X \in D$ и значение $T(X)$ возрастало по мере удаления точки X от границы области допустимых значений D . Критерий скользящего допуска Δ^r определяет требуемую точность выполнения ограничений, которые формируют область допустимых значений D , и конструируется таким образом, чтобы обеспечить его уменьшение с ростом количества итераций r .

Точка X называется **допустимой точкой**, если $T(X) = 0$, **почти допустимой точкой** – если $0 < T(X) \leq \Delta^r$, **недопустимой точкой** – если $T(X) > \Delta^r$. Поскольку величина Δ^r с ростом номера итерации уменьшается, отклонение от границы области D , при котором точка считается допустимой, сужается, так что в пределе рассматриваются только допустимые точки.

Метод скользящего допуска может быть скомбинирован со многими из рассмотренных ранее многомерных методов локальной безусловной оптимизации. Будем называть метод, с которым комбинируется метод скользящего допуска, **базовым методом**.

Одна итерация метода скользящего допуска состоит из одного или двух этапов:

1. С помощью базового метода, исходя из точки X^r , выполняем итерацию по решению задачи локальной безусловной оптимизации:

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

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

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

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

с условием окончания итераций

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

и заканчиваем данную итерацию.

Достоинством метода скользящего допуска является то, что степень нарушения ограничений по мере приближения к минимуму минимизируемой функции постепенно уменьшается. Т.е. на первых итерациях ограничения могут удовлетворяться приближенно, а высокая точность удовлетворения ограничений необходима лишь в окрестности решения. Это обстоятельство позволяет сократить полный объем вычислений по сравнению с другими методами.

Одна из сложностей применения метода скользящего допуска – возможные осцилляции решения относительно границы области D (см. ниже).

Комбинация метода скользящего допуска с методом Нелдера-Мида

При комбинации метода скользящего допуска с методом Нелдера-Мида можно предложить разные виды критерия скользящего допуска. Чаще всего в

качестве этого критерия используют следующую функцию координат вершин деформируемого многогранника $(X_1^r, X_2^r, \dots, X_{n+1}^r)$:

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

Здесь X_c^r – вектор координат центра тяжести многогранника

$(X_1^r, X_2^r, \dots, X_{n+1}^r)$, так что величина $\frac{1}{n+1} \sum_{i=1}^{n+1} |X_i^r - X_c^r|$ – есть среднее

расстояние вершин многогранника $(X_1^r, X_2^r, \dots, X_{n+1}^r)$ от его центра тяжести.

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

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

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

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

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

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

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

Примечание 4.2.1. Поскольку метод Нелдера-Мида является детерминированным методом нулевого порядка, комбинация метода скользящего допуска с методом Нелдера-Мида также представляет собой детерминированный метод нулевого порядка.

Упрощенная схема комбинации метода скользящего допуска и метода Нелдера-Мида

Симплекс с вершинами X_i^r , $i \in [1, \dots, n+1]$ обозначим S^r .

1. Задаем начальный симплекс S^0 и полагаем счетчик числа итераций $r = 0$.
2. С помощью метода Нелдера-Мида, исходя из симплекса S^r , выполняем одну итерацию по решению задачи локальной безусловной оптимизации (4.2.4) – находим симплекс \tilde{S}^{r+1} с вершинами \tilde{X}_i^r , $i \in [1, \dots, n+1]$.
3. Вычисляем значения функционала $T(X)$ во всех вершинах симплекса \tilde{S}_i^{r+1} и значение критерия скользящего допуска Δ^r . Находим вершину симплекса \tilde{S}^{r+1} , в которой значение функционала $T(X)$ максимально, т.е. вершину, которая расположена дальше всех от границы области D . Обозначим эту вершину \tilde{X}_T^{r+1} .
4. Если $T(\tilde{X}_T^{r+1}) \leq \Delta^{r+1}$ (точка \tilde{X}_T^{r+1} является допустимой точкой или почти допустимой точкой), то проверяем условие окончания поиска (см. схему метода Нелдера-Мида). Если это условие выполнено, то завершаем итерации. Если условие окончания поиска не выполнено, то формируем симплекс S^{r+1} с вершинами $X_i^{r+1} = \tilde{X}_i^{r+1}$, $i \in [1, \dots, n+1]$, полагаем $r = r + 1$ и переходим к п.2.
5. Если $T(\tilde{X}_T^{r+1}) > \Delta^{r+1}$ (точка \tilde{X}_T^{r+1} является недопустимой точкой), то с помощью метода Нелдера-Мида, исходя из точки \tilde{X}_T^{r+1} , решаем задачу локальной безусловной оптимизации (4.2.5) с критерием окончания итераций (4.2.6) –

находим точку X_T^{r+1} . Формируем новый симплекс S^{r+1} с вершинами $X_i^{r+1} = \tilde{X}_i^{r+1}$, $i \in [1, \dots, n+1]$, $i \neq T$, X_T^{r+1} , полагаем $r = r+1$ и переходим к п.2.

Ослабление осцилляций решения

Как отмечалось выше, одной из сложностей применения метода скользящего допуска являются возможные осцилляции решения относительно границы области D . Поясним суть этого явления на примере.

Пример 4.2.1. Рассмотрим двумерную задачу условной оптимизации (4.2.1), когда критерий оптимальности равен $\Phi(X) = -x_1 - x_2$ и множество допустимых значений D определяется ограничениями:

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

Положим, что на r -ой итерации координаты вершин текущего симплекса S^r равны X_1^r, X_2^r, X_3^r . Тогда после одной итерации по решению задачи локальной безусловной оптимизации (4.2.4) методом Нелдера-Мида получим симплекс \tilde{S}^{r+1} с вершинами \tilde{X}_i^r , $i \in [1, \dots, n+1]$ (см. рис. 4.2.1).

Положим далее, что точка \tilde{X}_1^{r+1} , расположенная далее всех от границы области D , является недопустимой точкой, т.е. $T(\tilde{X}_1^{r+1}) > \Delta^{r+1}$. Тогда при решении помощью метода Нелдера-Мида задачи локальной безусловной оптимизации (4.2.5) возможна ситуация, приведенная на рис. 4.2.2. Из (4.2.8), (4.2.9) следует, что если $X \notin D$ и точка X лежит в первой четверти системы координат $0x_1x_2$, то $T(X) = x_1^2 + x_2^2 - 9$. На рис. 4.2.2 показаны линии уровня функции $T(X)$ именно для этого случая.

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

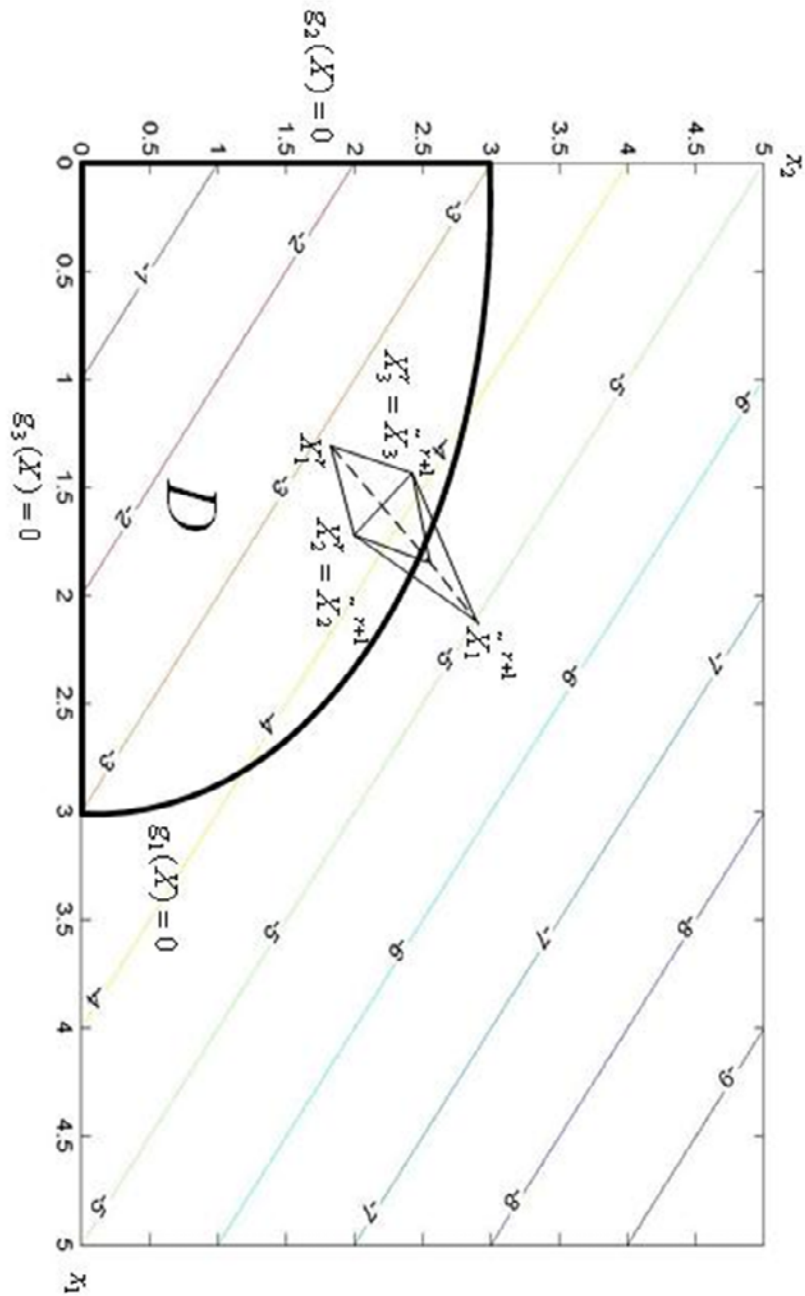


Рис. 4.2.1. К примеру 4.2.1. После успешного отражения вершины X_1^r выполнено успешное растяжение симплекса

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

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

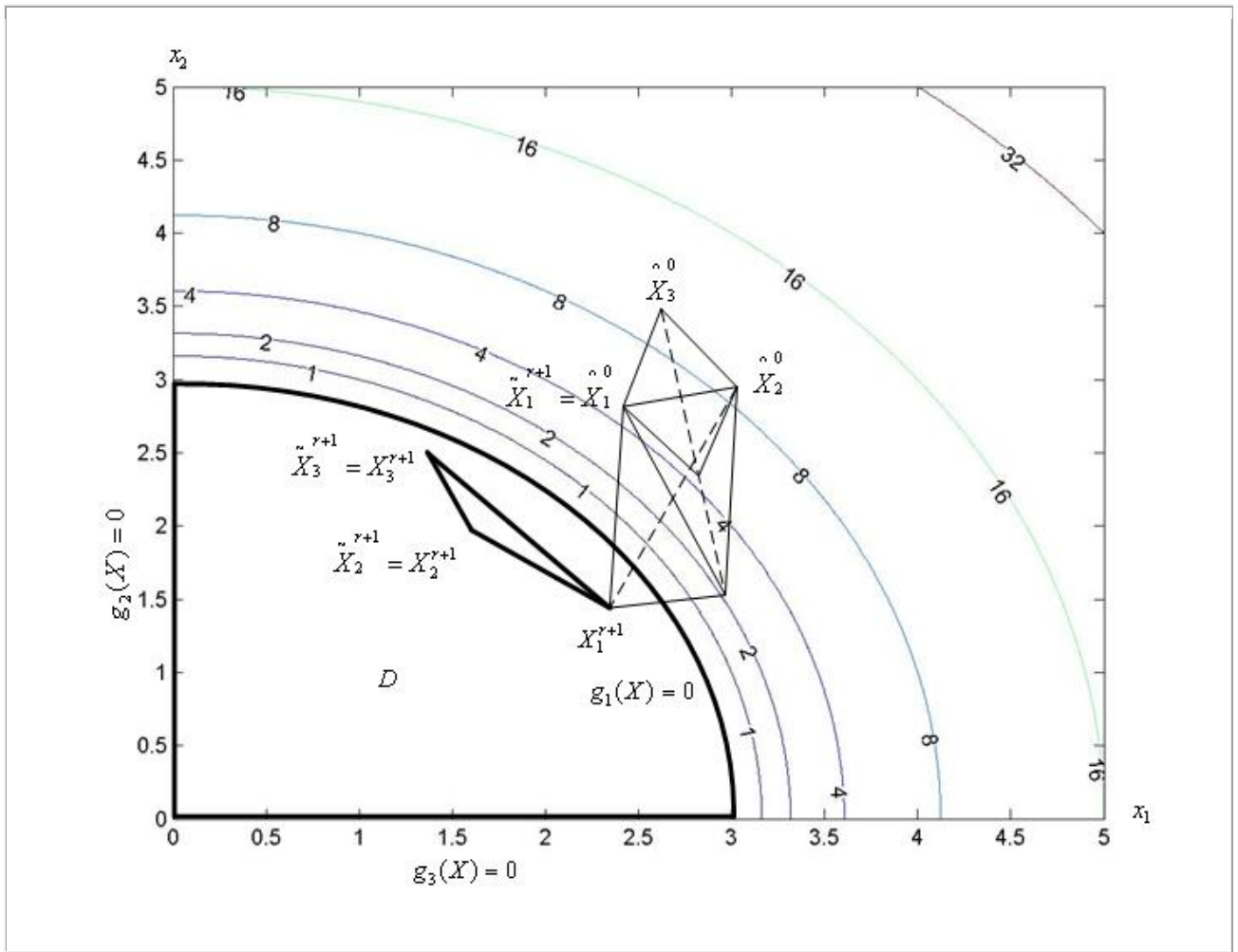


Рис. 4.2.2. К примеру 4.2.1. После успешного отражения вершины X_3^0 выполнено растяжение симплекса и отражение вершины X_2^0

Чаще с этой целью используют квадратичную интерполяцию функции $T(X)$ на отрезке $[X_T^r, \tilde{X}_T^{r+1}]$ по трем точкам $(X_T^r, \Phi(X_T^r))$, $(A^r, \Phi(A^r))$, $(\tilde{X}_T^{r+1}, \Phi(\tilde{X}_T^{r+1}))$, где A^r – также середина отрезка $[X_T^r, \tilde{X}_T^{r+1}]$ – см. рис. 4.2.3.

Обозначим эту интерполирующую функцию $y(x) = \alpha x^2 + \beta x + \gamma$. Вместо точки

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

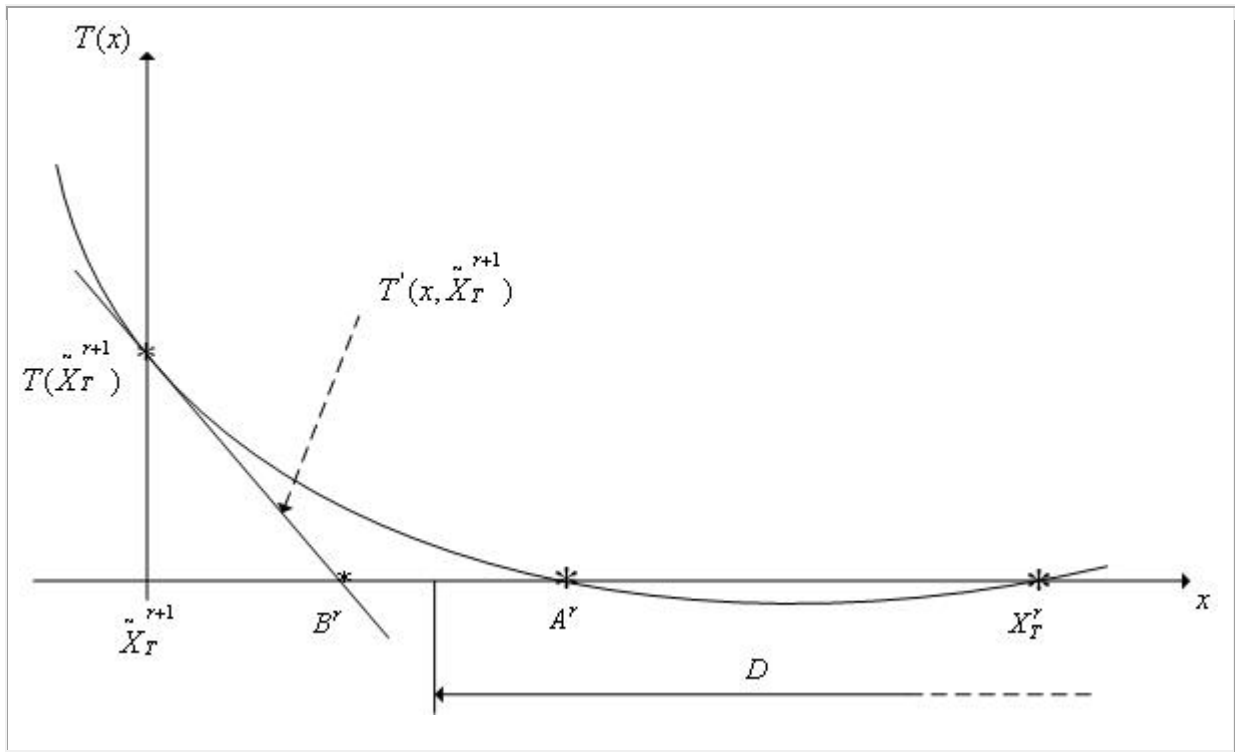


Рис. 4.2.3. Использование квадратичной интерполяции функции $T(X)$ на отрезке по трем точкам для ослабления осцилляций. Случай, когда точка A^r принадлежит области допустимых значений D

4.3 Модифицированный метод комплексов

Рассмотрим многомерную задачу локальной условной оптимизации:

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

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

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

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

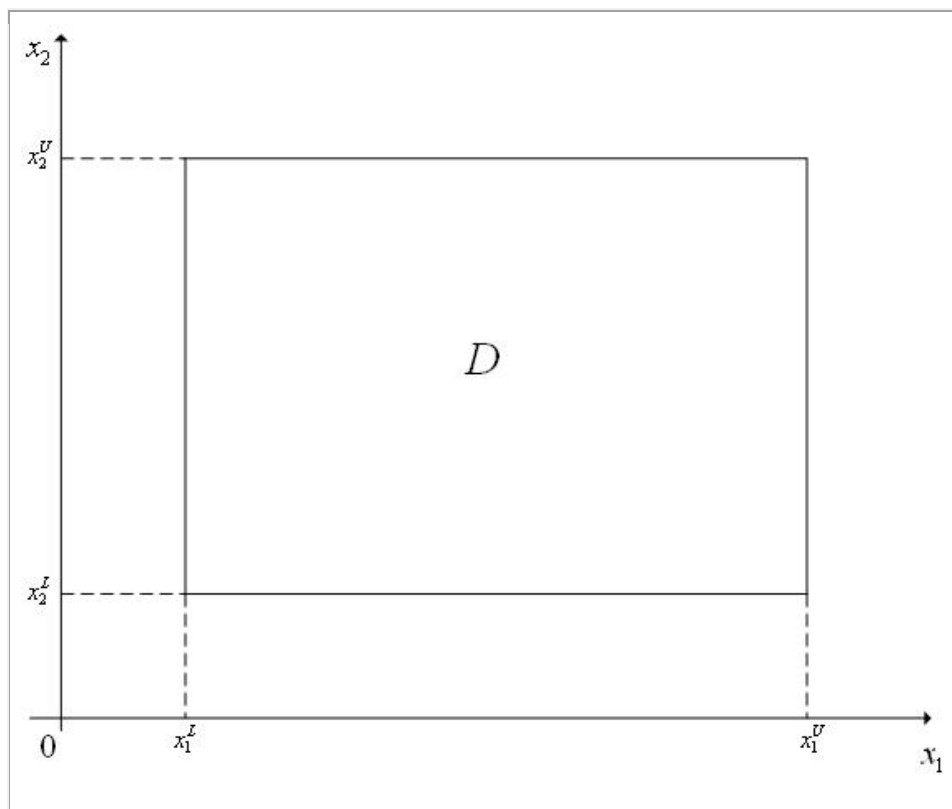


Рис. 4.3.1. Область допустимых значений D в виде гиперпараллелепипеда;
 $n = 2$

Метод комплексов в многомерной задаче безусловной оптимизации рассмотрен в параграфе 3.10. В данном параграфе рассматривается модификация этого метода для решения многомерной задачи условной оптимизации – **модифицированный метод комплексов**.

Основные операции метода комплексов

Напомним, что комплексом называется многогранник с $N > n + 1$ вершинами (не обязательно выпуклый). Рекомендуется использовать комплекс с $N = 2n$ вершинами. Так же, как при решении задачи безусловной оптимизации, при решении задачи (4.3.1) методом комплексов используются следующие операции:

- генерация случайного комплекса;
- отражение вершины комплекса с растяжением;
- сжатие комплекса.

Генерация случайного комплекса

Координаты вершин случайного комплекса с N вершинами могут быть найдены по формуле

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

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

Отражение вершины комплекса с растяжением

Положим, что задан комплекс C^r с N вершинами X_i^r , $i \in [1, \dots, N]$, и его вершину X_k^r необходимо отразить через центр тяжести комплекса с растяжением. В новом комплексе C^{r+1} все вершины, кроме k -ой, совпадают с соответствующими вершинами исходного комплекса C^r , а k -я вершина находится на прямой, проходящей через центр тяжести этого комплекса и его вершину X_k^r (см. рис. 4.3.2). Обозначим координаты вершин нового комплекса X_i^{r+1} , $i \in [1, \dots, N]$.

Тогда имеем:

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

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

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

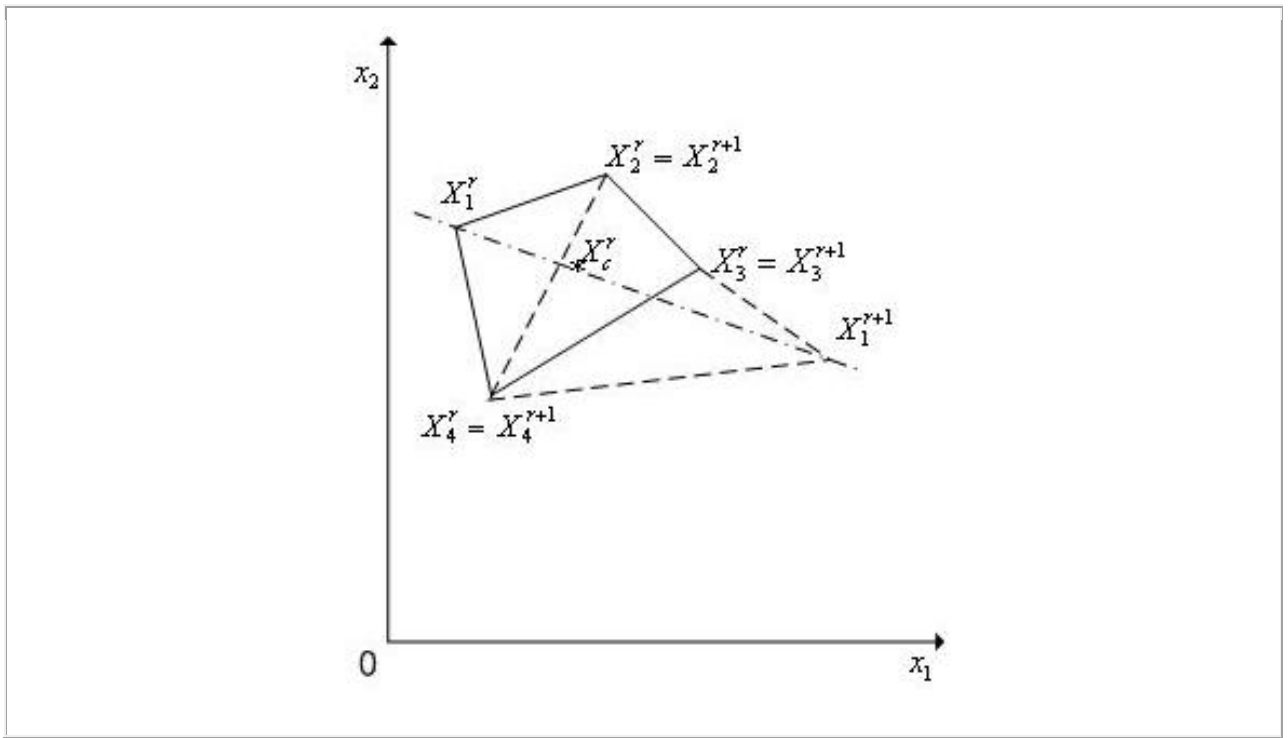


Рис. 4.3.2. Отражение вершины комплекса C^r через центр его тяжести с растяжением. Пунктиром показан новый комплекс C^{r+1}

Сжатие комплекса

Положим, что задан комплекс C^r с N вершинами X_i^{r+1} , $i \in [1, \dots, N]$, и его X_k^r вершину необходимо переместить ближе к центру тяжести комплекса C^r – выполнить сжатие комплекса. В новом комплексе C^{r+1} все вершины, кроме k -ой, совпадают с соответствующими вершинами исходного комплекса C^r , а k -я вершина находится на прямой, проходящей через центр тяжести этого комплекса и его вершину X_k^r (см. рис. 4.3.3). Обозначим координаты вершин нового комплекса X_i^{r+1} , $i \in [1, \dots, N]$. Тогда имеем

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

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

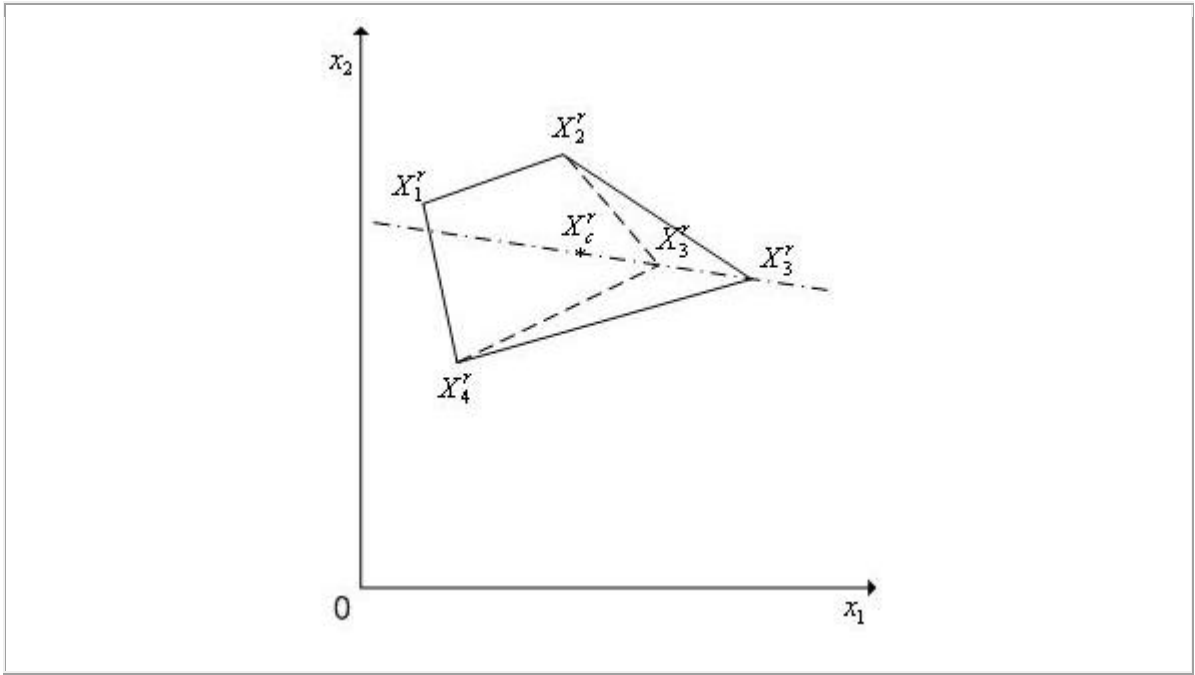


Рис. 4.3.3. Сжатие комплекса C^r . Пунктиром показан новый комплекс C^{r+1}

Упрощенная схема модифицированного метода комплексов

1. Задаем начальную точку $X_0^0 \in D$, исходя из которой должен быть построен комплекс C^0 , величину l^0 , а также коэффициенты α, β ; полагаем счетчик числа итераций $r=0$.
2. Строим начальный комплекс C^0 :
 - поочередно для $i=1, 2, \dots, N$ по формуле (4.3.3) находим координаты вершин комплекса X_i^r комплекса C^r ;
 - если вершина X_p^r является недопустимой (выходит за границы области D), то по формуле, аналогичной формуле (4.3.6), выполняем сжатие уже построенного комплекса с p вершинами, вдоль направления $\left(X_p^r, (X_c^r)_p \right)$, где $(X_c^r)_p$ — центр тяжести уже найденных $(p-1)$ -ой вершин комплекса (см. рис. 4.3.4);
 - если после сжатия комплекса вершина X_p^r по-прежнему является недопустимой, повторяем описанную процедуру сжатия;
 - вычисляем значения $\Phi(X_i^r)$ функции $\Phi(X)$ во всех вершинах построенного комплекса C^r .

3. Находим максимальное из значений функции $\Phi(X)$ в вершинах комплекса

$$C^r : \Phi(X_k^r) = \max_{i \in [1, \dots, N]} \Phi(X_i^r).$$

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

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

$$\Phi(X_i^{r+1}) > \Phi(X_i^r),$$

то по формуле (4.3.6), выполняем сжатие комплекса C^{r+1}

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

до тех пор, пока точка X_k^{r+1} не станет допустимой (см. рис. 4.3.5). Переходим к

п.5;

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

$$\Phi(X_i^{r+1}) < \Phi(X_i^r),$$

то переходим к шагу 5;

– если точка X_k^{r+1} является не допустимой, но $\Phi(X_i^{r+1}) < \Phi(X_i^r)$, то переходим к п. 6.

5. Проверяем условие окончания поиска (см. ниже). Если условие окончания поиска выполнено, то в качестве точки X^* принимаем вершину комплекса C^{r+1} , к

которой функция $\Phi(X)$ имеет наименьшее значение, вычисляем соответствующие значения $\Phi(X)$ и завершаем итерации. Иначе – переходим к п. 3.

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

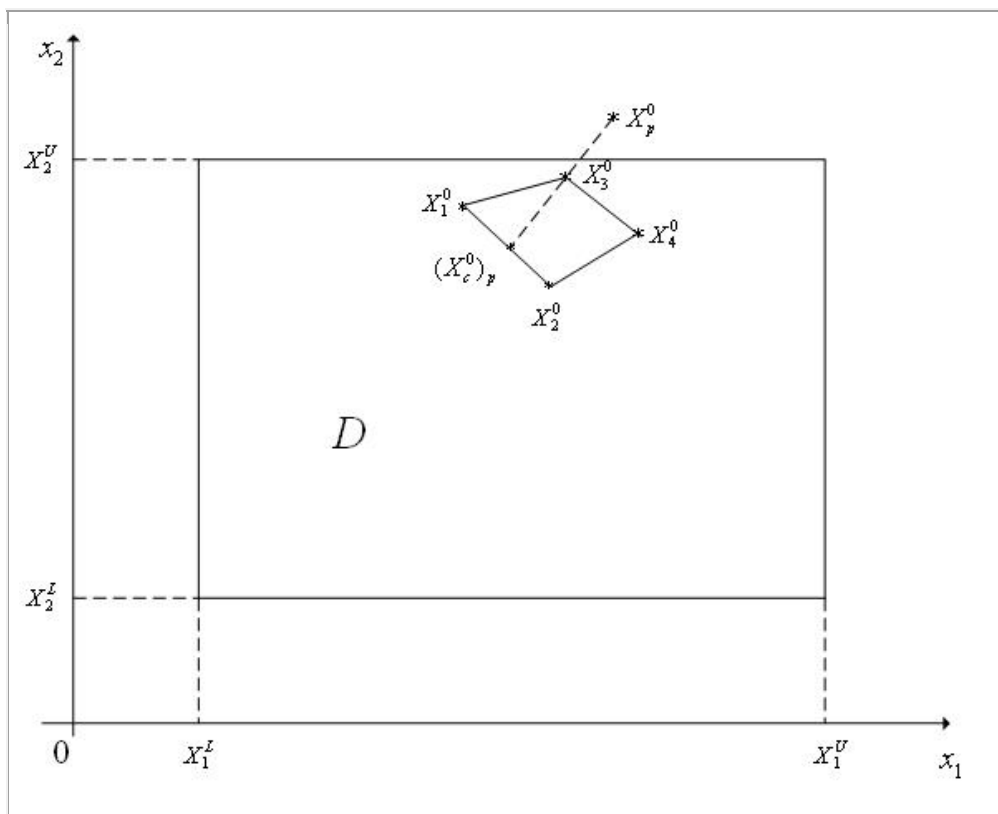


Рис. 4.3.4. Построение комплекса C^0

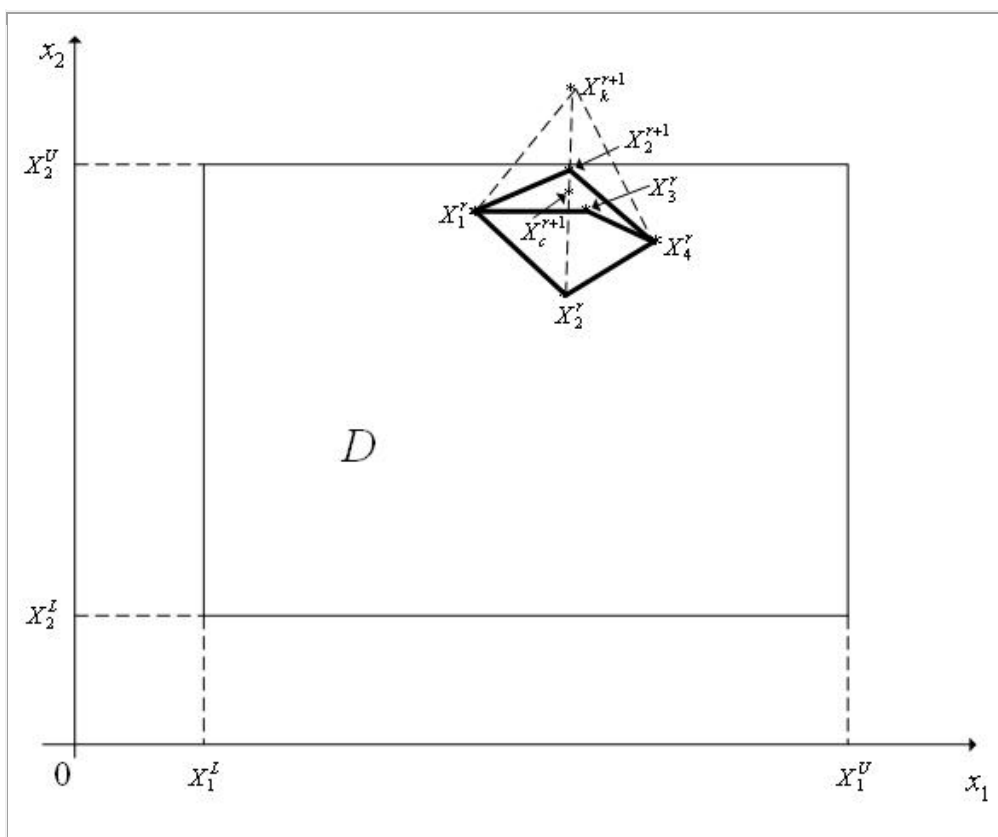


Рис. 4.3.5. Построение комплекса C^{r+1}

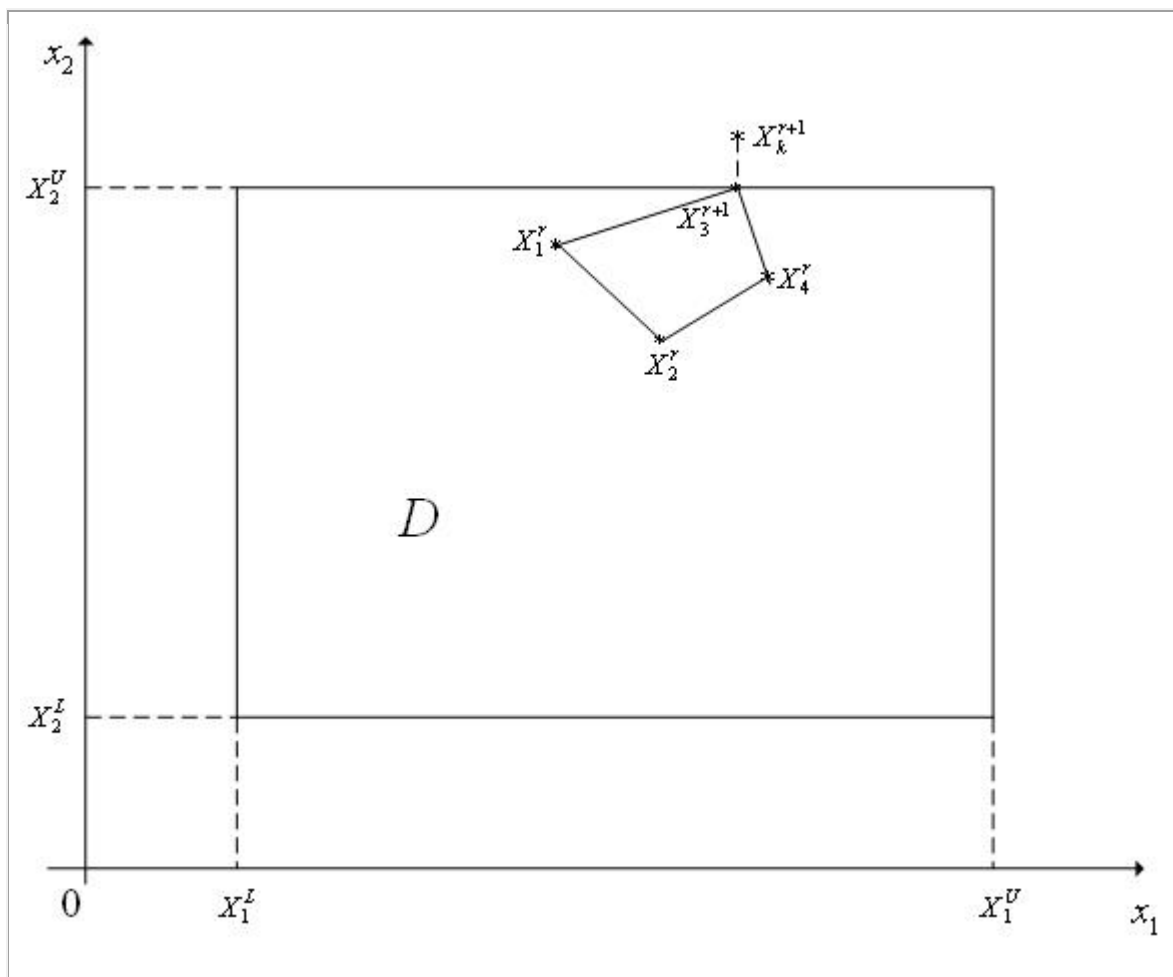


Рис. 4.3.6. Построение комплекса C^{r+1}

На рис. 4.3.4 точка X_0^p оказалась за границей области D . После операции сжатия комплекса вершины X_0^p комплекса вдоль направления $(X_0^p, (X_c^o)_p)$ получаем вершину X_3^0 . Здесь $(X_c^o)_p$ – центр тяжести комплекса.

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

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

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

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

Изложенная схема метода комплексов приводит к "уплощению" комплекса вблизи границы области допустимых значений D , что может значительно уменьшить эффективность метода. С целью преодоления этого недостатка через фиксированное количество итераций находятся максимальная и минимальная диагонали комплекса и, если их отношение превышает заданное, то по рассмотренной схеме производится построение нового комплекса.

4.4 Метод линейной аппроксимации

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

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

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

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

Суть метода линейной аппроксимации

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

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

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

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

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

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

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

Пример 4.4.1. Рассмотрим следующую двумерную задачу условной оптимизации с тремя ограничениями типа неравенств (первое ограничение – нелинейное, второе и третье ограничения – линейные), где $\Phi(X) = x_1^2 + (x_2 - 6)^2 - 12$, $g_1(X) = -x_1^2 - x_2^2 + 25$, $g_2(X) = x_1$, $g_3(X) = x_2$. Положим, что текущая точка есть $X^r = (2, 4)$. Линеаризуем целевую функцию $\Phi(X)$ и ограничивающую функцию $g_1(X)$ в окрестности этой точки. Поскольку $\Phi(X^r) = 2^2 + (4 - 6)^2 - 12 = -4$,

$\nabla\Phi(X)=\begin{pmatrix} 2x_1 \\ 2(x_2-6) \end{pmatrix}$, $\nabla\Phi(X^r)=\begin{pmatrix} 4 \\ -4 \end{pmatrix}$ по формуле (4.4.3) имеем

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

Аналогично для ограничивающих функции по формуле (4.4.4) имеем:

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

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

Пример 4.4.1 иллюстрирует рис. 4.4.1.

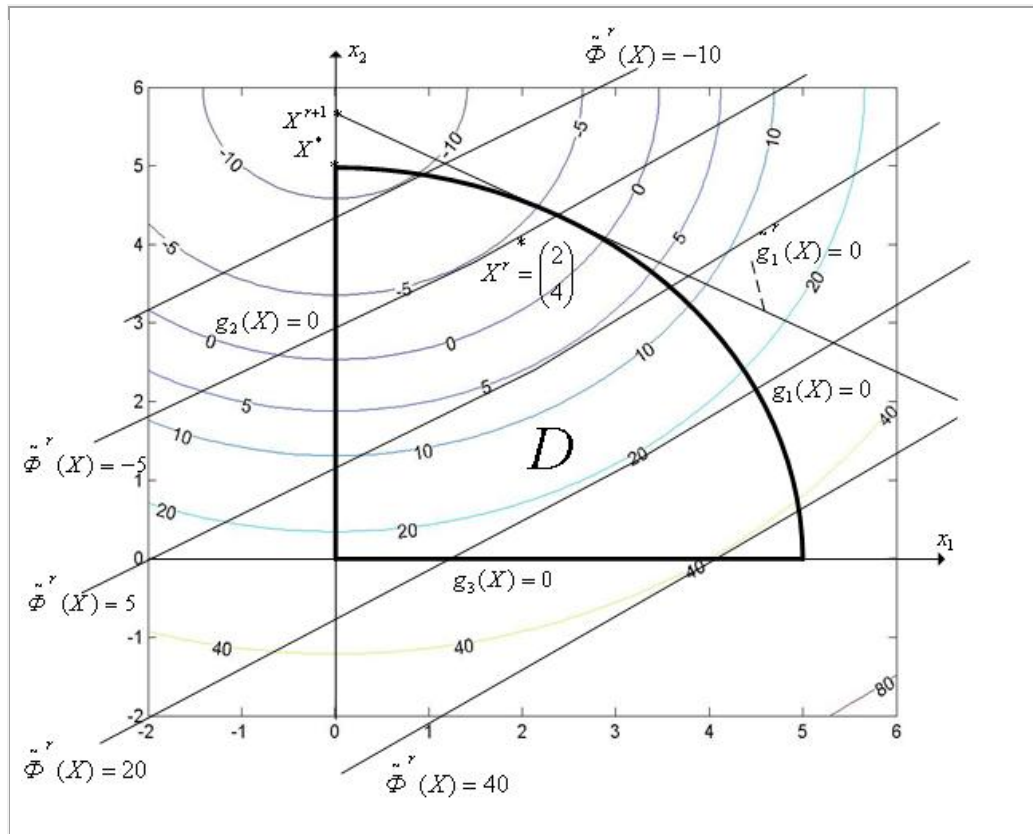


Рис. 4.4.1. Точка X^{r+1} лежит вне области допустимых значений D

Примечание 4.4.1. Прямая \tilde{g}_1^r представляет собой след от пересечения плоскости, которая является касательной к поверхности $z = g_1(X)$ в точке X^r , с плоскостью $0x_1x_2$. Эта прямая не обязательно является касательной к линии $g_1(X)=0$ – прямая $\tilde{g}_1^r(X)=0$ может пересекать кривую $g_1(X)=0$, быть касательной к ней или не иметь с ней общих точек. Аналогично, линия уровня

$\Phi^r(X) = c$ функции $\Phi(X)$ представляет собой след от пересечения плоскости, которая является касательной к поверхности $Z = \Phi(X)$ в точке X^r , с плоскостью $Z = c$.

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

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

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

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

Приведем два из множества возможных способов.

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

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

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

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

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

Схема метода линейной аппроксимации

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

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

В качестве критерия окончания поиска можно использовать стандартные условия окончания итераций:

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

или условие

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

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

Отметим следующие трудности, возникающие при использовании метода линейной аппроксимации:

1. Если функция $\Phi(X)$ имеет высокую степень нелинейности, то на основе решения вспомогательной задачи минимизации (4.4.5) направление поиска может быть выбрано слишком неточно (см. рис. 4.4.2), что приводит к медленной сходимости метода.

2. Метод требует, чтобы точка X^0 принадлежала множеству допустимых значений D . Если это требование не выполнено, то прежде придется использовать какой-либо метод поиска точки, принадлежащей множеству допустимых значений.

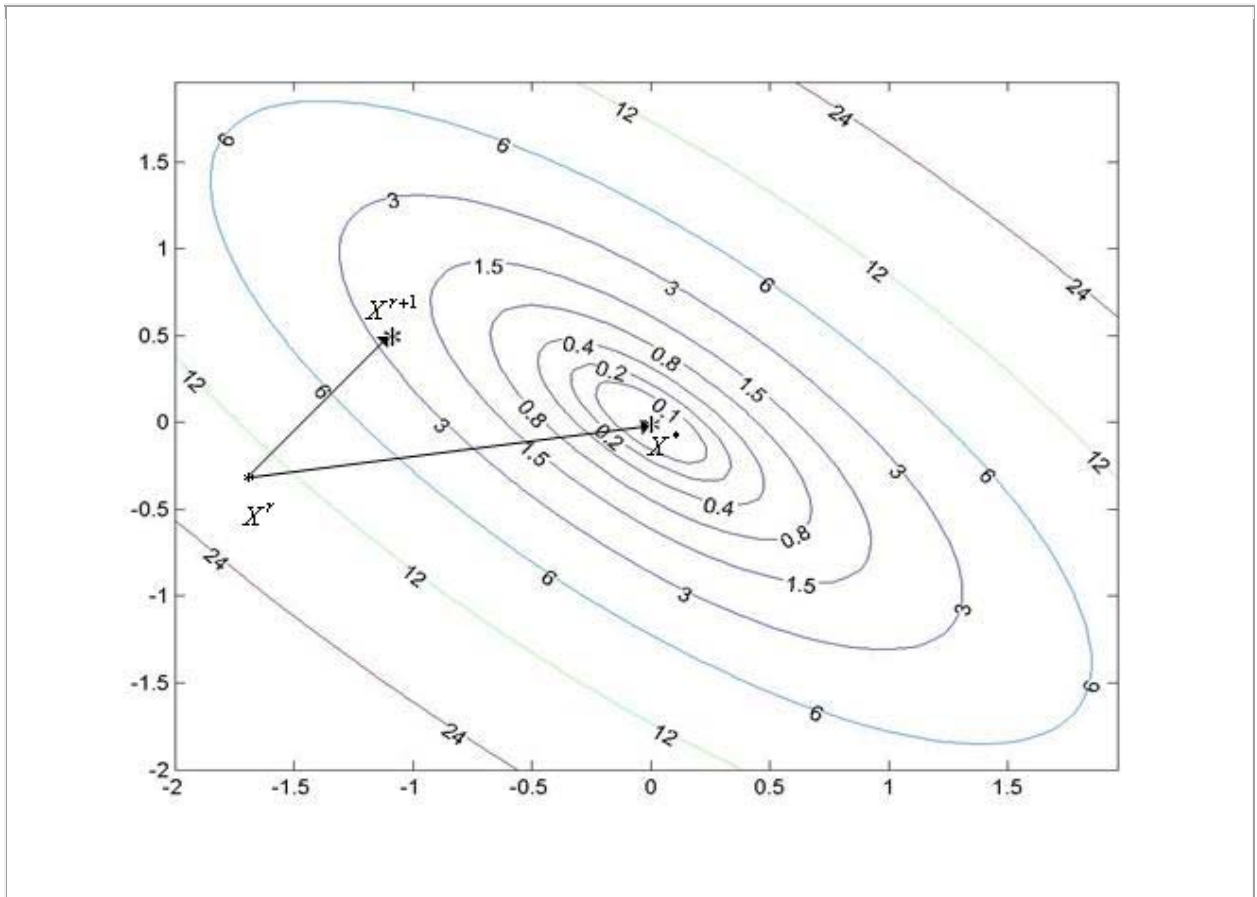


Рис. 4.4.2. Направление поиска ($X^{r+1} - X^r$), которое обеспечивает метод на основе линейной аппроксимации, далеко от оптимального направления ($X^* - X^r$)

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

4.5 Метод проекции градиента

Рассмотрим многомерную задачу локальной условной оптимизации:

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

где множество допустимых значений определяется только ограничениями типа неравенств

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

и целевая функция $\Phi(X)$ и ограничивающие функции $g_j(X), j \in [1, \dots, l]$ являются непрерывными и дифференцируемыми функциями, а ограничивающие функции еще и выпуклы.

Проектирование точки на множество

Идея метода проекции градиента состоит в том, что если на некоторой итерации точка

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

полученная с помощью градиентного метода наискорейшего спуска (см. параграф 3.7), оказывается вне множества допустимых значений D , то она возвращается на это множество. Возврат производится с помощью процедуры "проекция точки на множество". Напомним, что в формуле (4.5.3) λ^r - длина шага на r -ой итерации в направлении S^r ;

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

– единичный вектор направления антиградиента функции $\Phi(X)$ в точке X^r , $\|\ast\|$

– некоторая векторная норма, например, евклидова.

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

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

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

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

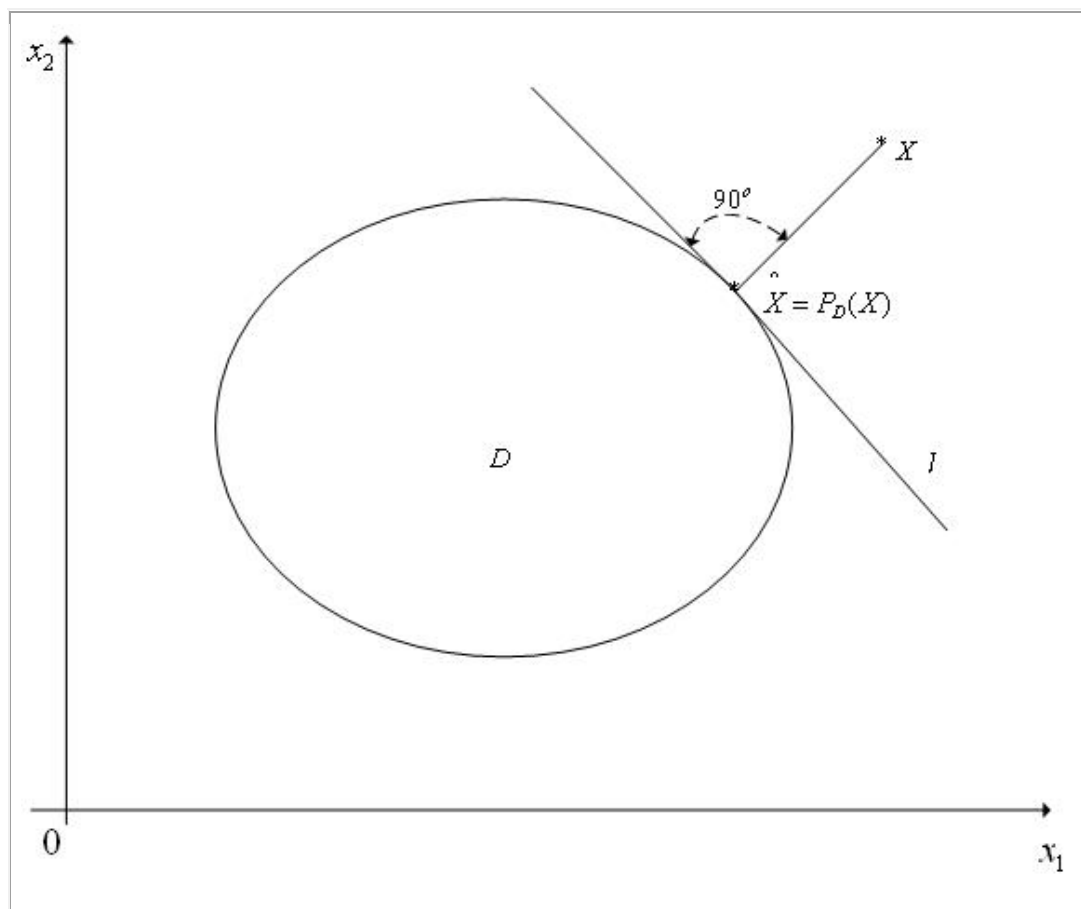


Рис. 4.5.1. К определению проекции точки на множество. Прямая l является касательной к границе области D в точке $P_D(X) = \hat{X}$

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

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

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

Наибольший практический интерес представляет ситуация, когда множество D таково, что задача (4.5.4) может быть решена в явном виде. Приведем несколько наиболее практически важных примеров таких множеств.

Схема комбинации метода проекции градиента с методом дробления шага

Метод проекции градиента может быть скомбинирован со многими градиентными методами (см. параграф 3.7).

Рассмотрим комбинацию метода проекции градиента с градиентным методом дробления шага.

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

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

Схема метода

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

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

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

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

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

или условие

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

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

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

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

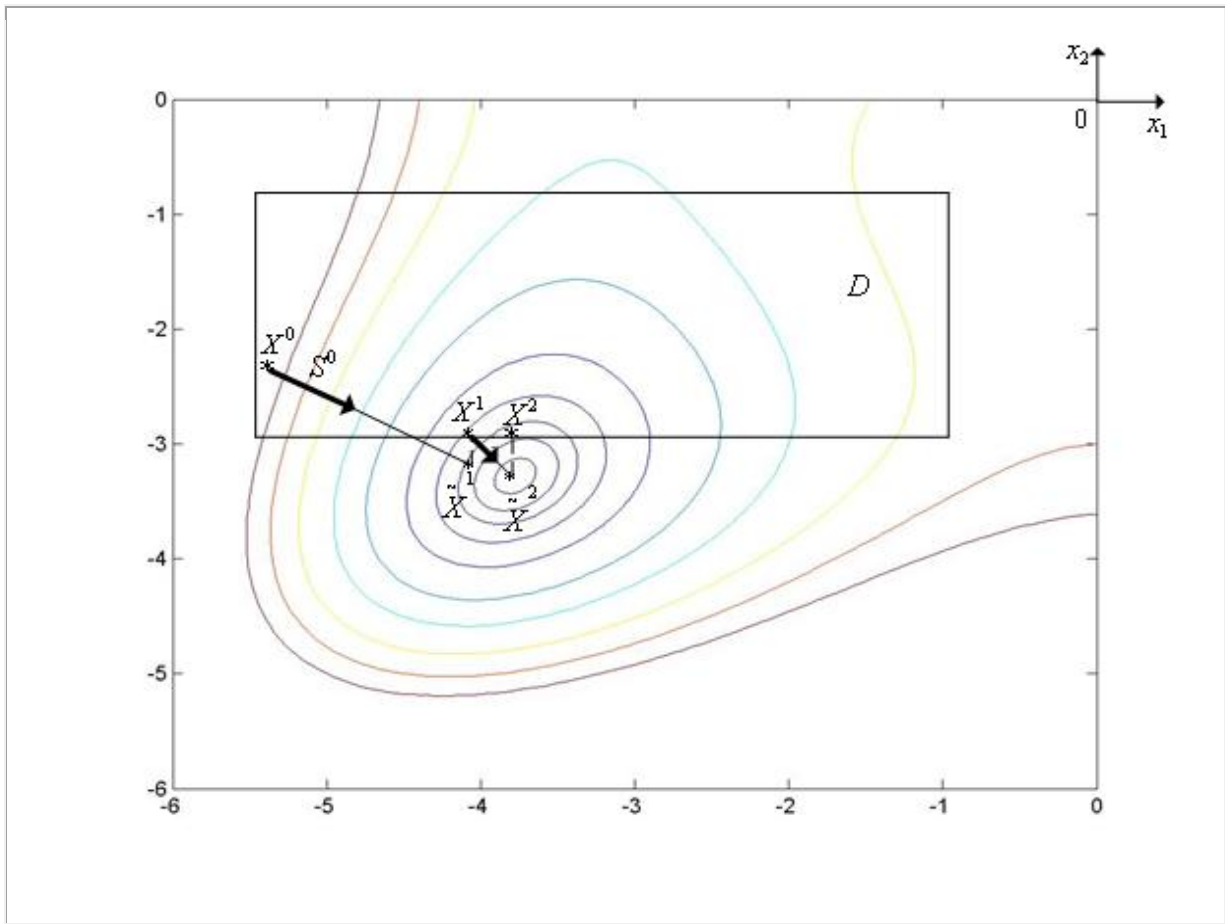


Рис. 4.5.2. Траектория поиска минимума функции Химмельблау комбинацией метода проекции градиента и градиентного метода с дроблением шага

4.6 Метод сведения к совокупности вложенных задач глобальной одномерной минимизации

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

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

где множество допустимых значений

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

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

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

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

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

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

Поясним смысл метода с помощью примера.

Пример 4.6.1. Положим, что $\Phi(X) = \Phi(x_1, x_2)$ и

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

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

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

где $D(x_1)$ — сечение области D прямой, параллельной оси Ox . Задача (4.6.5) представляет собой одномерную задачу глобальной оптимизации критерия оптимальности $\Phi(x_1)$ по параметру $x_1 \in [x_1^L, x_1^U]$, для вычисления значения которого при данном фиксированном x_1 необходимо решить одномерную задачу глобальной оптимизации критерия оптимальности $\Phi(x_1, x_2)$ по параметру $x_2 \in D(x_1)$

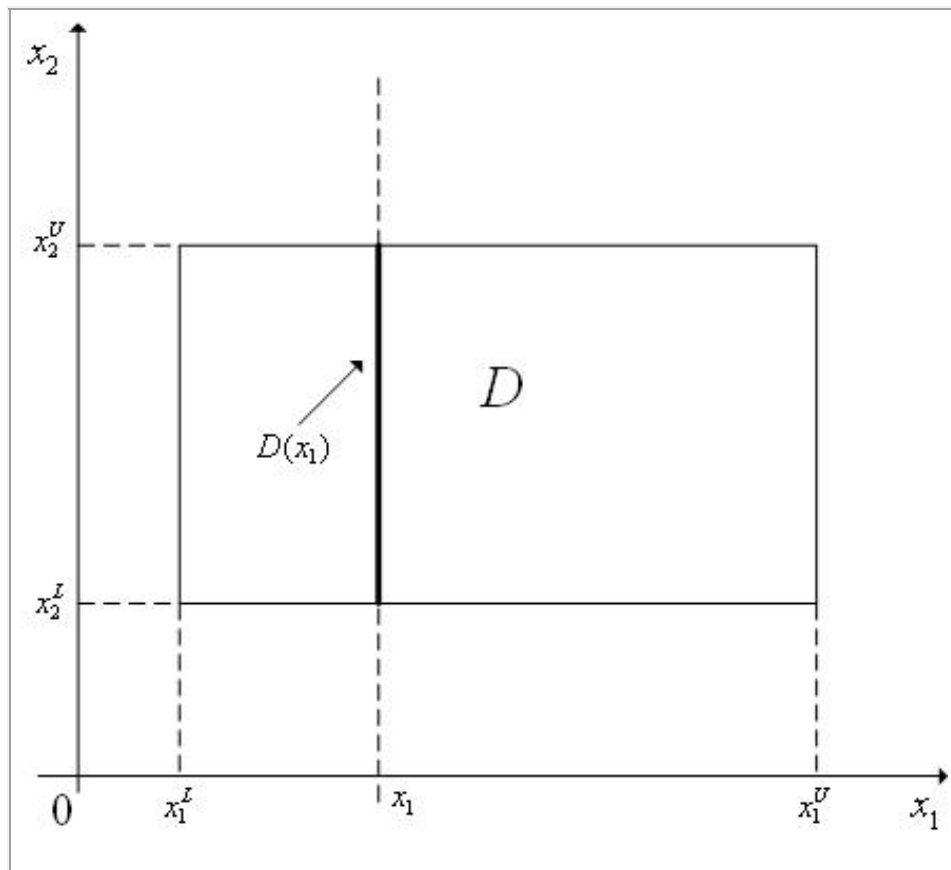


Рис. 4.6.1. К примеру 4.6.1. При решении задачи (4.6.5) вычисление значения критерия оптимальности $\Phi(X)$ при некотором $x = x_1$ требует решения задачи минимизации (4.6.6) на множестве $D(x_1)$

Положим, что для решения всех вложенных одномерных задач глобальной оптимизации (4.6.4) используется метод случайного поиска.

Обозначим N_k число испытаний, необходимых для отыскания методом перебора с заданной точностью глобального минимума функции $\Phi(x_1, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_n)$ по параметру $x_k \in [x_k^L, x_k^U]$ (когда параметры $x_1, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_n$ фиксированы). Тогда общее количество испытаний для решения задачи (4.6.1), (4.6.3) равно, очевидно, $N = \prod_{k=1}^n N_k$.

Поэтому при $n > 3$ такой алгоритм становится неэффективным. При $n \leq 3$ надежность алгоритма достаточно высока, а затраты на поиск значительно меньше затрат на полный перебор на той же сетке.

Метод решения многомерной задачи глобальной условной оптимизации путем сведения к совокупности вложенных одномерных задач глобальной оптимизации может быть скомбинирован со всеми рассмотренными в параграфах 2.7-2.10 методами решения одномерных задач глобальной оптимизации. Рассмотрим комбинацию этого метода с методом случайного поиска для двумерной задачи (4.6.1), (4.6.3).

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

1. Задаем величины N_1, N_2 – количества испытаний при решении задач (4.6.5), (4.6.6), соответственно. Полагаем $r = 1$.

2. Генерируем с помощью какого-либо программного генератора случайных чисел, равномерно распределенных в интервале $[x_1^L, x_1^U]$, случайное число $x_1^0 = x_1^*$.

3. Методом случайного поиска решаем задачу (4.6.6) при $x_1 = x_1^*$ – находим точку $x_2^0 = x_2^*$ и вычисляем значение критерия оптимальности $\Phi^* = \Phi(x_1^*, x_2^*)$.

4. Аналогично п.2 генерируем случайное число x_1^r .

5. Методом случайного поиска решаем задачу (4.6.6) при $x_1 = x_1^r$ – находим точку \tilde{x}_1^r и вычисляем значение критерия оптимальности $\Phi^r = \Phi(x_1^r, \tilde{x}_2^r)$.

6. Если $\Phi^r < \Phi^*$, то выполняем присваивания $x_1^* = x_1^r$, $x_2^* = \tilde{x}_2^r$, $\Phi^* = \Phi^r$.

7. Если $r_1 < N_1$, то выполняем присваивание $r_1 = r_1 + 1$ и переходим на п.4.

Иначе принимаем точку (x_1^*, x_2^*) в качестве приближенного значения точки глобального минимума функции $\Phi(X)$ в области D или каким-либо из рассмотренных ранее методов организуем в окрестности указанной точки поиск локального минимума функции $\Phi(X)$ и заканчиваем вычисления.

Отметим еще раз, что рассмотренный метод, как и любой другой метод глобальной оптимизации, при отсутствии априорной информации о свойствах минимизируемой функции не гарантирует нахождение глобального минимума.

Комбинацию рассматриваемого метода с методом случайного поиска для двумерной задачи иллюстрирует рис. 4.6.2, на котором показан фрагмент линий уровня функции Химмельблау ($n = 2$).

Принято, что X^* – точка минимума функции $\Phi(X)$ в области D после $(r-1)$ -ой итерации. Точки на прямой $x_1 = x_1^r$ случайным образом сгенерированы на r -ой итерации.

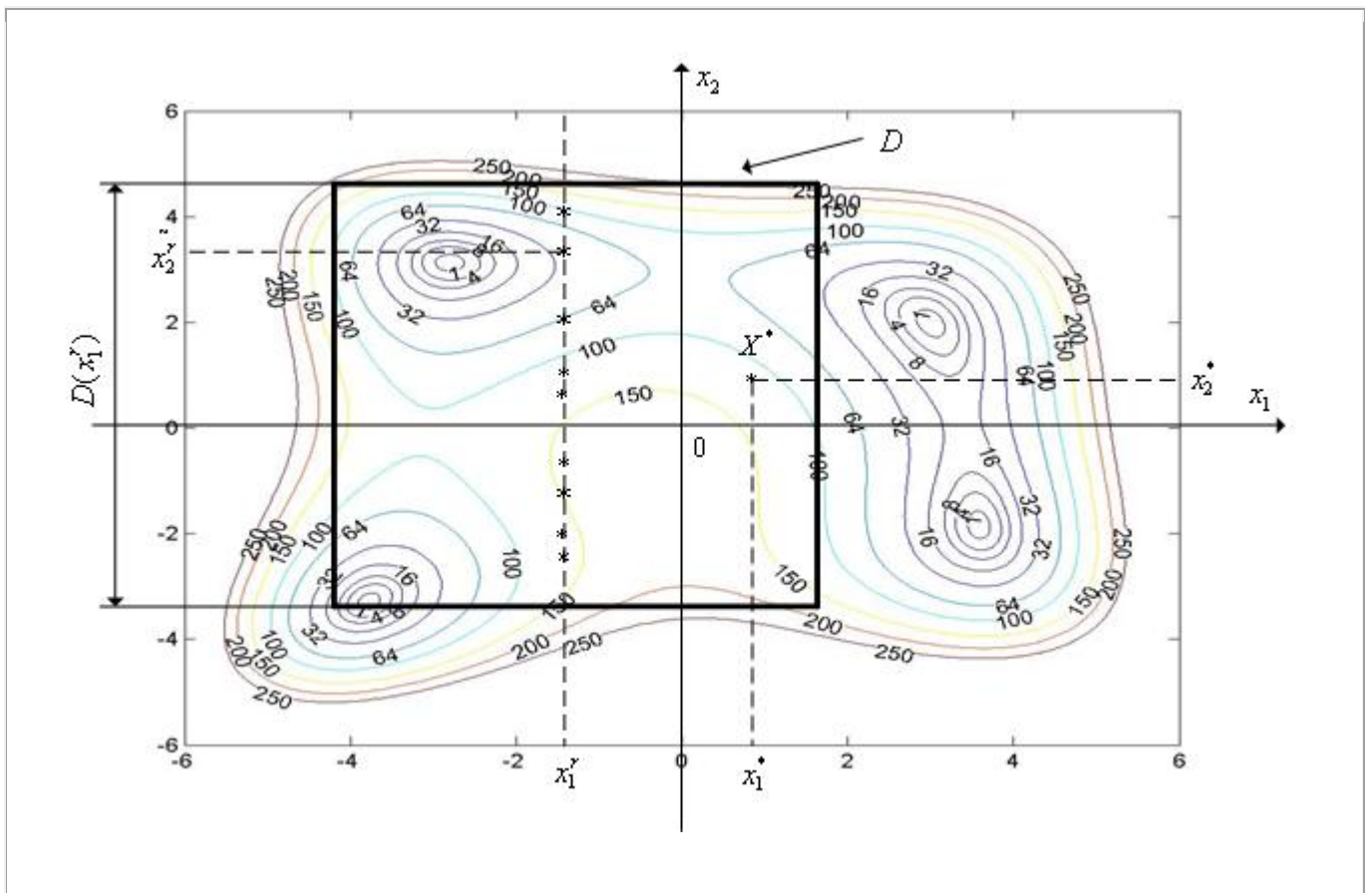


Рис. 4.6.2. Итерация номер r комбинации метода сведения с методом случайного поиска для двумерной задачи

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

Рассмотрим многомерную задачу глобальной условной оптимизации:

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

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

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

Отметим, что произвольный гиперпараллелепипед с помощью линейного преобразования может быть сведен к гиперкубу (4.7.2), так что рассмотрение в качестве множества D гиперкуба (4.7.2), а не гиперпараллелепипеда, не сужает общности рассуждений.

Рассматриваемый метод основан на использовании непрерывного отображения гиперкуба D на отрезок вещественной оси.

Разбиение гиперкуба. Развертка Пеано

Шаг 1 ($s=1$). Координатными плоскостями гиперкуб D разбивается на 2^n ги-

перкубов первого разбиения с длиной ребра, равной $\frac{1}{2} = \left(\frac{1}{2}\right)^{s=2}$ (см.

рис. 4.7.1(а)). Пронумеруем их с помощью переменной $z_1 \in [0, 2^n - 1]$ таким образом, чтобы гиперкубы с номерами, отличающимися на единицу, имели общую грань. Соединим центры гиперкубов ломаной Π_1 в порядке введенной нумерации. Гиперкуб первого разбиения с номером z_1 обозначим $D(z_1)$.

Шаг 2 ($s=2$). По рассмотренной схеме каждый гиперкуб первого разбиения разобьем плоскостями, параллельными координатным плоскостям и проходящими через его центр, на 2^n гиперкубов второго разбиения с длиной ребра, рав-

ной $\frac{1}{4} = \left(\frac{1}{2}\right)^{s=2}$ (см. рис. 4.7.1(б)). Пронумеруем полученные гиперкубы с помощью переменной $z_2 \in [0, 2^n - 1]$ по тому же правилу, что и гиперкубы первого разбиения, с тем отличием, что нулевой гиперкуб второго разбиения, входящий в гиперкуб $D(z_1)$, должен иметь общую грань с $(2^n - 1)$ -м гиперкубом второго разбиения, входящим в гиперкуб $D(z_1 - 1)$. Соединим центры гиперкубов ломаной Π_2 в порядке введенной нумерации. Обозначим гиперкубы второго разбиения $D(z_1, z_2)$.

.....

Шаг S . Аналогично шагу 2 разбиваем гиперкубы $(s - 1)$ -го разбиения на гиперкубы S -го разбиения с длиной ребра, равной $\left(\frac{1}{2}\right)^s$, нумеруем их с помощью переменной $z_s \in [0, 2^n - 1]$, соединяем центры гиперкубов ломаной Π_s в порядке введенной нумерации и обозначаем $D(z_1, z_2, \dots, z_s)$.

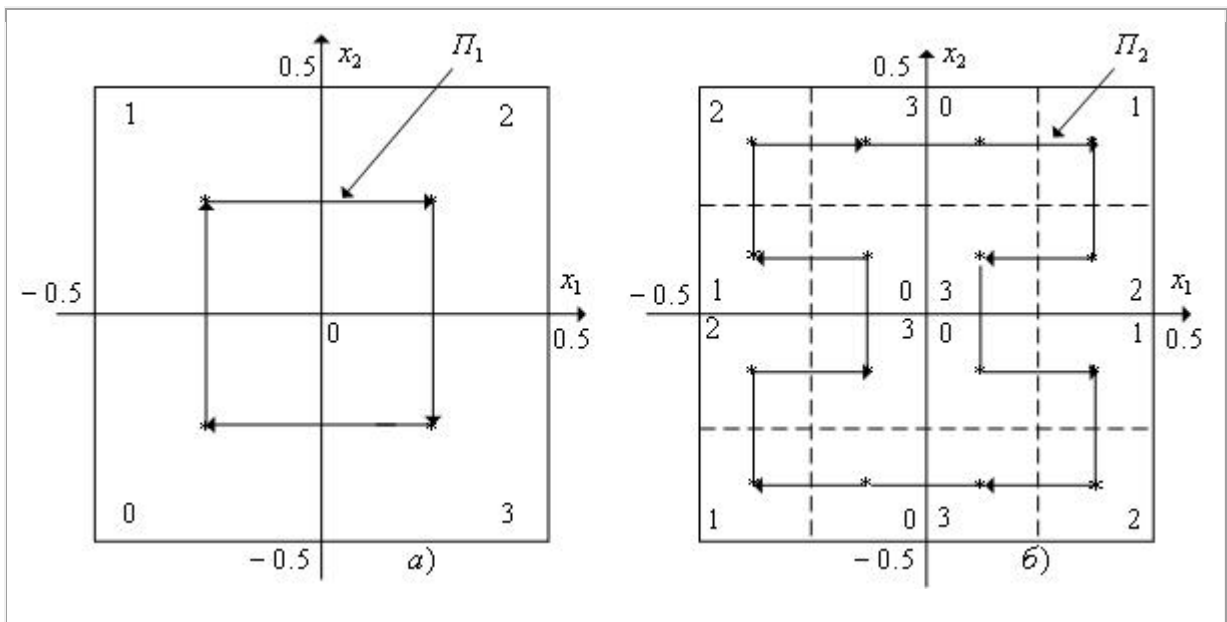


Рис. 4.7.1. К разбиению гиперкуба ($n = 2$). а) Первое разбиение. б) Второе разбиение. Стрелками показано направление нумерации гиперкубов

Ломаная Π_s называется **разверткой Пеано**. В пределе при $s \rightarrow \infty$ ломаная Π_s называется **кривой Пеано**. Кривая Пеано обладает тем свойством, что проходит через все точки гиперкуба и имеет в каждой точке излом.

Разбиение отрезка $[0, 1]$

Шаг 1 ($s=1$). Разобьем отрезок $[0,1]$ на 2^n равных частей длиной $\left(\frac{1}{2^n}\right)^{s=1}$ (см. рис. 4.7.2(а)), пронумеруем их слева направо с помощью переменной $z_1 \in [0, 2^n - 1]$ и обозначим Δz_1 .

Шаг 2 ($s=2$). Каждый из отрезков Δz_1 , $z_1 \in [0, 2^n - 1]$ разобьем на 2^n равных частей длиной $\left(\frac{1}{2^n}\right)^{s=2}$ (см. рис. 4.7.2(б)), пронумеруем их слева направо с помощью переменной $z_2 \in [0, 2^n - 1]$ и обозначим $\Delta(z_1, z_2)$.

.....

Шаг S . Аналогично шагу 2 каждый из отрезков $(s-1)$ -го разбиения разобьем на 2^n равных частей длиной $\left(\frac{1}{2^n}\right)^s$, пронумеруем их слева направо с помощью переменной $z_n \in [0, 2^n - 1]$ и обозначим $\Delta(z_1, z_2, \dots, z_n)$.

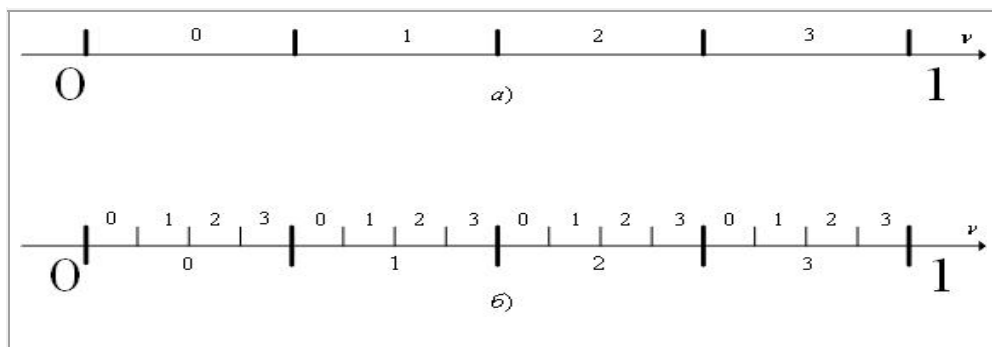


Рис. 4.7.2. К разбиению отрезка $[0,1]$. а) Первое разбиение. б) Второе разбиение

Отображение отрезка $[0,1]$ на гиперкуб

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

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

В пределе при $s \rightarrow \infty$ введенное отображение отображает отрезок $[0,1]$ на кривую Пеано. Можно показать, что в пределе при $s \rightarrow \infty$ построенное отображение является непрерывным и взаимнооднозначным.

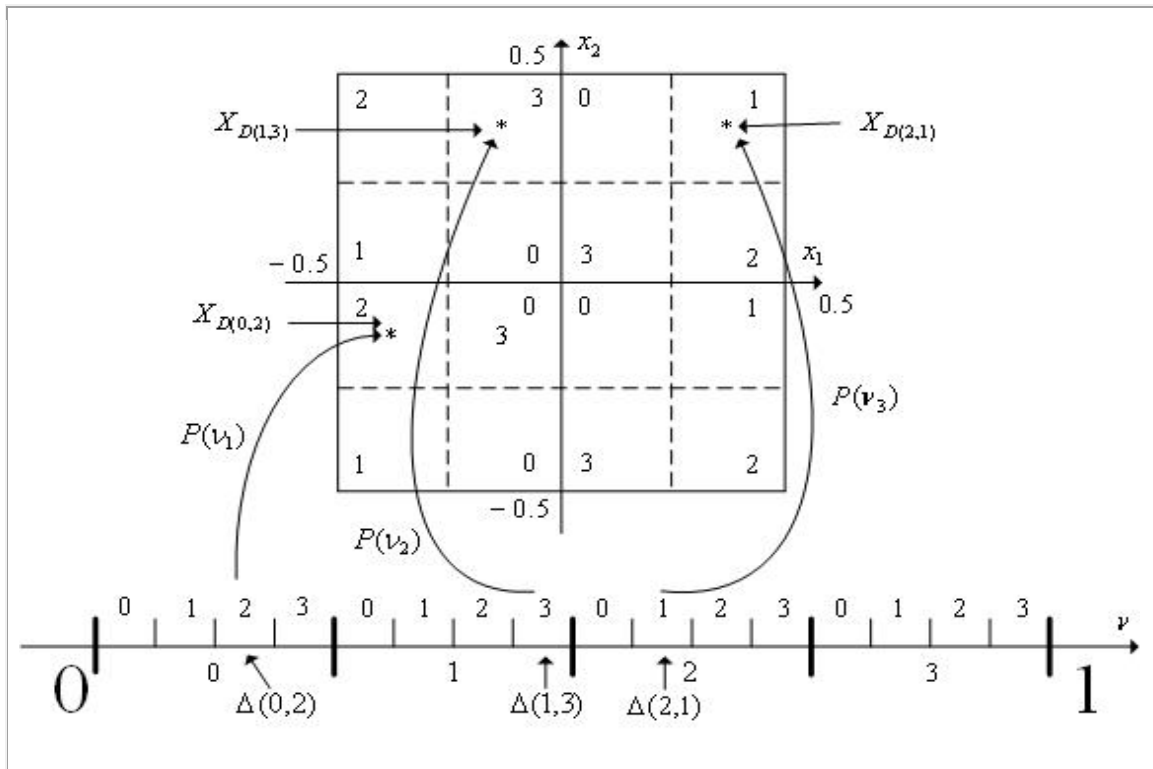


Рис. 4.7.3. К отображению отрезка $[0,1]$ на гиперкуб

Пусть $0, \alpha_1 \alpha_2 \dots \alpha_k \dots$ – двоичное представление числа V , т.е.
 $V_b = 0, \alpha_1 \alpha_2 \dots \alpha_k \dots$

Утверждение 4.7.1. Если $V \in \Delta(z_1, z_2, \dots, z_s)$, то первые sn двоичных цифр числа $V_b = 0, \alpha_1 \alpha_2 \dots \alpha_k \dots$ определяют разбиение $\Delta(z_1, z_2, \dots, z_s)$ отрезка $[0, 1]$:

$$z_1 = \langle \alpha_1 \alpha_2 \dots \alpha_n \rangle;$$

$$z_2 = \langle \alpha_{n+1} \alpha_{n+2} \dots \alpha_{2n} \rangle;$$

.....

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

Здесь $\langle * \rangle$ – операция преобразования двоичного числа в десятичное.

Пример 4.7.1. Пусть область D представляет собой квадрат ($n=2$). На отрезке $[0, 1]$ рассмотрим точки $V_1 = 0.26$, $V_2 = 0.49$, $V_3 = 0.74$ – см. рис. 4.7.4.

реобразуем $V_1 = 0.26$ в двоичную систему счисления:

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

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

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

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

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

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

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

Итого, $V_1 = 0.26 = 0.0100001\dots_b$,

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

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

Таким образом, $V_1 \in \Delta(z_1, z_2) = \Delta(1, 0)$.

Аналогично $v_2 = 0.49 = 0.0111111\dots_b$, $z_1 = \langle \alpha_1, \alpha_2 \rangle = \langle 01 \rangle = 1_d$,
 $z_2 = \langle \alpha_3, \alpha_4 \rangle = \langle 00 \rangle = 3_d$. Таким образом, $v_2 \in \Delta(z_1, z_2) = \Delta(1, 3)$. И
 $v_3 = 0.74 = 0.1011101\dots_b$, $z_1 = \langle \alpha_1, \alpha_2 \rangle = \langle 10 \rangle = 2_d$, $z_2 = \langle \alpha_3, \alpha_4 \rangle = \langle 11 \rangle = 3_d$.

Таким образом, $v_3 \in \Delta(z_1, z_2) = \Delta(2, 3)$.

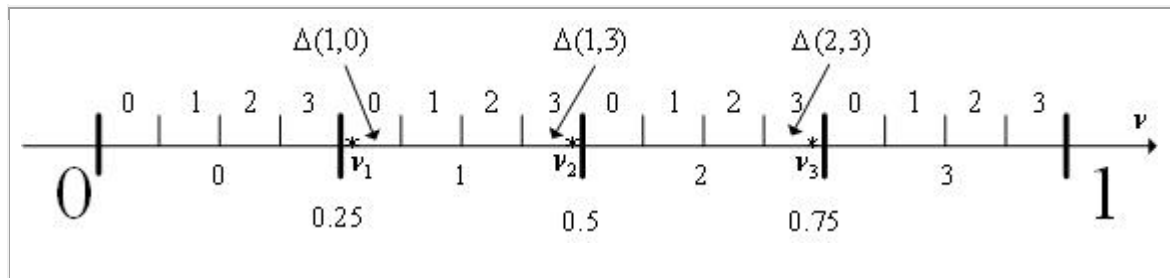


Рис. 4.7.4. К примеру 4.7.1

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

Таким образом, с помощью развертки Пеано многомерная задача глобальной условной оптимизации (4.7.1), (4.7.2) сводится к одномерной задаче условной глобальной оптимизации $\min_{v \in [0, 1]} \Phi(v) = \Phi(v^*) = \Phi^*$.

Решение задачи многомерной глобальной условной оптимизации с помощью развертки Пеано

Метод решения многомерной задачи глобальной условной оптимизации с использованием развертки Пеано называется **методом развертки Пеано** и может быть скомбинирован со всеми рассмотренными в параграфах 2.7-2.10 методами решения одномерных задач глобальной оптимизации. При этом тот факт, что фактически решается задача не одномерной, а многомерной оптимизации, вносит следующие особенности в указанные методы:

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

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

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

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

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

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

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

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

При заданной точности ε_x решения задачи (4.7.1), (4.7.2) по X необходимое количество S разбиений гиперкуба D может быть найдено из следующих соображений. Гиперкуб S -го разбиения $D(z_1, z_2, \dots, z_s)$ имеет длину ребра, равную $\left(\frac{1}{2}\right)^s$. Максимальное расстояние точек этого гиперкуба до его центра равно половине диагонали гиперкуба, которая, очевидно, равна корню квадратному и

суммы квадратов n ребер гиперкуба, т.е. $n \left(\frac{1}{2}\right)^{2s}$.

Таким образом, S может быть найдено из условия $n \left(\frac{1}{2}\right)^{2s} \leq \varepsilon_x$.

Отметим еще раз, что рассмотренный метод, как и любой другой метод глобальной оптимизации, при отсутствии априорной информации о свойствах минимизируемой функции не гарантирует нахождение глобального минимума.

4.8 Метод Монте-Карло

Рассмотрим многомерную задачу глобальной условной оптимизации:

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

где множество допустимых значений

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

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

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

Схема метода Монте-Карло

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

Отметим, что в простейшем случае точки $X^r \in D$ генерируются равномерно распределенными в области D . С целью сокращения вычислительных затрат и при наличии априорной информации о положении точки глобального минимума, целесообразно использовать законы распределения, в которых вероятность гене-

рации точки в окрестности предполагаемого глобального минимума выше, чем вне этой окрестности.

Для локализации с помощью метода Монте-Карло глобального минимума с высокой вероятностью и точностью, требуется очень большое количество испытаний N . Поэтому метод Монте-Карло обычно комбинируют с каким-либо детерминированным методом локальной оптимизации.

Комбинация метода Монте-Карло с детерминированным методом локальной оптимизации

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

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

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

СПИСОК ЛИТЕРАТУРЫ

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

Миссия университета – генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки, технологий и других областей для содействия решению актуальных задач.

КАФЕДРА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ И УПРАВЛЕНИЯ ИННОВАЦИЯМИ

Кафедра была основана в 2007 г. Основная цель кафедры - подготовка грамотных специалистов в области таможенного дела и логистики, что, с одной стороны, отвечает потребностям времени. С 2006 г. кафедра ИСиУИ является платформой опорной организации Роспатента в Северо-Западном федеральном округе. Кафедра ведет учебную, научную, проектную и международную деятельность, которая координируется и экспертируется Экспертным Советом по интеллектуальной собственности Северо-Западного федерального округа. Учебная деятельность ведется по программам магистратуры, программам дополнительного профессионального образования и программам Летних школ. Научно-исследовательская деятельность кафедры ведется в рамках открытой в 2008 г. научной школы "Модернизация инновационной среды в целях эффективного развития российской экономики", основанной в 2008 г. профессором, д.э.н., Е.Л.Богдановой и Секции «Интеллектуальная собственность и инноватика» Дома Учёных им. М.Горького Российской Академии Наук, которая функционирует с 2011 г. С сентября 2011 г. кафедра ИСиУИ входит в проект Роспатента и ВОИС по созданию сети Центров поддержки технологий и инноваций – Technology and Innovation Support Centers, целью которых является упрощение доступа к техническим знаниям и повышение эффективности использования патентной информации в ряде стран, в региональных и областных центрах научно-технической информации. Работа кафедры по данному направлению представлена в справочнике Роспатента. В 2011 г. почетным знаком Роспатента "Во благо России" награждена заведующая кафедрой ИСиУИ Богданова Е.Л. В 2014 г. благодаря активному содействию кафедры Университет был награжден высшей наградой Всемирной организации интеллектуальной собственности для Инновационных Предприятий (WIPO Trophy for Innovative Enterprises) в качестве признания вклада Университета в развитие инновационного и технического творчества и содействия развитию и охраны интеллекту-

альной собственности мира. В 2015 г. ведущие преподаватели кафедры награждены дипломами, серебряными и золотыми орденами Салона изобретений и инновационных технологий "Архимед 2015", а в 2016 г. – серебряными и золотыми орденами Салона изобретений и инновационных технологий "Архимед 2016", Почетный знак Салона "Гранд Архимед" вручен заведующей кафедрой ИСиУИ Богдановой Е.Л.

Богданова Елена Леонардовна
Соловейчик Кирилл Александрович
Аркина Ксения Георгиевна

**Оптимизация в проектном менеджменте:
нелинейное программирование**

Учебное пособие

В авторской редакции
Редакционно-издательский отдел Университета ИТМО
Зав. РИО Н.Ф. Гусарова
Подписано к печати
Заказ №
Тираж
Отпечатано на ризографе

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверкский пр., 49