МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Пензенский государственный университет
архитектуры и строительства»
(ПГУАС)

В.В. Кузина, А.Н. Кошев

ЧИСЛЕННЫЕ МЕТОДЫ
И МЕТОДЫ ОПТИМИЗАЦИИ

Рекомендовано Редсоветом университета
в качестве учебного пособия для студентов,
обучающихся по направлению подготовки 09.03.02
«Информационные системы и технологии»

Пенза 2016

УДК 519.6 (075.8)
ББК 22.193я73
К89

Рецензенты: доктор технических наук, профессор И.А. Прошин (ПензГТУ);
доктор технических наук, профессор А.М. Данилов (ПГУАС)

Кузина В.В.

К89 Численные методы и методы оптимизации: учеб. пособие для направления подготовки 09.03.02 «Информационные системы и технологии» / В.В. Кузина, А.Н. Кошев. – Пенза: ПГУАС, 2016. – 60 с.

Учебное пособие посвящено теоретическим основам численных методов и методов оптимизации.

Учебное пособие подготовлено на кафедре «Информационно-вычислительные системы» и предназначено для студентов, обучающихся по программе подготовки академического бакалавриата по направлению 09.03.02 «Информационные системы и технологии» очной формы обучения при изучении дисциплины «Численные методы и методы оптимизации».

ПРЕДИСЛОВИЕ

Предлагаемое учебное пособие предназначено для студентов бакалавриата, обучающихся по направлению 09.03.02 «Информационные системы и технологии» при изучении дисциплины «Численные методы и методы оптимизации».

Пособие разработано на кафедре «Информационно-вычислительные системы» ФГБОУ ВО ПГУАС в соответствии с рабочей программой дисциплины и требованиями ФГОС ВО.

Цель дисциплины – формирование у обучающихся базы для развития профессиональных компетенций в области численных методов и методов оптимизации, а именно, овладение численными методами решения задач оптимизации целевых функций без ограничений и с ограничениями различного вида с целью их дальнейшего применения в профессиональной деятельности.

Основные задачи дисциплины:

– приобретение обучающимися знания о численных методах алгебры и математического анализа, об оптимизации целевых функций без ограничений и с ограничениями различного вида, их применимости в профессиональной деятельности;

– формирование умения применять методы оптимизации для отыскания точек экстремума функций без ограничений и при их наличии;

– выработка умения реализовывать численные методы решения задач оптимизации в различных интегрированных математических средах (MathCad, Matlab и др.).

Процесс изучения дисциплины направлен на формирование следующих компетенций:

– владение широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий;

– способность использовать основные законы естественнонаучных дисциплин в профессиональной деятельности, применять методы математического анализа и моделирования, теоретического и экспериментального исследования.

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенции):

знать:

– математический аппарат современных численных методов;

– основные положения и методы задач линейного, выпуклого и нелинейного программирования, о приложениях теории в информатике, программировании и вычислительной технике;

– методы математического программирования и математического моделирования;

– современные математические пакеты программ для решения задач математического программирования;

уметь:

– реализовать численные методы решения задач оптимизации на ПЭВМ;

– решать типовые задачи;

– использовать встроенные функции математических пакетов для решения задач оптимизации;

– программировать вычислительные алгоритмы и решать типовые задачи на компьютере;

владеть:

– базовыми знаниями и навыками численных методов решения задач оптимизации;

– навыками решения проблемных задач, требующих применения логико-математического аппарата;

– навыками работы в интегрированных математических средах, навыками работы с прикладными математическими пакетами программ;

– навыками решения проблемных задач, используя вычислительный эксперимент;

иметь представление:

– о математическом аппарате современных численных методов;

– об основных положениях и методах решения задач оптимизации, о приложениях теории в информатике, программировании и вычислительной технике;

– о применении методов математического программирования в профессиональной деятельности;

– о математическом моделировании и вычислительном эксперименте.

ВВЕДЕНИЕ

Дисциплина изучается в четвертом семестре и завершается зачетом.

Курс «Численные методы и методы оптимизации» включает 18 часов лекционных, 38 часов лабораторных занятий и 52 часа – самостоятельной работы студента, в рамках которой он выполняет расчетно-графическую работу (РГР).

Для успешного освоения курса рекомендуется следующее.

– Внимательно слушать лекции и записывать их основные положения; отвечать на поставленные во время лекции вопросы; читать необходимую литературу, выполнять лабораторные работы.

– При прослушивании и проработке лекций особое внимание следует уделить терминологии, используемой в дисциплине и основным понятиям.

– Записывать следует только основные положения. Необходимо активно участвовать в обсуждении тем, предлагаемых преподавателем, высказывать собственные соображения.

Подготовка к зачету заключается в изучении и тщательной проработке обучающимся учебного материала дисциплины с учётом учебного пособия, лекционных и лабораторных занятий, сгруппированном в виде контрольных вопросов.

Преподавателю предоставляется право воспользоваться примерными тестовыми заданиями или составить новые тестовые задания в полном соответствии с материалом учебной дисциплины.

Для лучшего усвоения учебного материала и удобства подготовки студентов к зачету преподавателями данного курса – авторами учебного пособия – на файл-сервере кафедры представлен комплекс материалов: полно-текстовые файлы лекционного материала, методические указания по выполнению РГР, по подготовке к лабораторным работам и всем видам контроля (входного, текущего, промежуточного).

Студент, пропустивший занятия по уважительной причине или при отсутствии таковой, обязан отработать материал самостоятельно.

К зачету допускается студент, успешно выполнивший все предусмотренные программой курса лабораторные работы и РГР и защитивший их в компьютерном классе, продемонстрировав преподавателю электронный вариант и сдавший печатный либо рукописный вариант отчета.

Зачет по курсу проводится, как правило, в два этапа: 1) в виде компьютерного тестирования, включающего как теоретические, так и практические задания по всему пройденному материалу, в том числе выносимого на самостоятельное изучение; 2) по билетам, включающим один теоретический вопрос и задачу.

Студент, набравший при тестировании 95 – 100% может быть освобожден от второго этапа по усмотрению преподавателя.

На 2-м этапе студент даёт ответы на вопросы билета после предварительной подготовки. Студенту предоставляется право отвечать на вопросы билета без подготовки по его желанию.

Преподаватель имеет право задавать дополнительные вопросы, если студент недостаточно полно осветил тематику вопроса, если затруднительно однозначно оценить ответ, если студент не может ответить на вопрос билета, если студент отсутствовал на занятиях в семестре.

1. ВВЕДЕНИЕ В ЧИСЛЕННЫЕ МЕТОДЫ И МЕТОДЫ ОПТИМИЗАЦИИ

1.1. Общие сведения. Классификация задач математического программирования

Курс «Численные методы и методы оптимизации» ориентирован на изучение материала об использовании методов оптимизации различных порядков при решении задач математического программирования.

Методы оптимизации (МО) составляют один из разделов науки об исследовании операций.

Задачу оптимизации можно определить как задачу поиска наилучшего решения из множества возможных решений (альтернатив) на основе некоторого критерия оптимальности. Такой выбор может быть осуществлен различными способами. Один из подходов заключается в количественной оценке каждого возможного решения и выборе наилучшего (минимального или максимального) варианта решения.

Предмет и содержание курса по численным методам и методам оптимизации заключается в изучении теории и методов решения задач отыскания экстремумов функций на множествах, определенных линейными и нелинейными ограничениями, заданными в виде числовых равенств и неравенств.

МО широко используются на практике, и прежде всего при решении задач:

- оптимального проектирования:
  - выбор наилучших технологических режимов, элементов конструкций, структуры технологических цепочек, условий экономической деятельности;
  - повышение доходности и т.д.;
- оптимального управления:
  - построение линейных и нелинейных математических моделей объектов управления;
  - распределение материальных и финансовых ресурсов;
  - минимизация невязок различной структуры модели и реального объекта,

а также многих других задач из технических, экономических, строительных и социальных областей знаний, таких, как управление запасами, трудовыми ресурсами, транспортными потоками, техническими системами и т.п.

МО находят применение при решении таких задач, для которых оптимальное развитие рассматриваемого процесса или явления можно сформулировать как достижение экстремума некоторого показателя. Поиск оптимального решения является одной из существенных проблем в исследовании операций.

Термин «*математическое программирование*» объединяет классы задач по отысканию экстремумов функций нескольких переменных определенного вида при заданных ограничениях на свободные переменные (независимые переменные, параметры оптимизации), в математической записи которых участвуют функции того же вида.

Задачу оптимизации можно определить как задачу поиска наилучшего решения из множества возможных решений (альтернатив) на основе некоторого критерия оптимальности. Такой выбор может быть осуществлен различными способами. Один из подходов заключается в количественной оценке каждого возможного решения и выборе наилучшего (минимального или максимального) варианта решения.

Существуют различные способы классификации задач математического программирования (ЗМП). Чаще всего за основу классификации выбирают тип исследуемой функции (целевой функции, критерия оптимизации) и характер допустимого множества. Именно эти особенности задачи оптимизации определяют выбор методов ее решения.

Различают *задачи безусловной оптимизации*, когда на входные переменные не накладывается никаких ограничений, и *задачи условной оптимизации*: *с ограничениями типа равенства* и *ограничениями типа неравенства*, описывающими множество допустимых значений параметров оптимизации.

По виду целевой функции и функций ограничений задачи математического программирования подразделяют на *задачи линейного программирования* (ЗЛП) и *задачи нелинейного программирования* (ЗНП), к которым, в свою очередь, относятся: задачи выпуклого, квадратичного, дробно-линейного, сепарабельного, геометрического программирования.

В ЗЛП линейными являются и целевая функция, и ограничения. Если целевая функция линейна, то область определения ее совпадает с n-мерным арифметическим пространством R^n, и она может не достигать конечного экстремума в области определения. Поэтому в задаче оптимизации с линейной целевой функцией, как правило, должны присутствовать ограничения на свободные переменные.

Задача оптимизации является нелинейной, если нелинейной является целевая функция или хотя бы одна из функций в записи ограничений.

Среди ЗНП выделяется класс *задач выпуклого программирования* (ЗВП), в которых и целевая функция, и ограничения являются выпуклыми функциями, определенными на выпуклом множестве.

В *задаче квадратичного программирования* ограничения линейные, а целевая функция является многочленом второй степени.

В *задаче дробно-линейного программирования* ограничения линейные, а целевая функция представляет собой отношение двух линейных функций. Задача должна включать ограничения, поскольку область определения целевой функции не совпадает с R^n.

В *задаче сепарабельного программирования* целевая функция и ограничения являются *сепарабельными функциями*, представляющими собой сумму функций одной переменной

$$f(x)=\sum_{j=1}^{n}\varphi_i(x_j),\quad x=(x_1,\ x_2,\ ...,\ x_n).$$

В *задаче геометрического программирования* целевая функция и ограничения являются *позиномами*, т.е. функциями вида

$$f(x)=\sum_{i=1}^{m}c_i\prod_{j=1}^{n}x_j^{a_{ij}},\quad x=(x_1,\ x_2,\ ...,\ x_n),\quad x_j>0,$$

где коэффициент $c_i>0,\quad i=1,2,\ldots,m$, а у функций $x_j^{a_{ij}},\quad j=1,2,\ldots,n$, показатели степеней a_{ij} могут быть любыми действительными числами (не только целыми неотрицательными, как у полиномов). Область определения таких функций ограничена строго положительными вещественными числами.

По размерности задачи математического программирования можно разделить на *одномерные* и *многомерные*.

К отдельному классу задач относят *задачи дискретного программирования*, в которых параметры оптимизации могут принимать дискретное, в частности конечное, множество значений. Среди них – *задачи целочисленного программирования*.

Выделяют классы задач параметрического, стохастического и динамического программирования.

В *задачах параметрического программирования* целевая функция или функции ограничений либо и то, и другое зависят от некоторых параметров.

Если в целевой функции или в функциях ограничений на переменные содержатся случайные величины, то такая задача относится к *задаче стохастического программирования*.

Задача динамического программирования – задача, процесс нахождения решения которой является многоэтапным.

1.2. Основные определения

Задача математического программирования состоит в том, чтобы найти экстремум (min, max) некоторой функции, зависящей от одной или нескольких переменных при выполнении ряда ограничений на эти переменные, заданных в виде равенств и неравенств:

$$\begin{gathered}\min_{x_1,\,x_2,\,\ldots,\,x_n}\Phi(x_1,x_2,\ldots,x_n);\\ f_k(x_1,\ ...,\ x_n)=0;\quad k=1,\ldots,N;\\ g_j(x_1,\ ...,\ x_n)\le 0;\quad j=1,\ldots,M.\end{gathered}\tag{1}$$

Задачу (1) называют задачей математического программирования, а функцию $\Phi(x_1, x_2, \ldots, x_n)$, определенную на множестве $\Omega \subset R^n$, заданном ограничениями, – *целевой функцией*. Переменные $x_1, x_2, \ldots, x_n$ – это *параметры оптимизации*, каждая точка $x = (x_1, x_2, \ldots, x_n) \in \Omega$ является допустимым решением, а множество $\Omega \subset R^n$ – множеством допустимых решений или *допустимым множеством*. Точка $x^* \in \Omega$, в которой целевая функция принимает экстремальное значение, называется *оптимальным решением*. Обозначим

$$X = (x_1,\ x_2,\ ...,\ x_n);$$

$$F(X) = \begin{pmatrix} f_1(x_1, \ldots, x_n) \\ \vdots \\ f_N(x_1, \ldots, x_n) \end{pmatrix};$$

$$G(X) = \begin{pmatrix} g_1(x_1, \ldots, x_n) \\ \vdots \\ g_M(x_1, \ldots, x_n) \end{pmatrix}.$$

Тогда задачу нелинейного программирования можно записать в векторной форме:

$$\begin{aligned} &\min_X \Phi(X); \\ &F(X) = 0; \\ &G(X) \le 0. \end{aligned} \qquad (2)$$

В дальнейшем частные задачи оптимизации будем рассматривать, в основном, как задачи поиска минимума функции. Однако при решении реальных практических задач очень часто встречаются задачи и на максимум. Легко перейти от одного вида экстремума к другому путем смены знака у критерия оптимальности: $\min\limits_X \Phi(X) = \max\limits_X(-\Phi(X))$.

Точка $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ называется точкой *локального минимума* функции $\Phi(X)$ в некоторой окрестности этой точки, если существует вектор $\Delta = (\delta_1, \delta_2, \ldots, \delta_n)$ такой, что для любых значений x_i, подчиняющихся неравенству $x_i^* - \delta_i \le x_i \le x_i^* + \delta_i$, выполняется неравенство $\Phi(x_1, \ldots, x_n) \ge \Phi(x_1^*, \ldots, x_n^*)$.

Локальный минимум функции, который соответствует самому наименьшему значению функции из области значений, называется *глобальным минимумом*.

Поверхностью уровня функции $\Phi(X)$ называется множество точек, в которых функция принимает постоянное значение, т.е. $\Phi(X) = \text{const}$. Если $n = 2$, поверхность уровня изображается *линией уровня* на плоскости.

Рассмотрим некоторую точку X_0 из области определения функции и вектор $D = (d_1, d_2, \dots, d_n)$, задающий направление в пространстве R^n: $|D| = 1, \ \ d_i \ge 0$.

Совокупность точек $X = X^0 + D\tau$, где τ – некоторый одномерный параметр, образует прямую, если $-\infty \le \tau \le \infty$, и луч, если $0 \le \tau \le \infty$.

Градиентом функции $\Phi(x_1, x_2, \dots, x_n)$ называют вектор частных производных

$$\nabla\Phi(X) = \begin{pmatrix} \dfrac{\partial\Phi}{\partial x_1}(X) \\ \dfrac{\partial\Phi}{\partial x_2}(X) \\ \dots \\ \dfrac{\partial\Phi}{\partial x_n}(X) \end{pmatrix}. \tag{3}$$

Обозначение градиента: $\mathrm{grad}\Phi(x)$ или $\nabla\Phi(x)$, где ∇ – оператор набла.

Вектор $-\nabla\Phi(X)$, противоположный вектору градиента, называется *антиградиентом* функции.

Точка, в которой градиент функции $\nabla f\left(x^{(k)}\right) = 0$, называется *стационарной*.

На рис. 1 изображены линии уровня функции y и вектор градиента в точке M_0 (перпендикулярен линии уровня, проходящей через эту точку). *Модуль градиента* показывает максимальную скорость изменения функции в окрестности M_0, то есть частоту линий уровня. Модуль градиента равен наибольшей производной по направлению в данной точке.

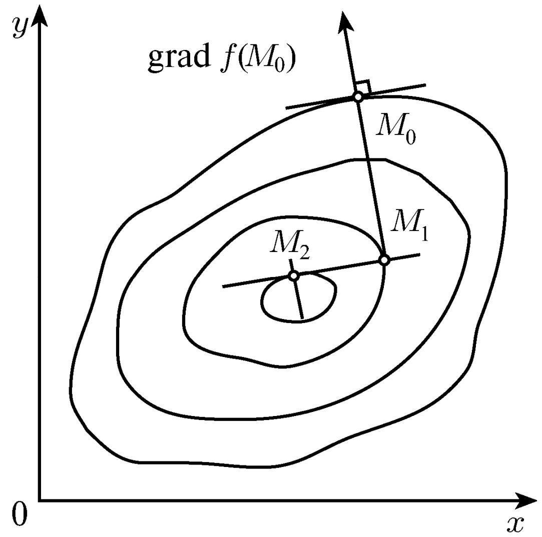


Рис. 1. Линии уровня и градиент функции

1.3. Теоремы о функции, имеющей производную по заданному направлению

Пусть $D=\left(d_1, d_2, \ldots, d_n\right)$ – некоторое заданное направление. Величина, показывающая мгновенную скорость изменения функции Ф вдоль направления *D*, называется *производной от функции Ф в направлении D*:

$$\nabla\Phi^T\left(x^0\right)\cdot D=\lim_{\tau\to 0}\frac{\Phi\left(X^0+D\tau\right)-\Phi\left(X^0\right)}{\tau}. \qquad (4)$$

Теорема 1.

Производная по направлению *D* равна произведению градиента на вектор *D*:

$$\Phi'_D=\nabla\Phi^T\left(X^0\right)\cdot D.$$

Доказательство.

$$\Phi'_D=\lim_{\tau\to 0}\frac{\Phi\left(X^0+D\tau\right)-\Phi\left(X^0\right)}{\tau}.$$

Рассмотрим $\Phi\left(X^0+D\tau\right)-\Phi\left(X^0\right)=\Delta\Phi$ – полное приращение функции $Z=\Phi\left(X\right)$ в направлении *D*.

Известно, что полное приращение функции ΔZ может быть представлено в окрестности точки *X* следующим образом:

$$\Delta Z=\sum_{i=1}^{n}\frac{\partial\Phi}{\partial x_i}\Delta x_i, \qquad (5)$$

$$\Delta x_i=d_i\tau=\cos(\gamma_i)\tau, \qquad (6)$$

где $\cos(\gamma_i)$ – направляющие косинусы единичного вектора *D*.
Подставив выражения (5) и (6) в формулу (4), получим:

$$\Phi'_D=\lim_{\tau\to 0}\frac{\sum_{i=1}^{n}\frac{\partial\Phi}{\partial x_i}\cos(\gamma_i)\tau}{\tau}=\nabla\Phi^T\cdot D.$$

Теорема 2.

Если $\nabla\Phi^T\left(X^0\right)\cdot D>0$, то существует такая величина σ, что $\Phi\left(X^0+D\tau\right)>\Phi\left(X^0\right)$ для любого $\tau\in[0,\sigma]$.

Доказательство.

Первое неравенство в условии теоремы означает, что производная от функции Φ по направлению D положительна.

$$\lim_{\tau\to 0}\frac{\Phi\left(X^0+D\tau\right)-\Phi\left(X^0\right)}{\tau}>0\,.$$

Так как $\tau>0$, то, для того чтобы предел отношения был положительным, необходимо, чтобы числитель был не меньше нуля для всех τ, достаточно близких к нулю, что доказывает утверждение теоремы.

Теорема 3.

Градиент $\nabla\Phi\left(X^0\right)$ показывает направление наискорейшего возрастания функции Φ в точке X^0.

Доказательство.

Рассмотрим все направления $D=\left(d_1, d_2, \ldots, d_n\right)$, в которых наблюдается возрастание функции, т.е.

$$\nabla\Phi^T\left(X^0\right)\cdot D>0\,.$$

Так как $\nabla\Phi^T(X^0)\cdot D=\left|\nabla\Phi^T(X^0)\right|\cdot\left|D\right|\cdot\cos\angle(\nabla\Phi^T, D)$, то свое наибольшее значение эта величина примет при $\cos(\nabla\Phi^T, D)=1$, т.е. когда угол $\angle(\nabla\Phi^T, D)=0$ и, следовательно, направление D совпадает с направлением градиента.

Так как величина производной по направлению показывает скорость возрастания функции в данном направлении, то отсюда следует утверждение теоремы.

Теорема 4.

В точке максимума X^* функции $\Phi(X)$ градиент равен нулю:

$$\nabla\Phi\left(X^*\right)=0\,.$$

Доказательство.

Предположим, что $\nabla\Phi\left(X^*\right)\neq 0$. Зададим направление $D=\nabla\Phi\left(X^*\right)$.

$$\nabla\Phi\left(X^*\right)\cdot D=\nabla\Phi\left(X^*\right)\cdot\nabla\Phi\left(X^*\right)=\left[\nabla\Phi\left(X^*\right)\right]^2>0\,.$$

Следовательно, направление $D=\nabla\Phi\left(X^*\right)$ есть направление, в котором возможно возрастание функции $\Phi(X)$ в сравнении со значением функции в точке X^*, что невозможно, так как X^* – точка максимума. Значит, предположение о том, что $\nabla\Phi\left(X^*\right)\neq 0$, неверно.

Методические рекомендации к теме 1

Знать основные понятия и теоремы, иметь представление о системах классификации задач математического программирования. Самостоятельно проработать материал «Математическая модель, этапы моделирования, вычислительный эксперимент» [15].

Уметь отвечать на вопросы и выполнять задания по теме.

Контрольные вопросы и задания

1. Какие задачи решают и где применяются методы оптимизации?
2. На какие этапы можно разделить процесс принятия решений в исследовании операций?
3. Что такое информационное моделирование?
4. Сформулируйте в общем виде задачу математического программирования.
5. По каким признакам классифицируются задачи математического программирования?
6. Что в задаче математического программирования называют допустимым решением, а что – оптимальным решением?
7. Приведите пример задачи дробно-линейного программирования.
8. Какая функция называется целевой?
9. Чем задача безусловной оптимизации отличается от задачи условной оптимизации?
10. Можно ли задачу поиска максимума функции преобразовать в задачу поиска минимума?
11. Приведите пример задачи математического программирования, не имеющей решения. Можно ли утверждать, что задача математического программирования имеет решение, если допустимое множество замкнуто и ограничено?
12. Как построить линии уровня целевой функции?
13. Что такое градиент функции?
14. Что показывает вектор градиента, антиградиента?
15. Как определить производную по направлению D?
16. Чему равен градиент функции в точке экстремума?
18. Какая точка называется стационарной?

2. ЭЛЕМЕНТЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

2.1. Общая постановка задачи

Рассмотрим множество векторов $X \in R^n$, $X = (x_1, x_2, ..., x_n)$ и функцию $f(x) = \sum_{j=1}^{n} c_j \cdot x_j$, где c_j – некоторые числовые коэффициенты. Функция $f(x) = f(x_1, x_2, \ldots, x_n)$ линейна относительно переменных $x_1, x_2, ..., x_n$, то есть представляет собой линейную форму.

Предположим, что на компоненты векторов *X* наложена система ограничений в виде линейных равенств и неравенств. Совокупность векторов *X*, удовлетворяющих этим ограничениям, составляет некоторое множество $\Omega \subset R^n$.

Говорят, что в пространстве R^n поставлена задача линейного программирования (ЗЛП), если определена линейная целевая функция $f(x)$, подлежащая минимизации, а также задана система линейных ограничений на компоненты векторов *X* в виде равенств и неравенств, определяющая некоторую область $\Omega \subset R^n$:

$$f(x) = \sum_{j=1}^{n} c_j \cdot x_j \to \min_{X \in \Omega}; \tag{7}$$

$$\begin{cases} \sum_{j=1}^{n} a_{ij} \cdot x_j = a_i, & i \in I; \\ \sum_{j=1}^{n} b_{kj} \cdot x_j \le b_k, & k \in K; \\ x_l \ge 0, & l \in L. \end{cases} \tag{8}$$

Здесь *I*, *K*, *L* – заданные целочисленные множества индексов.

Заметим, что решение такой задачи методом приравнивания к нулю частных производных и решения соответствующей системы невозможно, так как все производные равны константам.

2.2. Каноническая форма представления ЗЛП

Задачи линейного программирования возникают в различных областях науки, техники, экономической теории, медицине, социологии и т.д. В зависимости от целей решения, ЗЛП могут быть записаны в том или ином виде, однако существуют стандартные формы представления задач линейного программирования, для которых разработаны методы и алгоритмы решения.

В *канонической форме* ЗЛП все ограничения системы (8) (кроме неравенств, выражающих условие неотрицательности переменных) представлены в виде равенств и требуется найти максимум целевой функции.

Приведение задачи к стандартной или канонической форме легко осуществить, если использовать следующие положения.

1) Задача минимизации может быть преобразована в задачу максимизации путем смены знака у критерия оптимальности: $\min\limits_X f(x) = \max\limits_X\left(-f(x)\right)$ и наоборот: $\max\limits_X f(x) = \min\limits_X\left(-f(x)\right)$.

2) Ограничения типа равенств и неравенств можно привести к необходимой форме, используя следующие приемы:

а) ограничение типа $\geq$ можно привести к ограничению $\leq$ путем умножения обеих частей неравенства на -1;

б) ограничение типа = приводится к двум ограничениям типа неравенств $\leq$ и $\geq$, если потребовать их одновременного выполнения. Например, равенство $x_1 + x_2 = 1$ эквивалентно системе двух неравенств

$$\begin{cases} x_1 + x_2 \leq 1; \\ x_1 + x_2 \geq 1; \end{cases}$$

в) ограничение типа неравенства можно всегда привести к ограничению типа равенства путем добавления еще одной переменной x_{n+1} в левую часть неравенства. Так, например, ограничение $a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + ... + a_{in} \cdot x_n \leq b_i$ можно привести к ограничению типа равенства добавлением положительной переменной x_{n+1}: $a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + ... + a_{in} \cdot x_n + x_{n+1} = b_i$, $x_{n+1} \geq 0$. При этом нужно помнить, что размерность задачи (число переменных x_i) станет на единицу больше, а неизвестная x_{n+1} войдет в качестве слагаемого в целевую функцию $f(x)$ с коэффициентом c_{n+1}, равным нулю ($c_{n+1} = 0$);

г) если не задано ограничение вида $x_i \geq 0$ на некоторую переменную, а оно необходимо для канонической формы, то вместо одной переменной x_i можно рассмотреть две переменные x_i' и x_i'', такие, что $x_i' \geq 0$ и $x_i'' \geq 0$, а $x_i = x_i' - x_i''$.

Используя перечисленные приемы, произвольную задачу линейного программирования можно привести к основной *канонической форме*: максимизировать целевую функцию $f(x) = \sum\limits_{j=1}^{n} c_j \cdot x_j$ по переменным $x_1,...,x_n$ при выполнении ограничений:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + ... + a_{1n} \cdot x_n = b_1; \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + ... + a_{2n} \cdot x_n = b_2; \\ \dotfill \\ a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + ... + a_{mn} \cdot x_n = b_m; \\ x_j \geq 0,\ j = 1,...,n. \end{cases} \tag{9}$$

В матричной форме задача записывается следующим образом:

$$\begin{cases} (C, X) \to \max; \\ A \cdot X^T = B; \\ X \geq 0. \end{cases} \qquad (10)$$

Всякое решение системы (9) называется *допустимым решением*. Допустимое решение, при котором линейная функция $f(x)$ принимает оптимальное (в данном случае максимальное) значение, называется *оптимальным решением*, или *оптимальным планом*.

2.3. Формулировка некоторых типичных задач линейного программирования

2.3.1. Задача о раскрое

Пусть имеется n вариантов раскроя листа жести на заготовки m типов. Заготовок i-го типа требуется в количестве b_i $(i = 1, .., m)$. При j-м способе раскроя $j = 1, \ldots, n$ получаются заготовки i-го вида в количестве a_{ij}, а отходы (по весу, по площади и др.) равны c_j. Обозначим как x_j количество листов, раскраиваемых по варианту j. Вектор $X = (x_1, x_2, ..., x_n)$ представляет собой общий план раскроя. Тогда задача по отысканию оптимального плана раскроя может быть записана следующим образом.

Определить вектор X, минимизирующий функцию отходов $f(X) = \sum_{j=1}^{n} c_j \cdot x_j$ при ограничениях, направленных на выполнение плана выпуска заготовок требуемых типов:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + ... + a_{1n} \cdot x_n = b_1; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{i1} \cdot x_1 + a_{i2} \cdot x_2 + ... + a_{in} \cdot x_n = b_i; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + ... + a_{mn} \cdot x_n = b_{m;} \\ x_j \geq 0, \quad j = 1, ..., n. \end{cases}$$

При постановке конкретной задачи обычно все исходные данные сводятся в таблицу.

2.3.2. Задача о планировании производства

Для n различных технологий предприятие может использовать m видов ресурсов – сырье, энергию, оборудование. Запас ресурсов ограничен величинами $b_i\ (i=1,...,m)$. Известно, что при применении технологии с номером $j\ (j=1,\dots,n)$ в единицу времени расходуется a_{ij} единиц ресурса, и производится c_j единиц продукта.

Требуется определить интенсивность (продолжительность) использования каждой технологии, чтобы выпустить максимальное количество продукции, не допустив перерасхода ресурсов.

Пусть x_j – время, затраченное на j-тую технологию.

Тогда справедлива следующая математическая формулировка рассматриваемой задачи:

$$\sum_{j=1}^{n} C_j \cdot x_j \to \max$$

при ограничениях $$\begin{cases} \sum\limits_{j=1}^{n} x_j \cdot a_{ij} \le b_i, \ i=1,...,m; \\ x_j \ge 0, \ j=1,...,n. \end{cases}$$

2.3.3. Задача о рационе

Пусть имеется n видов продуктов, в которых содержатся m типов питательных веществ. Обозначим символами a_{ij} количество питательных веществ i-го типа ($i=1,\dots,m$) в j-м продукте ($j=1,\dots,n$). Потребность организма в i-м питательном веществе на протяжении определенного времени задается величиной b_i. Стоимость единицы j-го продукта равна c_j.

Требуется определить количество каждого продукта в рационе так, чтобы суммарная стоимость рациона была минимальной.

Пусть x_j – количество j-го продукта в рационе. Тогда $c_j x_j$ – общая стоимость j-го продукта, $\sum\limits_{j=1}^{n} c_j x_j$ – суммарная стоимость рациона.

Получаем следующую ЗЛП:

$$\sum_{j=1}^{n} c_j x_j \to \min,$$

$$\begin{cases} \sum\limits_{j=1}^{n} a_{ij} x_j \ge b_i, i=1,...,m, \\ x_j \ge 0, \ j=1,...,n. \end{cases}$$

2.3.4. Транспортная задача

Пусть имеется n строительных объектов, в которые поступают строительные материалы, хранимые на m складах. Известна стоимость c_{ij} перевозки материалов с i-го склада ($i = 1, \ldots, m$) на j-й строительный объект ($j = 1, \ldots, n$), количество a_i единиц строительных материалов на i-м складе и заказанный объем b_j материалов для доставки на j-й строительный объект. Требуется составить план перевозок строительных материалов со складов на объекты так, чтобы суммарная стоимость перевозок была минимальной.

Пусть x_{ij} – количество материалов, которое планируется перевезти с i-го склада на j-й объект. Тогда стоимость перевозки $c_{ij}x_{ij}$, а общая стоимость S всех перевозок будет равна

$$S = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} .$$

Строительные объекты должны быть обеспечены материалами в точном соответствии с заказом. Поэтому планируемые объемы перевозок x_{ij} должны удовлетворять условиям

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1, 2, \ldots, n.$$

Однако со склада нельзя вывезти строительных материалов больше, чем там имеется. Следовательно, должны быть выполнены условия

$$\sum_{i=1}^{m} x_{ij} \le a_i, \quad i = 1, 2, \ldots, m.$$

Получаем следующую ЗЛП:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min$$

$$\begin{cases} \sum\limits_{i=1}^{m} x_{ij} = b_j, \quad j = 1, 2, \ldots, n \\ \sum\limits_{j=1}^{n} x_{ij} \le a_i, \; i = 1, \ldots, m, \\ x_{ij} \ge 0, \; i = 1, \ldots, m, \quad j = 1, \ldots, n. \end{cases}$$

Отметим, что сформулированная ЗЛП может иметь решение только в том случае, если сумма заказов всех строительных объектов не превышает суммарного запаса строительных материалов на всех складах, т.е.

$$\sum_{j=1}^{n} b_j \le \sum_{i=1}^{m} a_i .$$

2.4. Свойства решений задач линейного программирования

1) Множество допустимых значений задач линейного программирования является либо пустым, либо выпуклым множеством.

2) Если множество допустимых значений не пусто и целевая функция ограничена сверху на этом множестве, то задача максимизации имеет решение.

3) Оптимальное решение ЗЛП всегда находится на границе допустимой области.

Примечание.

Множество в пространстве R^n называется *выпуклым*, если вместе с двумя точками $x_1 = \left(x_1^1, x_2^1, \ldots, x_n^1\right)$, $x_2 = \left(x_1^2, x_2^2, \ldots, x_n^2\right)$ множество содержит и отрезок, их соединяющий (рис. 2).

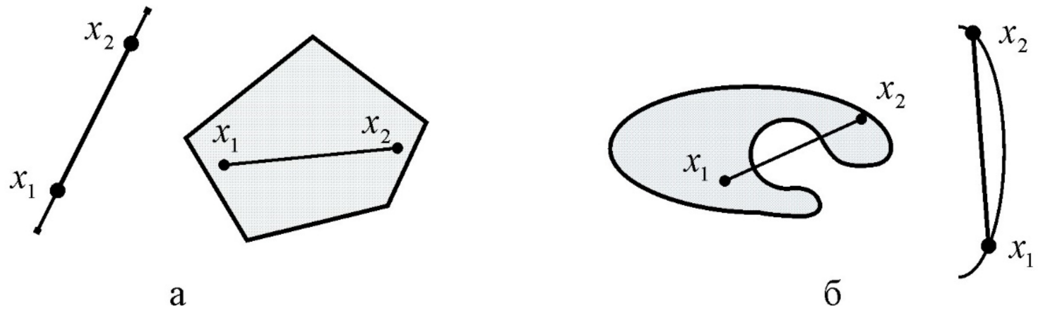


Рис. 2. Примеры выпуклых (а) и невыпуклых (б) множеств

2.4.1. Опорные решения задач линейного программирования

Предположим, что ЗЛП приведена к канонической форме

$$\sum_{i=1}^{n} c_i \cdot x_i \to \max_{x_i \in D},$$

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \ldots + a_{1n} \cdot x_n = b_1, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \ldots + a_{mn} \cdot x_n = b_m, \\ x_j \ge 0, \ \ j = 1, \ldots, n. \end{cases}$$

В векторном виде:

$$C = (c_1, c_2, \ldots, c_n), \ \ A = \left(A_1, A_2, \ldots, A_n\right), \ \ \ A_i = \left(a_{1i}, a_{2i}, \ldots, a_{mi}\right)^T,$$

$$B = (b_1, b_2, \ldots, b_m)^T, \ \ X = (x_1, x_2, \ldots, x_n),$$

$$\begin{cases} (C, X) \to \max, \\ A_1 \cdot x_1 + A_2 \cdot x_2 + \ldots + A_n \cdot x_n = B, \\ x_j \ge 0, j = 1, \ldots, n. \end{cases}$$

Допустимое решение задачи в канонической форме – вектор $X=(0,\ldots,x_{j_1},0,\ldots,0,x_{j_2},0,...,x_{j_k})$, где $x_{j_m}\neq 0$ – называется *опорным решением* (планом) ЗЛП, если векторы условий $A_{j_1},A_{j_2},A_{j_3},...,A_{j_k}$, соответствующие ненулевым координатам вектора X, образуют линейно независимую систему векторов.

Свойства опорных решений

1) Если допустимое множество ЗЛП в канонической форме не является пустым множеством, то ЗЛП имеет опорное решение.

2) Опорные решения являются крайними точками допустимого множества ЗЛП (вершины многогранника).

3) ЗЛП в канонической форме имеет конечное число различных опорных решений (либо не имеет их вообще).

4) Если ЗЛП в канонической форме имеет оптимальное решение, то это оптимальное решение является опорным.

Из свойств опорных решений следует, что искать оптимальное решение задачи линейного программирования необходимо среди ее опорных решений, что всегда можно сделать, так как опорных решений конечное число.

Поиск опорного решения

Для того чтобы найти некоторое опорное решение ЗЛП, достаточно выбрать базис системы векторов $A_1,A_2,\ldots,A_n$ так, чтобы вектор ограничений B раскладывался по нему с неотрицательными коэффициентами, то есть найти базис $A_{i_1},A_{i_2},...,A_{i_r}$ такой, что $B=d_1A_{i_1}+d_2A_{i_2}+...+d_rA_{i_r}$; причем все $d_1,d_2,\ldots,d_r$ – неотрицательные числа $(d_j\geq 0)$.

После определения такого базиса опорное решение $\bar{x}$ записывается в виде

$$\bar{x}=\left(0,...,d_{i_1},0,...,0,d_{i_2},0,...,0,d_{i_r}\right).$$

Напомним, что *базис* векторов в *n*-мерном векторном пространстве – это множество линейно независимых векторов, такое, что любой вектор из рассматриваемого пространства представим в виде линейной комбинации векторов базиса.

2.5. Графический метод решения ЗЛП

Рассмотрим ЗЛП с двумя переменными $x_1, x_2 \in D$:

$$\max(c_1 \cdot x_1 + c_2 \cdot x_2), \tag{11}$$

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 \le b_1, \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 \le b_2, \\ \vdots \\ a_{n1} \cdot x_1 + a_{n2} \cdot x_2 \le b_m. \end{cases} \qquad \begin{array}{l} x_j \ge 0, \\ j \in K. \end{array}$$

Ограничения, заданные в виде равенств и неравенств, определяют в плоскости x_1Ox_2 следующие геометрические формы:

1) равенство типа $a_{i_1}x_1 + a_{i_2}x_2 = b_i$ – некоторую прямую;

2) неравенство $a_{i_1}x_1 + a_{i_2}x_2 \le b_i$ – полуплоскость;

3) неравенство $x_i \ge 0,\ i = 1, 2$ – полуплоскость, а совокупность неравенств $\begin{cases} x_1 \ge 0, \\ x_2 \ge 0 \end{cases}$ – первый квадрант на координатной плоскости.

Таким образом, ЗЛП (11) представляет собой задачу нахождения максимума линейной формы, когда переменные x_1, x_2 принадлежат пересечению m полуплоскостей или отрезков прямых.

Следовательно, в этом случае допустимая область является выпуклым многоугольником, вершины которого находятся в точках пересечения прямых $a_{i_1}x_1 + a_{i_2}x_2 = b_i$. Если вершина получается в результате пересечения трех и более прямых, то она называется *вырожденной*.

Для решения задачи (11) геометрическим методом (рис. 3) в случае 2-х переменных необходимо:

1) задать такую константу, чтобы линия уровня $c_1x_1 + c_2x_2 = \text{const}$ пересекала допустимую область;

2) построить вектор-градиент $\text{grad} f = (c_1, c_2)$;

3) перемещать линию уровня – перпендикулярную градиенту – вдоль градиента до пересечения этой линии с крайней точкой допустимого многоугольника; эта точка и будет решением задачи;

4) точные координаты крайней точки вычислять как точки пересечения соответствующих прямых.

Примечание. Крайних точек может быть несколько или даже бесконечное множество (отрезок прямой). В этом случае любая из них является решением задачи.

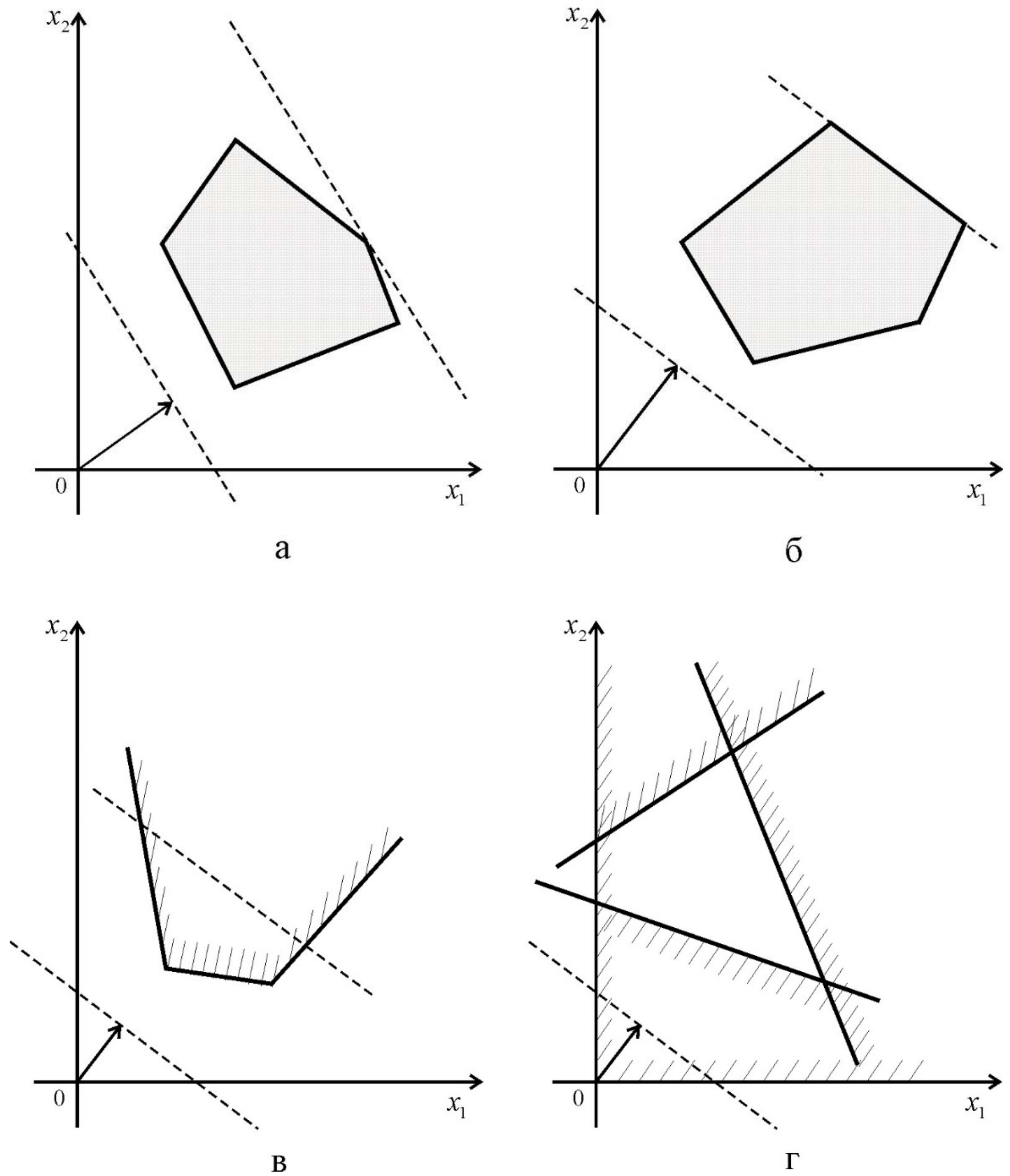


Рис. 3. Нахождение точки максимума целевой функции:
а – решение единственно; б – бесконечное множество решений;
в – целевая функция не ограничена сверху на множестве допустимых значений; г – система ограничений несовместна

2.6. Симплексный метод решения ЗЛП

Симплексный метод является одним из универсальных методов решения ЗЛП. *Симплексом* в n-мерном пространстве называют множество $(n+1)$-й точки. Геометрически симплекс представляет собой многогранник с $(n+1)$-й вершиной: на плоскости – это треугольник, в трехмерном пространстве – тетраэдр. Симплекс в R^n образуют любые точки $x^0, x^1, \ldots, x^n$, для которых векторы $x^1 - x^0, x^2 - x^0, \ldots, x^n - x^0$ линейно независимы. При этом точки $x^0, x^1, \ldots, x^n$ называют *вершинами симплекса*.

Предположим, что ЗЛП приведена к канонической форме

$$f(x) = \sum_{i=1}^{n} c_i \cdot x_i \to \max_{x_i \in D},$$

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \ldots + a_{1n} \cdot x_n = b_1, \\ \vdots \\ a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \ldots + a_{mn} \cdot x_n = b_m, \\ x_j \geq 0, \;\; j = 1, \ldots, n. \end{cases} \tag{12}$$

Идея симплекс-метода заключается в следующем итерационном алгоритме. Вычислив в $(n + 1)$ вершинах симплекса значения целевой функции, выберем наихудшую из них – ту, в которой значение функции наименьшее. Заменив эту вершину новой по определенному правилу, получим новый симплекс, с которым повторяем такую же процедуру. В результате получим последовательность симплексов, стягивающихся к точке максимума целевой функции.

Для этого сначала находим какое-либо допустимое (неотрицательное) *базисное решение*. Считаем, что уравнения системы ограничений (12) линейно независимы, т.е. ранг этой системы $r = m < n$. Произвольные переменные $x_1, x_2, \ldots, x_m$ принимаем за *базисные* (основные), остальные $(n - m)$ переменных – за *свободные* (небазисные). Приравняв к нулю все свободные переменные, решим полученную систему уравнений. Если некоторые значения найденных базисных переменных окажутся отрицательными, то нужно выбрать другие базисные переменные, то есть перейти к новому базису. После того, как найдено допустимое базисное решение, проверяем, не достигнут ли максимум целевой функции $f(x)$. Если нет, то ищем новое допустимое базисное решение, при котором значение функции $f(x)$ возрастает. Затем процедура повторяется.

Алгоритм симплекс-метода

Алгоритм симплекс-метода состоит в последовательном переборе базисных решений, определяющих крайние точки многогранника решений. Переход от одного базисного решения к другому (от одной крайней точки к другой) осуществляется таким образом, чтобы значение $f(x)$ возрастало от итерации к итерации.

1. Находим ранг r системы (12). Предположим, что $r < n$. Запишем систему ограничений и целевую функцию в виде:

$$x_{k+i} = \beta_i - \sum_{j=1}^{k} \alpha_{ij} \cdot x_j; \quad i = 1, \ldots, r; \quad k = n - r;$$

$$f = \gamma_0 - \sum_{j=1}^{k} \gamma_j \cdot x_j,$$

где α_{ij}, β_i, γ_0, γ_j – некоторые константы, полученные в результате преобразований функций ограничений и целевой функции.

2. Составляем исходную симплекс-таблицу (табл. 1), состоящую из $(m+1)$ строк (m – количество уравнений ограничений плюс одна строка для целевой функции) и $(n+1)$ столбцов (n – количество свободных переменных плюс один столбец для правых частей ограничений).

Таблица 1

Базис		x_1	…	x_s	…	x_j	…	x_m
x_{m+1}	β_1	α_{11}	…	α_{1s}	…	α_{1j}	…	α_{1m}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_{m+l}	β_l	α_{l1}	…	α_{ls}	…	α_{lj}	…	α_{lm}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_{m+i}	β_i	α_{i1}	…	α_{is}	…	α_{ij}	…	α_{im}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_n	β_r	α_{r1}	…	α_{rs}	…	α_{rj}	…	α_{rm}
f	γ_0	γ_1	…	γ_s	…	γ_j	…	γ_m

3. Выбираем генеральный (ведущий) элемент α_{ij} в соответствии с правилами:

– выбираем положительный коэффициент целевой функции (любой положительный элемент в последней строке симплекс-таблицы, кроме свободного члена γ_0), например элемент j-го столбца γ_j. Если положительных элементов нет, то записанное в симплекс-таблице базисное решение будет *оптимальным*;

– из столбца j выбираем строки l, для которых $\alpha_{lj} > 0$, и находим отношения $\frac{\beta_l}{\alpha_{lj}}$;

– выбираем из этих отношений наименьшее, например $\frac{\beta_l}{\alpha_{lj}}$ (если наименьшее отношение достигается при нескольких значениях l, то можно взять любое).

Элемент α_{lj} – генеральный.

4. Переходим от табл. 1 к табл. 2.

Преобразуем столбец j путем деления всех коэффициентов в этом столбце, кроме генерального элемента, на значение генерального элемента со знаком « – ».

Преобразуем строку i путем деления всех коэффициентов в этой строке, кроме генерального элемента, на значение генерального элемента со знаком « + ».

Обозначим $\lambda = \dfrac{1}{\alpha_{ij}}$. Запишем значение λ в ячейку генерального элемента.

Таблица 2

		x_1	…	x_s	…	x_j	…	x_k
x_{m+1}	$\beta_1 - \lambda\beta_i\alpha_{1j}$	$\alpha_{11} - \lambda\alpha_{i1}\alpha_{1j}$	…	$\alpha_{1s} - \lambda\alpha_{is}\alpha_{1j}$	…	$-\lambda\alpha_{1j}$	…	$\alpha_{1m} - \lambda\alpha_{im}\alpha_{1j}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_{m+l}	$\beta_l - \lambda\beta_i\alpha_{lj}$	$\alpha_{l1} - \lambda\alpha_{il}\alpha_{lj}$	…	$\alpha_{ls} - \lambda\alpha_{is}\alpha_{lj}$	…	$-\alpha_{lj}$	…	$\alpha_{lm} - \lambda\alpha_{im}\alpha_{lj}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_n	$\lambda\beta_i$	$\lambda\alpha_{i1}$	…	$\lambda\alpha_{is}$	…	λ	…	$\lambda\alpha_{im}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_r	$\beta_r - \lambda\beta_i\alpha_{rj}$	$\alpha_{r1} - \lambda\alpha_{i1}\alpha_{rj}$	…	$\alpha_{rs} - \lambda\alpha_{is}\alpha_{rj}$	…	$-\lambda\alpha_{rj}$	…	$\alpha_{rm} - \lambda\alpha_{im}\alpha_{rj}$
f	$\gamma_0 - \lambda\beta_i\gamma_j$	$\gamma_1 - \lambda\alpha_{i1}\gamma_j$	…	$\gamma_s - \lambda\alpha_{is}\gamma_j$	…	$-\lambda\gamma_j$	…	$\gamma_m - \lambda\alpha_{im}\gamma_j$

Преобразуем коэффициенты в оставшихся ячейках таблицы по правилу: из значения коэффициента соответствующей ячейки α_{rs} (с номерами строки r и столбца s) вычитаем произведение коэффициента, находящегося в этой же строке r и в столбце j генерального элемента (α_{rj}), и коэффициента, находящегося в этой же строке i генерального элемента и в столбце s (α_{ir}).

5. Повторяем пп. 3–4 до получения оптимального решения (признак оптимальности решения – в п.3).

Замечание. Если в последней строке симплекс-таблицы имеется положительный элемент, но в соответствующем столбце нет положительных чисел, это значит, что целевая функция не ограничена сверху (не имеет максимума).

Методические рекомендации к теме 2

Знать общую постановку задачи линейного программирования, каноническую форму представления ЗЛП, иметь представление о типичных задачах линейного программирования. Самостоятельно рассмотреть транспортную задачу по критерию времени [14].

Знать свойства решений задач линейного программирования, уметь находить опорные решения ЗЛП, иметь представление об алгоритмах графического и аналитического способов решения ЗЛП, уметь решать ЗЛП графическим и симплекс-методом.

Уметь отвечать на контрольные вопросы по теме и решать поставленные задачи.

Контрольные вопросы и задания

1. Какой вид может иметь область допустимых значений в задаче линейного программирования? Чем она ограничена?
2. Сформулируйте задачу линейного программирования в общем виде.
3. Какой вид могут иметь ограничения в ЗЛП?
4. Можно ли в ЗЛП ограничение типа неравенства привести к ограничению типа равенства? Каким образом?
5. Как привести известные вам типичные ЗЛП к каноническому виду?
6. Приведите примеры типичных задач линейного программирования.
7. Можно ли задачу минимизации преобразовать в задачу максимизации? Если это возможно, покажите на примере.
8. Сколько решений может иметь ЗЛП? Отчего это зависит? Можно ли определить эти решения графически?
9. Что представляет собой линия уровня?
10. Когда задача оптимизации имеет решение?
11. Дайте определение выпуклого множества и приведите примеры выпуклых и невыпуклых множеств.
12. В каком месте допустимой области ЗЛП находится ее оптимальное решение?
13. Какие решения ЗЛП называются опорными?
14. Как можно определить опорные решения ЗЛП аналитически и графически?
15. К какому виду необходимо привести ЗЛП для решения ее симплексным методом?
16. Как выбирается генеральный (ведущий) элемент в симплекс-методе?
17. Может ли ведущий элемент симплекс-таблицы быть: а) нулевым; б) отрицательным?

3. ЭЛЕМЕНТЫ ВЫПУКЛОГО ПРОГРАММИРОВАНИЯ

3.1. Основные определения

Как нам уже известно, множество $X \in R^n$ называется выпуклым, если для любой пары точек $x_1, x_2 \in X$ выполняется включение $\lambda x_1 + (1-\lambda)\cdot x_2 \in X$, где $0 \le \lambda \le 1$.

Точка x из множества X называется *внутренней* точкой, если существует ε – окрестность этой точки, которая целиком содержится во множестве X.

Точка x из множества X называется *граничной* точкой, если для нее существует окрестность, часть которой лежит в множестве X, а часть – за его пределами, но не существует окрестности, которая целиком содержится в множестве X.

Функция $f(x)$, определенная на некотором выпуклом множестве X, называется *выпуклой*, если для любых точек $x_1, x_2 \in X$ и для любого значения $\lambda \in [0, 1]$ выполняется неравенство

$$f(\lambda\cdot x_1 + (1-\lambda)\cdot x_2) \le \lambda\cdot f(x_1) + (1-\lambda)\cdot f(x_2).$$

Если неравенство строгое, то функция называется *строго выпуклой*:

$$f\left(\lambda\cdot x_1 + \left(1-\lambda\right)\cdot x_2\right) < \lambda\cdot f\left(x_1\right) + \left(1-\lambda\right)\cdot f\left(x_2\right).$$

Если неравенство нестрогое, но развернуто в другую сторону, то функция называется *вогнутой*:

$$f\left(\lambda\cdot x_1 + \left(1-\lambda\right)\cdot x_2\right) \ge \lambda\cdot f\left(x_1\right) + \left(1-\lambda\right)\cdot f\left(x_2\right),$$

для строгого неравенства – *строго вогнутой*:

$$f\left(\lambda\cdot x_1 + \left(1-\lambda\right)\cdot x_2\right) > \lambda\cdot f\left(x_1\right) + \left(1-\lambda\right)\cdot f\left(x_2\right).$$

Можно доказать, что функция является выпуклой тогда и только тогда, когда для любой пары точек из области определения выполняется неравенство

$$f\left(\frac{1}{2}\cdot x_1 + \frac{1}{2}\cdot x_2\right) \le \frac{1}{2}\cdot f\left(x_1\right) + \frac{1}{2}\cdot f\left(x_2\right).$$

Квадратичной формой относительно переменных $x_1, x_2, \ldots, x_n$ называется числовая функция от этих переменных, имеющая вид

$$f = c_{11}x_1x_1 + c_{12}x_1x_2 + \ldots = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_ix_j.$$

Квадратичная форма называется *положительно* (*отрицательно*)-*определенной*, если $f(x) > 0$ ($f(x) < 0$) для всех значений переменных $x = (x_1, x_2, \ldots, x_n)$, кроме $x = 0$.

Квадратичная форма называется *положительно* (*отрицательно*)-*полуопределенной*, если $f(x) \geq 0$ ($f(x) \leq 0$) для любого набора значений переменных $x = (x_1, x_2, \dots, x_n)$ и, кроме того, существует такой набор переменных $x' = (x'_1, x'_2, \dots, x'_n)$, где не все значения переменных одновременно равны нулю, что $f(x') = 0$.

Квадратичная форма является выпуклой функцией, если она положительно-полуопределенная, и вогнутой функцией, если она отрицательно-полуопределенная.

3.2. Теоремы о выпуклых функциях

Будем рассматривать функции $f(x)$, выпуклые на выпуклом множестве $X \in R^n$.

Теорема 1.

Выпуклая функция, заданная на некотором выпуклом множестве, непрерывна в любой его внутренней точке.

Условие для непрерывности функции: $\lim\limits_{x \to x_0} f(x) = f(x_0)$.

(Без доказательства).

Теорема 2.

Выпуклая функция $f(x)$, определенная на некотором выпуклом множестве, имеет в любой внутренней точке этого множества производную по любому направлению.

$$\frac{\partial f}{\partial s} = \lim_{\lambda \to 0} \frac{f(x + \lambda s) - f(x)}{\lambda}$$ – производная по направлению s.

(Без доказательства).

Теорема 3.

Функция $f(x)$, дифференцируемая в точке x на выпуклом замкнутом множестве X, выпукла в том и только в том случае, если для любой пары точек $x, y \in X$ выполняется неравенство

$$(f'(x), y - x) \leq f(y) - f(x).$$

Докажем *необходимость* выполнения условия теоремы для выпуклости функции $f(x)$.

Запишем условие выпуклости в виде

$$f(x + \lambda \cdot (y - x)) \leq f(x) + \lambda \cdot [f(y) - f(x)].$$

Действительно,

$$f(x+\lambda\cdot(y-x))=f(\lambda\cdot y+(1-\lambda)x)\le\lambda\cdot f(y)+(1-\lambda)\cdot f(x)=$$
$$=\lambda\cdot f(y)+f(x)-\lambda\cdot f(x)=f(x)+\lambda\cdot(f(y)-f(x)).$$

Таким образом,

$$f(x+\lambda\cdot(y-x))-f(x)\le\lambda(f(y)-f(x)).$$

То есть

$$\frac{f(x+\lambda\cdot(y-x))-f(x)}{\lambda}\le f(y)-f(x).$$

Рассмотрим функцию $\psi(\lambda)=f(x+\lambda\cdot(y-x))$.

Тогда

$$\frac{\Psi(\lambda)-\Psi(0)}{\lambda}\le f(y)-f(x)$$

и
$$\lim_{\lambda\to 0}\frac{\Psi(\lambda)-\Psi(0)}{\lambda}=\psi'_\lambda\big|_{\lambda=0}\le f(y)-f(x).$$

Рассмотрим

$$\Psi'(\lambda)\,|_{\lambda=0}=\sum_{i=1}^{n}\frac{\partial f}{\partial x_i}\cdot\frac{\partial x_i}{\partial\lambda}=\sum_{i=1}^{n}\frac{\partial f}{\partial x_i}(y_i-x_i)=(f'(x),y-x).$$

Следовательно, получим:

$$(f'(x),y-x)\le f(y)-f(x). \tag{13}$$

Теорема 4.

Локальный минимум $x^*\in X$ выпуклой функции $f(x)$ на выпуклом множестве X совпадает с ее глобальным минимумом.

Доказательство.

Допустим, что в области определения функции $f(x)$ существует точка $x'\in X$ такая, что $f(x')<f(x^*)$, где x^* – точка локального минимума.

Выберем точку $x=\alpha\cdot x'+(1-\alpha)\cdot x^*$, $\alpha\in[0,1]$ и рассмотрим функцию $f(x)$ в этой точке:

$$f(x)=f(\alpha\cdot x'+(1-\alpha)x^*)\le\alpha\cdot f(x')+(1-\alpha)f(x^*)\le$$
$$\le\alpha\cdot f(x^*)+(1-\alpha)f(x^*)=f(x^*),$$

т.е. получим для любой точки x, лежащей на отрезке $\{x',x^*\}$, неравенство $f(x)\le f(x^*)$. Это означает, что значение функции $f(x)$ в точках, как угодно близких к точке x^* ($\alpha\to 0$), может быть меньше, чем значение функции $f(x^*)$; следовательно, точка x^* не является точкой локального минимума, что противоречит условию теоремы.

Теорема 5.

Система неравенств

$$g_i(x) \geq 0, \quad i = 1, ..., m,$$

где g_i – вогнутые функции, определяет выпуклое множество

$$X = \{x : g_i(x) \geq 0, \quad i = 1, ..., m\}.$$

Доказательство.

Пусть $x, y \in X$, тогда $g_i(x) \geq 0$ и $g_i(y) \geq 0$. Так как функции g_i – вогнутые, то для любой точки $\alpha \in [0,1]$ имеем:

$$g_i(\alpha x + (1 - \alpha)y) \geq \alpha g_i(x) + (1 - \alpha)g_i(y) \geq 0.$$

Следовательно, точка $\alpha x + (1 - \alpha)y \in X$.

Теорема 6.

Функция $f(x)$, определенная и дважды дифференцируемая на выпуклом множестве X, выпукла тогда и только тогда, когда матрица Гёссе этой функции является положительно-полуопределенной в любой точке множества X.

(Без доказательства).

Напомним, что матрица Гёссе $H(x)$ – симметричная матрица вторых производных функции $\left[\dfrac{\partial^2 f}{\partial x_i \partial x_j}\right]$, а необходимыми и достаточными условиями положительной определенности симметричной матрицы являются: либо положительность определителя и всех главных миноров матрицы, либо положительность всех ее собственных чисел.

Теорема 7.

Сумма любого конечного числа выпуклых функций является выпуклой функцией.

(Доказательство методом математической индукции).

Теорема 8.

Если функция $f(x)$ выпуклая, тогда для $x = \sum_{i=1}^{N} \beta_i s_i, \quad \sum_{i=1}^{N} \beta_i = 1$ выполняется неравенство

$$f(x) = f\left(\sum_{i=1}^{N} \beta_i s_i\right) \leq \sum_{i=1}^{N} \beta_i f(s_i).$$

(Без доказательства).

3.3. Задача выпуклого программирования (ЗВП)

Пусть X – выпуклое множество. Задача выпуклого программирования заключается в следующем: найти $\min\limits_{x \in X} f(x)$ при ограничениях

$$\begin{cases} g_j(x) \le 0, & j = 1, ..., m, \quad x = (x_1, ..., x_n), \\ x_i \ge 0, & i = 1, ..., n, \end{cases}$$

где $f(x), g_j(x)$ – выпуклые функции на множестве X.

3.3.1. Условие регулярности Слейтера

Область Ω, являющаяся допустимой областью ЗВП, называется *регулярной*, если существует точка $x^{(0)} \in \Omega,\ x^{(0)} = (x_1^{(0)}, ..., x_n^{(0)})$ такая, что $x_i^{(0)} \ge 0,\ i = 1, ..., n$ и $g_j\left(x^{(0)}\right) \le 0,\ j = 1, ..., m$.

Говорят, что для регулярной области Ω выполняется *условие Слейтера*.

3.3.2. Функция Лагранжа

Пусть задача выпуклого программирования сформулирована в канонической форме:

$$\begin{aligned} & f(x) \to \min_{x \in \Omega}; \quad x = (x_1, x_2, ..., x_n); \\ & g_j(x) \le 0,\ j = 1, 2, ..., m; \\ & x_i \ge 0,\ i = 1, 2, ..., n. \end{aligned} \tag{14}$$

Рассмотрим вектор $u = (u_1, u_2, \ldots, u_m)$, у которого $u_j \in R$ и $u_j \ge 0,\ \ j = 1, 2, \ldots, m$. Построим функцию Лагранжа $L(x, u)$ для задачи выпуклого программирования:

$$L(x,u) = f(x) + \sum_{j=1}^{m} u_j g_j(x) = f(x) + (G,u). \tag{15}$$

Здесь $G = \{g_1(x), g_2(x), ..., g_m(x)\}$; (G,u)– скалярное произведение, а компоненты вектора $u = (u_1, u_2, \ldots, u_m)$ – множители Лагранжа.

3.3.3. Седловая точка

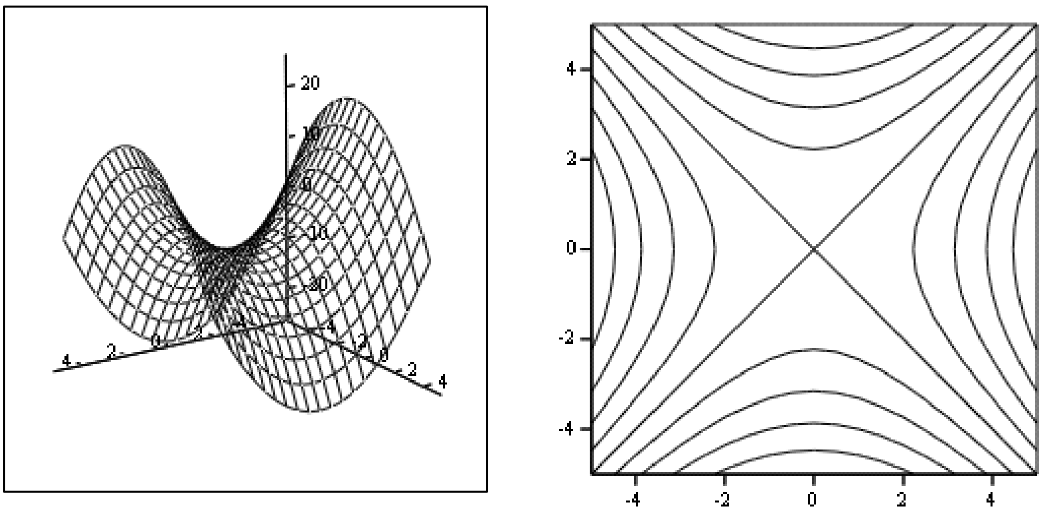
Пара векторов x^*, u^* называется *седловой точкой* функции Лагранжа $L(x,u)$, если вектор x^* принадлежит области Ω, $u_j^* \ge 0,\ \ j = 1, 2, ..., m$, и выполняется неравенство

$$L(x^*, u) \le L(x^*, u^*) \le L(x, u^*). \tag{16}$$

Это неравенство означает, что $L(x, u^*)$ имеет локальный минимум, а, следовательно, и глобальный минимум в точке x^* (так как L – выпуклая функция по переменной x) при $u = u^*$. Кроме того, функция $L(x^*, u)$ имеет и глобальный максимум в точке u^* при $x = x^*$. Следовательно, можно записать, что $L(x^*, u^*) = \min\limits_{x \in \Omega} \max\limits_{u_j \ge 0} L(x, u)$.

На рис. 4 показаны пример функции с седловой точкой и линии уровня.

$L(x, u) := x^2 - u^2$



L

L

Рис. 4. Седловая точка $L(0, 0)$ и линии уровня

3.3.4. Теорема Куна – Таккера

Пусть область Ω является областью, удовлетворяющей условию регулярности Слейтера для задачи выпуклого программирования (14).

Вектор $x^* = (x_1^*, x_2^*, ..., x_n^*)$ будет решением задачи выпуклого программирования (14) тогда и только тогда, когда существует вектор $u^* = (u_1^*, u_2^*, ..., u_m^*)$ такой, что $x_i^* \ge 0$, $u_j^* \ge 0$, где $i = 1, 2, ..., n$ и $j = 1, 2, ..., m$, и пара векторов x^*, u^* является седловой точкой функции Лагранжа для задачи выпуклого программирования (14), то есть выполняется неравенство $L(x^*, u) \le L(x^*, u^*) \le L(x, u^*)$.

Докажем достаточность условия теоремы.

Подставив выражения для функции $L(x, y)$ из равенства (15) в неравенство (16), получим:

$$f(x^*) + \sum_{j=1}^{m} u_j g_j(x^*) \le f(x^*) + \sum_{j=1}^{m} u_j^* g_j(x^*) \le f(x) + \sum_{j=1}^{m} u_j^* g_j(x). \qquad (17)$$

Так как u_j – произвольная неотрицательная величина, то можно заключить, что:

1) $g_j(x^*) \le 0$ (так как $\sum_{j=1}^{m} u_j g_j(x^*) \le \sum_{j=1}^{m} u_j^* g_j(x^*)$ и $u_j^* \ge 0$);

2) $\sum_{j=1}^{m} u_j^* g_j(x^*) = 0$, так как при $u_j = 0, \quad j = 1, 2, \ldots, m$ из пункта 1 следует, что $\sum_{j=1}^{m} u_j^* g_j(x^*) \ge 0,$ но $g_j\left(x^*\right) \le 0$, а $u_j^* > 0.$

Это возможно, только когда $\sum_{j=1}^{m} u_j^* g_j(x^*) = 0$.

3) Тогда из правой части неравенства (17) получим:

$$f(x^*) + \sum_{j=1}^{m} u_j^* g_j(x^*) \le f(x) + \sum_{j=1}^{m} u_j^* g_j(x),$$

$$f(x^*) \le f(x) + \sum_{j=1}^{m} u_j^* g_j(x).$$

Так как $g_j\left(x\right) \le 0$, то $f(x^*) \le f(x)$; следовательно, x^* – точка минимума.

3.3.5. Теорема Куна – Таккера для дифференцируемых функций

Если функции $f(x)$ и $g_j(x)$ из задачи выпуклого программирования (14) непрерывно дифференцируемы в области Ω, то, для того чтобы пара векторов x^*, u^* представляла собой седловую точку функции Лагранжа для данной ЗВП, необходимо и достаточно выполнение следующих условий:

$$1) \quad \frac{\partial L^*}{\partial x} \ge 0;$$

$$2) \quad \left(x^*, \frac{\partial L^*}{\partial x} \right) = 0;$$

$$3) \quad x^* \ge 0; \quad j = 1, 2, \ldots, n;$$

$$4) \quad \frac{\partial L^*}{\partial u} \le 0;$$

$$5) \quad \left(u^*, \frac{\partial L^*}{\partial u} \right) = 0;$$

$$6) \quad u_i^* \ge 0, \quad i = 1, 2, \ldots, m.$$

Здесь $L^* = L(x^*, u^*)$, $\dfrac{\partial L^*}{\partial x} = \left(\dfrac{\partial L^*}{\partial x_1}, \dfrac{\partial L^*}{\partial x_2}, \ldots, \dfrac{\partial L^*}{\partial x_n}\right)$, $\dfrac{\partial L^*}{\partial u} = \left(\dfrac{\partial L^*}{\partial u_1}, \dfrac{\partial L^*}{\partial u_2}, \ldots, \dfrac{\partial L^*}{\partial u_m}\right)$.

Данная теорема дает возможность решать задачу выпуклого программирования путем сведения ее к решению системы уравнений и неравенств.

Методические рекомендации к теме 3

Знать основные понятия и теоремы выпуклого программирования, формулировку ЗВП, иметь представление об условии регулярности Слейтера, уметь доказывать выпуклость функций, составлять функцию Лагранжа для ЗВП, составлять условия теоремы Куна – Таккера и решать ЗВП.

Уметь отвечать на контрольные вопросы по теме и решать поставленные задачи.

Контрольные вопросы и задания

1. Какая область называется регулярной?
2. Какую роль в задачах минимизации играет функция Лагранжа?
3. Что такое седловая точка функции Лагранжа?

4. МЕТОДЫ БЕЗУСЛОВНОЙ МИНИМИЗАЦИИ (минимизация без ограничений)

Общая задача нелинейного программирования без ограничений сводится к следующей: минимизировать целевую функцию $f(x),\ x \in R^n,\ f(x) \in R$.

Задача минимизации функции решается с помощью построения итерационного алгоритма; причем если в алгоритме не используются производные функции $f(x)$, то метод называется методом *нулевого порядка*. Если в алгоритме используются первые производные функции $f(x)$, то метод называется методом *первого* порядка; если вторые производные – методом *второго* порядка и т.д.

4.1. Методы нулевого порядка

Методы нулевого порядка применяются в тех случаях, когда:

1) вычисление производных функции $f(x)$ затруднительно;

2) при вычислении производных значительно теряется точность алгоритма;

3) функция $f(x)$ не имеет производных;

4) функция $f(x)$ является функцией «овражного типа», то есть когда она очень медленно убывает в одном направлении.

Методы нулевого порядка неприхотливы к свойствам целевой функции: не требуют ее непрерывности, дифференцируемости, монотонности и т.д. Рассмотрим некоторые из них, в частности методы одномерной оптимизации $f(x) \to \min, a \le x \le b$ и многомерной оптимизации $f(x_1, x_2, \ldots, x_n) \to \min,\ \ x \in \Omega \subset R^n$.

4.1.1. Методы одномерной оптимизации

4.1.1.1. Метод перебора

Метод перебора является простейшим из прямых методов минимизации. Пусть $f(x)$ - непрерывная функция на интервале $[a, b]$ и требуется найти точку минимума x^* функции $f(x)$ на этом отрезке с абсолютной погрешностью ε. Отрезок $[a, b]$ разбивается на *n* равных частей точками деления $x_i = a + h_i$, $h = \dfrac{b-a}{n}$, $i = 0, 1, \ldots, n$. Вычислив значения $f(x)$ в этих точках, путем сравнения найдем точку x_m, для которой $f(x_m) = \min\limits_{0 \le i \le n} f(x_i)$.

Далее полагаем, что $x^* \approx x_m$, $f^* \approx f(x_m)$. При этом максимальная погрешность определения точки x^* равна $\varepsilon = \dfrac{b-a}{n}$.

Существует ряд специальных методов поиска оптимальных решений с разными способами выбора узлов x_i и сужения интервала неопределенности: метод деления отрезка пополам, метод золотого сечения и др.

4.1.1.2. Метод золотого сечения

Деление отрезка на две неравные части таким образом, чтобы отношение длины всего отрезка к длине большей его части было равно отношению длины большей части к длине меньшей части, называется *золотым сечением* этого отрезка (рис. 5).

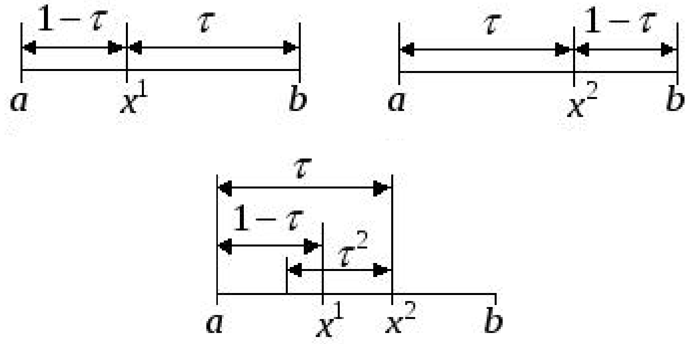


Рис. 5. Золотое сечение отрезка

$$\frac{1}{\tau} = \frac{\tau}{1-\tau}; \quad 1-\tau = \tau^2; \ \tau = \frac{\sqrt{5}-1}{2} \approx 0{,}618.$$

Метод золотого сечения для функции одной переменной состоит в построении последовательности отрезков, стягивающейся к точке минимума функции $f(x)$, таких, что $[a_0, b_0] \supset [a_1, b_1], \ldots$.

На первом шаге процесса оптимизации внутри отрезка $[a_0, b_0]$ выбираем две точки x_1 и x_2 $(x_1 < x_2)$, вычисляем значения целевой функции – $f(x_1)$ и $f(x_2)$. Пусть $f(x_1) < f(x_2)$. Тогда очевидно, что минимум расположен на одном из прилегающих к точке x_1 отрезков: $[a_0, x_1]$ или $[x_1, x_2]$. Поэтому отрезок $[x_2, b_0]$ можно отбросить, сузив тем самым первоначальный интервал поиска минимума.

Второй шаг производим на отрезке $[a_1, b_1]$, где $a_1 = a_0$, $b_1 = x_2$. Нужно снова выбрать две внутренние точки таким образом, чтобы одна из них (x_1) осталась из предыдущего шага; поэтому достаточно выбрать точку x_3, вычислить значение $f(x_3)$ и произвести сравнение.

Процесс оптимизации повторяется до тех пор, пока длина очередного отрезка $[a_n, b_n]$ не станет меньше заданной длины ε.

Деление отрезка на две неравные части производится точками золотого сечения.

Точки x_1 и x_2 (рис.6) являются точками золотого сечения отрезка $[a, b]$. При этом точка x_1 осуществляет золотое сечение отрезка $[a, x_2]$, а точка x_2 – отрезка $[x_1, b]$. Это позволяет на каждом шаге, за исключением первого, вычисление значения функции $f(x)$ производить лишь один раз.

Алгоритм метода золотого сечения

Строится последовательность интервалов

$$[a_0, b_0] \supset [a_1, b_1] \supset \ldots \supset [a_k, b_k] \supset x^*,$$

таких, что $|a_k - b_k| \to 0$, следующим образом. Предположим, что выполнено k шагов. Вычисляем

$$x_1 = b_k - \tau \cdot [b_k - a_k] = a_k + (1-\tau) \cdot [b_k - a_k]; \qquad x_2 = a_k + \tau \cdot [b_k - a_k];$$

$$\tau = 0{,}618; \quad 1 - \tau = 0{,}382.$$

Если $f(x) < f(y)$, то $a_{k+1} = a_k,\ \ b_{k+1} = y$, иначе $a_{k+1} = x,\ \ b_{k+1} = b_k$.

Вычислительный процесс заканчивается, когда $|a_n - b_n| < \varepsilon$, где ε – требуемая точность.

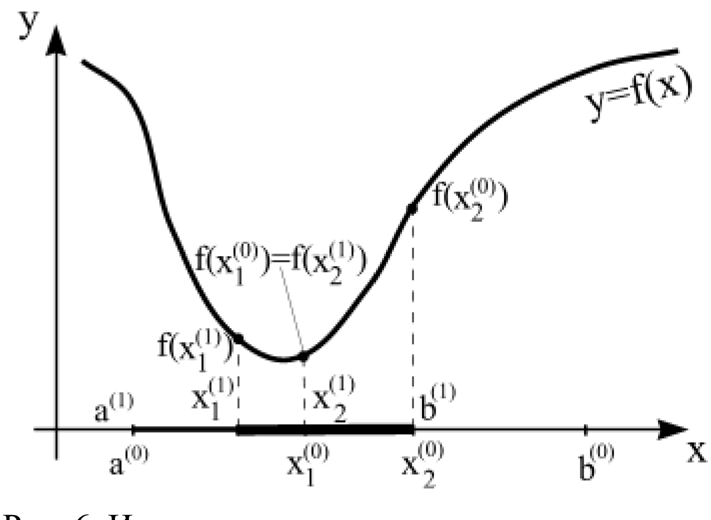


Рис. 6. Иллюстрация метода золотого сечения

4.1.2. Методы многомерной оптимизации

4.1.2.1. Метод покоординатного спуска

Идея всех методов спуска состоит в том, чтобы исходя из начального приближения – точки $x^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)}\right) \in D$ – перейти в следующую точку $x^{(1)}=\left(x_1^{(1)}, x_2^{(1)}, ..., x_n^{(1)}\right)\in D$ так, чтобы значение $f\left(x_1, x_2, ..., x_n\right)$ уменьшилось:

$$f\left(x^{(1)}\right)< f\left(x^{(0)}\right).$$

Метод *покоординатного спуска* заключается в следующем.

Пусть в области D задано нулевое приближение

$$x^{(0)}=\left(x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)}\right)\in D.$$

Рассматриваем функцию $f\left(x_1, x_2, ..., x_n\right)$ при фиксированных значениях $x_2, x_3, \ldots, x_n$ как функцию одной переменной x_1, т.е. на первом шаге находим

$$\min_{x_i\in D} f\left(x_1, x_2^{(0)}, ..., x_n^{(0)}\right).$$

Значение x_1, доставляющее минимум, обозначим через $x_1^{(1)}$:

$$f\left(x_1^{(1)}, x_2^{(0)}, ..., x_n^{(0)}\right)\le f\left(x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)}\right).$$

Далее при фиксированных значениях $x_1^{(1)}, x_3^{(0)}, ..., x_n^{(0)}$ рассматриваем $f\left(x_1^{(1)}, x_2, x_3^{(0)}, ..., x_n^{(0)}\right)$ как функцию одной переменной x_2.

Находим $\min\limits_{x_2\in D} f\left(x_1^{(1)}, x_2, x_3^{(0)}, ..., x_n^{(0)}\right)$.

Значение x_2, доставляющее минимум, обозначим через $x_2^{(1)}$, получаем:

$$f\left(x_1^{(1)}, x_2^{(1)}, x_3^{(0)}, ..., x_n^{(0)}\right)\le f\left(x_1^{(1)}, x_2^{(0)}, x_3^{(0)}, ..., x_n^{(0)}\right)$$

и т.д. После (n–1) шагов:

$$f\left(x_1^{(1)}, ..., x_n^{(1)}\right)\le f\left(x_1^{(1)}, x_2^{(1)}, ..., x_n^{(0)}\right).$$

В результате n шагов покоординатного спуска происходит переход из начальной точки $x^{(0)}=\left(x_1^{(0)}, ..., x_n^{(0)}\right)$ в точку $x^{(1)}=\left(x_1^{(1)}, ..., x_n^{(1)}\right)$. Если при этом оказывается, что $f\left(x_1^{(1)}, ..., x_n^{(1)}\right)=f\left(x_1^{(0)}, ..., x_n^{(0)}\right)$, то начальная точка $x^{(0)}$ доставляет минимум функции $f(x)$. Если $f\left(x^{(1)}\right)< f\left(x^{(0)}\right)$, то выполняются следующие n шагов покоординатного спуска, при этом за начальную точку принимается $x^{(1)}$. В результате получаем $x^{(2)}$ такую, что $f\left(x^{(2)}\right)< f\left(x^{(1)}\right)$, и т.д. Этот процесс продолжается до тех пор, пока не выполнится какое-либо условие окончания процесса, например:

$$\left|f(x^{(k+1)})-f(x^{(k)})\right|<\varepsilon, \tag{18}$$

где ε – заданная точность.

Таким образом, метод покоординатного спуска сводит задачу о нахождении наименьшего значения функции многих переменных к многократному решению одномерной задачи оптимизации по каждой компоненте вектора x.

4.1.2.2. Прямой поиск, или метод Хука и Дживса (метод конфигураций)

Алгоритм метода прямого поиска включает два этапа:

1) исследующий поиск вокруг базисной точки, который используется для выбора направления минимизации;

2) поиск по образцу, включающий в себя минимизацию по выбранному направлению.

Вычислительная схема имеет следующий вид:

1. Задается начальное значение вектора $x^{(0)}$ и начальное приращение $\Delta x^{(0)}$, вычисляется $f\left(x^{(0)}\right)$.

2. В циклическом порядке изменяется каждая переменная на выбранном значении $\pm\Delta x^{(0)}$, пока все переменные не будут изменены. При этом если $f\left(x_1^{(0)} \pm \Delta x_1^{(0)}, \ldots, x_i^{(0)} \pm \Delta x_i^{(0)}, x_{i+1}^{(0)}, \ldots, x_n^{(0)}\right) < f\left(x^{(0)}\right)$, то для вектора первого приближения $x^{(1)}$ принимаем $x_i^{(1)} = x_i^{(0)} \pm \Delta x_i^{(0)}$, иначе $x_i^{(1)} = x_i^{(0)}$.

То есть на каждом сдвиге по переменной x_i значение целевой функции сравнивается с ее значением в предыдущей точке. Если целевая функция уменьшается на этом шаге, то ее старое значение заменяется на новое, используемое при последующих сравнениях.

3. Осуществляется поиск по образцу: эффективные изменения переменных определяют некоторый вектор в пространстве Δx, т.е. вектор $\left(\ldots, 0, \ldots, \pm\Delta x_i, \ldots, 0, \ldots, \Delta x_j, \ldots\right)$, у которого ненулевые компоненты расположены на местах, соответствующих номерам переменных, изменение которых привело к уменьшению целевой функции.

Вектор Δx указывает направление минимизации. Следует продолжить движение по этому направлению, выполнив один или несколько шагов (до тех пор, пока функция не перестанет убывать).

4. Далее опять используется исследующий поиск.

5. Успех поиска по образцу оценивается только после очередного исследующего поиска.

Если функция не уменьшается в результате исследующего поиска, то уменьшают значение шага Δx и снова проводят исследующий поиск. Такой процесс повторяют до выполнения условия $\|\Delta x\| < \varepsilon$, где ε – константа, задающая точность поиска минимума.

4.1.2.3. Комплексный поиск Бокса

Для реализации метода комплексного поиска Бокса используется следующий алгоритм:

1. Строится многогранник с $2n$ вершинами (комплекс), для чего выбирается $2n$ точек из пространства E^n по формуле

$$x_i^{(0)} = l_i + r_i \cdot |u_i - l_i|, \quad i = 1, 2, \dots, 2n, \quad l_i \le x_i \le u_i, \tag{19}$$

где $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$, $l_i = (l_{i1}, l_{i2}, \dots, l_{in})$, $u_i = (u_{i1}, u_{i2}, \dots, u_{in})$, r_i – диагональная матрица псевдослучайных чисел, равномерно распределенных на интервале $[0, 1]$.

2. Целевая функция вычисляется в каждой вершине, и вершина, в которой $f(x)$ имеет наихудшее значение, заменяется новой вершиной, находящейся на прямой, проходящей через наихудшую вершину и центр тяжести оставшихся точек.

Координаты центра тяжести можно определить по формуле

$$x_{cj}^{(k)} = \frac{1}{2n-1}\left(\sum_{i=1}^{2n} x_{ij} - x_{hj}\right), \tag{20}$$

где j – номер координаты вектора x_i;
i – номер точки;
h – номер удаляемой точки.

Координаты новой точки определяются по формуле

$$x_{hj}^{(k+1)} = x_{cj} + \alpha(x_{cj} - x_{hj}^{(k)}), \quad j = 1, 2, \dots, 2n, \tag{21}$$

где $\alpha \ge 1$ – коэффициент отражения (например, $\alpha = 1{,}3$).

3. Если значение $f(x)$ в найденной новой вершине остается наихудшим, то она заменяется другой, расположенной на расстоянии, равном половине расстояния от новой вершины до центра тяжести. (Если нарушается ограничение $l_i \le x_i \le u_i$, то новая вершина также передвигается на половину расстояния к центру тяжести.)

4. Поиск продолжается до тех пор, пока многогранник не будет стянут в центр тяжести (в пределах заданной точности).

4.1.2.4. Повторяющийся случайный поиск

Метод случайного поиска представляет собой выбор наилучшего направления минимизации из многих возможных, получаемых случайным путем, и переход к новой точке с минимальным значением функции в этом направлении.

Алгоритм метода.

1. Задается начальная точка $x^{(0)}$.

2. Строится случайная траектория из последовательности шагов по формуле

$$x^{(k+1)} = x^{(k)} + \lambda^{(k)}\left[\beta\frac{z^{(k)}}{\left\|z^{(k)}\right\|} + (1-\beta)\, r^{(k)}\right],$$

где $\lambda^{(k)}$ – числовой параметр, увеличивающийся при удачном и уменьшающийся при неудачном шаге (можно начать с $\lambda^{(k)} = 1$);

β – числовой параметр, изменяемый в процессе поиска (можно принять $\beta = 0{,}5$);

$z^{(k)}$ – вектор «предыстории», учитывающий среднее направление поиска на предыдущих шагах:

$$z^{(k+1)} = \gamma z^{(k)} + (1-\gamma)\left(x^{(k+1)} - x^{(k)}\right)s^{(k)};$$

$r^{(k)}$ – единичный вектор, компоненты которого определяются случайным образом с помощью генератора псевдослучайных чисел на интервале [0, 1], а затем весь вектор нормируется (делится на длину);

$s^{(k)}$ – вектор масштабных множителей ($s^{(k)} = 1$);

γ – постоянный весовой множитель $\gamma = 0{,}5$.

3. Вектор $x^{(k+1)}$ будет принят или отвергнут в зависимости от того, выполняется или нет неравенство $f\left(x^{(k+1)}\right) < f\left(x^{(k)}\right)$. Если $x^{(k+1)}$ принят, то $\lambda^{(k+1)} = \lambda^{(k)} \cdot \alpha$, если нет, то $\lambda^{(k+1)} = \dfrac{\lambda^{(k)}}{\alpha}$, $\alpha > 1$, например $\alpha = 1{,}3$.

4.2. Методы первого порядка

Рассмотрим методы поиска, в которых наряду со значениями функции используется и ее градиент.

Напомним, что градиент функции $f(x)$ в точке $x^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, ..., x_n^{(k)}\right)$ определяется формулой

$$\nabla f(x^{(k)}) = \left(\frac{\partial f\left(x^{(k)}\right)}{\partial x_1}, \frac{\partial f\left(x^{(k)}\right)}{\partial x_2}, ..., \frac{\partial f\left(x^{(k)}\right)}{\partial x_n}\right)^T.$$

Вектор $\nabla f(x^{(k)})$ или $\operatorname{grad} f(x)$ ортогонален линиям уровня $f(x) = c = \mathrm{const}$; его направление совпадает с направлением наибольшего возрастания функции $f(x)$ в заданной точке. Следовательно, вектор $(-\operatorname{grad} f(x))$, противоположный градиенту и называемый антиградиентом, направлен в сторону наискорейшего убывания функции (*наискорейшего спуска*). В точке минимума функции $\operatorname{grad} f(x) = 0$.

4.2.1. Метод градиентного спуска

Идея метода градиентного спуска состоит в следующем. Выбираем некоторую начальную точку и вычисляем в ней градиент рассматриваемой функции. Делаем шаг в направлении антиградиента. В результате приходим в точку, значение функции в которой обычно меньше первоначального. Если это условие не выполнено, т.е. значение функции не изменилось либо даже возросло, то нужно уменьшить шаг. В новой точке процедуру повторяем: вычисляем градиент и снова делаем шаг в обратном к нему направлении. Процесс продолжается до получения наименьшего значения целевой функции. Момент окончания поиска наступит тогда, когда движение из полученной точки с любым шагом приведет к возрастанию целевой функции.

Определим итерационный процесс:

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)} = x^{(k)} + \lambda^{(k)} s^{(k)},$$

где $\Delta x^{(k)} = \lambda^{(k)} s^{(k)}$ – шаг минимизации;

$\lambda^{(k)}$ – скаляр;

$s^{(k)}$ – единичный вектор в направлении минимизации,

$$s^{(k)} = -\frac{\nabla f\left(x^{(k)}\right)}{\left\|\nabla f\left(x^{(k)}\right)\right\|}.$$

Следовательно,

$$x^{(k+1)} = x^{k} - \frac{\lambda^{(k)} \nabla f\left(x^{(k)}\right)}{\left\|\nabla f\left(x^{(k)}\right)\right\|}. \tag{22}$$

Поскольку один шаг в направлении наискорейшего спуска в общем случае не приводит в точку минимума $f(x)$, формула (19) должна применяться многократно до тех пор, пока не будет достигнут минимум. В точке минимума $\left\|\nabla f\left(x^{(k)}\right)\right\| < \varepsilon$.

Процедура поиска может закончиться в стационарной точке (в которой $\nabla f\left(x^{(k)}\right) = 0$) различного типа. Поэтому иногда следует определять, является ли данная точка точкой локального минимума (т.е. решением) или седловой точкой. Если это седловая точка, то необходимо выйти из нее, применив какой-либо метод нулевого порядка, после чего минимизация может опять продолжаться градиентным методом. Тип стационарной точки можно определить с помощью матрицы Гёссе $\nabla^2 f\left(x^{(k)}\right) = \left[\frac{\partial^2 f\left(x^{(k)}\right)}{\partial x_i \partial x_j}\right]$: если эта матрица является положительно-определенной, то имеем точку минимума, иначе стационарная точка является седловой.

Критерий окончания поиска основывают:

a) на значении $f(x)$ (когда уменьшение функции не происходит ни в одном направлении);

b) на значении $\nabla f\left(x^{(k)}\right)$ (когда вектор $\nabla f\left(x^{(k)}\right)$ близок к 0):

$$\left\|\nabla f\left(x^{(k+1)}\right)\right\| \le \varepsilon ,$$

где ε – заданная точность,

$$\left\|\nabla f\left(x^{(k+1)}\right)\right\| = \sqrt{\sum_{i=1}^{n}\left[\frac{\partial f\left(x^{(k+1)}\right)}{\partial x_i}\right]^2};$$

c) на комбинации значений $f(x)$ и $\nabla f\left(x^{(k)}\right)$.

Важным вопросом является выбор величины шага $\lambda^{(k)}$. Обычно используется один из следующих способов выбора вектора $\lambda^{(k)}$:

1) $\lambda^{(k)} = \text{const}$;

2) $\lambda^{(k)} \to 0$ при $k \to \infty$;

3) целевая функция минимизируется по λ:

$$\min_{\lambda} f\left(x^{(k)} + \lambda s^{(k)}\right).$$

Для этого значение λ может быть вычислено из уравнения $\dfrac{df\left(x^{(k)} + \lambda s^{(k)}\right)}{d\lambda} = 0$. Минимум функции $\min\limits_{\lambda} f\left(x^{(k)}\right)$ можно определить также каким-либо численным методом.

Имеет место следующая *теорема сходимости градиентного метода минимизации*.

Если $f(x)$ – выпуклая функция, имеющая производные до третьего порядка, то метод наискорейшего спуска сходится из любого начального приближения.

При использовании градиентного спуска в задачах оптимизации основной объем расчетов приходится обычно на вычисление градиента целевой функции в каждой точке траектории спуска. Поэтому целесообразно уменьшить количество таких точек без ущерба для самого уравнения. Это достигается в методе *наискорейшего градиентного спуска* (рис. 7) следующим образом. После определения в начальной точке направления, противоположного градиенту целевой функции, делают не один шаг, а несколько, продолжая двигаться в этом направлении, пока целевая функция убывает, таким образом достигая минимума по направлению в некоторой точке. В этой точке снова определяют направление спуска (с помощью антиградиента) и ищут новую точку минимума целевой функции и т.д. В этом методе спуск происходит гораздо более крупными шагами, и градиент вычисляется в меньшем числе точек.

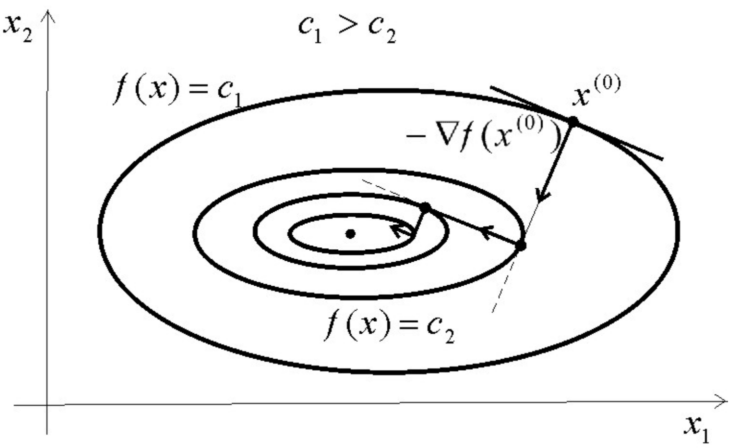


Рис. 7. Иллюстрация метода наискорейшего градиентного спуска

4.2.2. Метод сопряженного градиента Флетчера – Ривса

Пусть $Q_{n\times n}$ – положительно-определенная матрица; тогда, если для векторов $s^{(k)} = \left(s_1^{(k)}, ..., s_n^{(k)}\right)$ и $s^{(k+1)} = \left(s^{(k+1)}, ..., s_n^{(k+1)}\right)$ выполняется равенство $s^{(k)} \cdot Q \cdot (s^{(k+1)})^T = 0$, то векторы $s^{(k)}$ и $s^{(k+1)}$ называются *сопряженными* по направлению.

Известно, что использование сопряженных направлений на каждом шаге минимизации приводит к хорошему результату. При этом в качестве матрицы Q при вычислении сопряженных направлений применяется матрица Гёссе $\nabla^2 f$.

Известен следующий алгоритм метода сопряженного градиента Флетчера – Ривса для минимизации функции (рис. 8):

1) задается точка $x^{(0)}$;

2) вычисляется $s^{(0)} = -\nabla f(x^{(0)})$;

3) на k-м шаге с помощью одномерного поиска в направлении $s^{(k)}$ находится минимум функции $f(x)$, что определяет точку $x^{(k+1)}$;

4) вычисляются $f(x^{(k+1)})$ и $\nabla f(x^{(k+1)})$;

5) направление поиска $s^{(k)}$ определяется по формуле

$$s^{(k+1)} = -\nabla f(x^{(k+1)}) + \beta_0 s^{(0)}, \tag{23}$$

где $\beta_0 = \dfrac{\left|\nabla f(x^{(k+1)})\right|^2}{\left|\nabla f(x^{(k)})\right|^2}$;

6) после $(n+1)$-й итерации (при $k=n$) процедура циклически повторяется с заменой $x^{(0)}$ на $x^{(n+1)}$:

$$x^{(k+2)} = x^{(k+1)} + \lambda^{(k)} s^{(k+1)}, \tag{24}$$

где $\lambda^{(k)}$ – шаг;

7) алгоритм заканчивается, когда $\left\| s^{(k)} \right\| < \varepsilon$, где ε – произвольная малая величина.

Замечание. Выбор величины шага $\lambda^{(k)}$, как и в других градиентных методах, возможен одним из следующих способов: 1) $\lambda^{(k)} = \text{const}$; 2) $\lambda^{(k)} \to 0$ при $k \to \infty$; 3) целевая функция минимизируется по λ: $\min\limits_{\lambda} f\left(x^{(k)} + \lambda s^{(k)}\right)$.

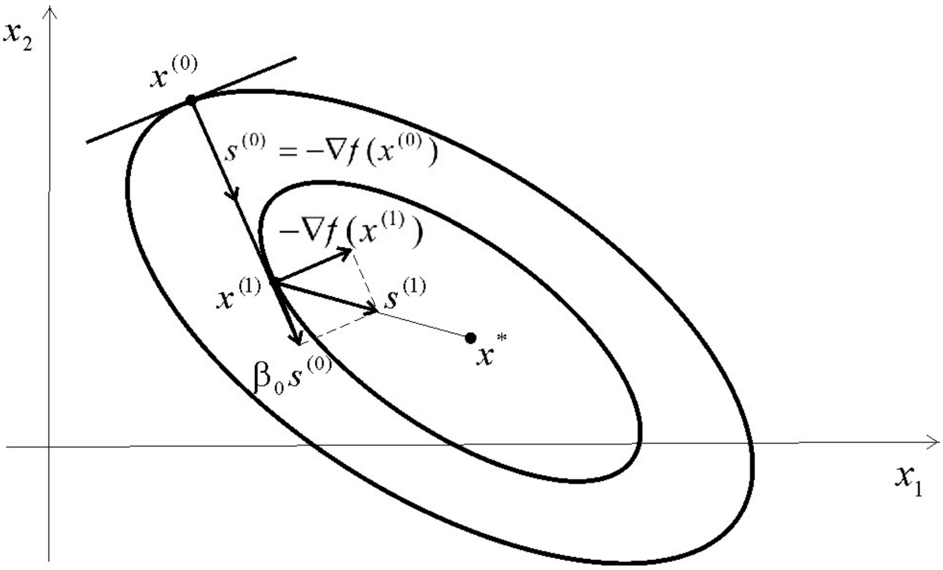


Рис. 8. Иллюстрация метода сопряженных градиентов

4.3. Методы второго порядка

Если целевая функция дважды дифференцируема в R^n, то для поиска точки минимума можно использовать не только градиент функции, но и матрицу Гёссе, повышая скорость сходимости итерационного процесса. Наиболее «мощными» и точными среди таких методов минимизации являются методы *ньютоновского типа*, классическим среди которых является метод Ньютона.

4.3.1. Метод Ньютона

Если частные производные минимизируемой функции приравнять к 0 и решать полученную систему нелинейных уравнений методом Ньютона, можно получить следующую схему минимизации:

$$x^{(k+1)} = x^{(k)} - \left[\nabla^2 f\left(x^{(k)}\right)\right]^{-1} \cdot \nabla f\left(x^{(k)}\right), \tag{25}$$

где
$$\nabla^2 f\left(x^{(k)}\right) = \begin{pmatrix} \frac{\partial^2 f\left(x^{(k)}\right)}{\partial x_1 \partial x_1} & \frac{\partial^2 f\left(x^{(k)}\right)}{\partial x_1 \partial x_2} \cdots & \frac{\partial^2 f\left(x^{(k)}\right)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f\left(x^{(k)}\right)}{\partial x_2 \partial x_1} & \frac{\partial^2 f\left(x^{(k)}\right)}{\partial x_2 \partial x_2} \cdots & \frac{\partial^2 f\left(x^{(k)}\right)}{\partial x_2 \partial x_n} \\ \cdots\cdots\cdots & \cdots\cdots\cdots & \cdots\cdots \\ \frac{\partial^2 f\left(x^{(k)}\right)}{\partial x_n \partial x_1} & \frac{\partial^2 f\left(x^{(k)}\right)}{\partial x_n \partial x_2} \cdots & \frac{\partial^2 f\left(x^{(k)}\right)}{\partial x_n \partial x_n} \end{pmatrix}.$$

Итерационную схему метода Ньютона можно также получить следующим путем. Представим $f\left(x^{(k+1)}\right)$ в виде ряда Тейлора для функции n переменных до 2-го порядка (квадратичная аппроксимация) в точке $x^{(k)}$:

$$f\left(x^{(k+1)}\right) = f\left(x^{(k)}\right) + \nabla^T f\left(x^{(k)}\right)\Delta x^{(k)} + \frac{1}{2}\left(\Delta x^{(k)}\right)^T \nabla^2 f\left(x^{(k)}\right)\left(\Delta x^{(k)}\right) + \ldots + O\left(\left(\Delta x^{(k)}\right)^2\right). \tag{26}$$

Определим наилучшее направление Δx путем дифференцирования $f\left(x^{(k+1)}\right)$ по каждой компоненте Δx и приравнивания к нулю полученных выражений:

$$0 = \nabla^T f\left(x^{(k)}\right) + \nabla^2 f\left(x^{(k)}\right)\Delta x^{(k)}.$$

Это приводит к выражению

$$\Delta x^{(k)} = -\left[\nabla^2 f\left(x^{(k)}\right)\right]^{-1} \nabla f\left(x^{(k)}\right), \tag{27}$$

или

$$x^{(k+1)} = x^{(k)} - \left[\nabla^2 f\left(x^{(k)}\right)\right]^{-1} \nabla f\left(x^{(k)}\right). \tag{28}$$

Заметим, что здесь и направление, и величина шага точно определены. Примечательно, что такая схема минимизации решает задачу нахождения минимума квадратичной формы с любого приближения за один шаг. Но в случае общей нелинейной целевой функции нельзя достичь минимума $f(x)$ за один шаг; поэтому вместо уравнения (28) используют следующую схему:

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} \frac{\left[\nabla^2 f\left(x^{(k)}\right)\right]^{-1} \nabla f\left(x^{(k)}\right)}{\left\| \left[\nabla^2 f\left(x^{(k)}\right)\right]^{-1} \nabla f\left(x^{(k)}\right) \right\|}. \tag{29}$$

Сложность метода заключается в определении $\lambda^{(k)}$ (аналогично методу Ньютона при решении СНАУ), в вычислении и обращении матрицы Гёссе.

Методические рекомендации к теме 4

Знать особенности методов минимизации нулевого, первого и второго порядков, иметь представление о влиянии шага на точность и скорость решения задачи нахождения оптимума. Знать преимущество методов минимизации, использующих производные функции, перед методами, не использующими производные.

Уметь составлять блок-схемы алгоритмов методов безусловной минимизации и реализовывать их на языке программирования.

Уметь отвечать на контрольные вопросы по теме и решать поставленные задачи.

Контрольные вопросы и задания

1. Приведите характерные отличия методов минимизации нулевого, первого и второго порядков.
2. Влияет ли шаг в методе перебора на точность, на скорость решения задачи нахождения оптимума?
3. Если отрезок $[a, b]$ содержит внутреннюю точку c, то какое условие называется золотым сечением?
4. Всегда ли метод золотого сечения гарантированно дает решение?
5. В чем преимущество методов минимизации, использующих производные функции, перед методами, не использующими производные? Обоснуйте на примере.

5. ЭЛЕМЕНТЫ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Напомним общую формулировку задачи нелинейного программирования (ЗНП).

Найти оптимум (минимум или максимум) некоторой целевой функции

$$f(x_1, ..., x_n) \to \min_{x_1, ..., x_n}$$

при заданных ограничениях на переменные $(x_1, x_2, \ldots, x_n)$ в виде равенств и неравенств:

$$\begin{cases} \Phi_i(x_1, ..., x_n) \geq 0, & i \in I; \\ \Phi_l(x_1, ..., x_n) = 0, & l \in L. \end{cases}$$

Здесь f, Φ_i, Φ_l – в общем случае нелинейные функции.

5.1. Метод множителей Лагранжа

Это один из немногих аналитических методов решения ЗНП, позволяющий в ряде случаев найти точное решение задачи с ограничениями типа равенств. Рассмотрим его на примере решения ЗНП с двумя переменными x, y и одним ограничением, заданным в виде равенства:

$$\begin{cases} z = f(x, y) \to \min\limits_{x,y}; \\ \varphi(x, y) = 0. \end{cases}$$

Может возникнуть три случая решения:

1) Пусть равенство $\varphi(x, y) = 0$ можно преобразовать к виду $y = \psi(x)$. Тогда $z = f(x, \psi(x))$, и задача сводится к задаче на нахождение минимума функции одной переменной.

2) $\varphi(x, y) = 0$ можно преобразовать к параметрическому заданию: $y = n(t),\ x = m(t)$. Тогда $z = f(m(t), n(t))$ – также задача на нахождение минимума функции одной переменной.

3) $\varphi(x, y) = 0$ не преобразуется ни к первому, ни ко второму виду. Вычислим полную производную функции z по переменной x (считая y зависящей от x):

$$\frac{dz}{dx} = f'_x + f'_y \cdot y'_x.$$

Так как $\dfrac{d\varphi}{dx} = \varphi'_x + \varphi'_y \cdot y'_x$, а $\dfrac{d\varphi}{dx} = 0$, то $y'_x = -\dfrac{\varphi'_x}{\varphi'_y}$ и $\dfrac{dz}{dx} = f'_x - f'_y \dfrac{\varphi'_x}{\varphi'_y}$.

В точке экстремума $\frac{dz}{dx} = 0$, следовательно,

$$f'_x - f'_y \frac{\varphi'_x}{\varphi'_y} = 0 \quad \text{или} \quad \frac{f'_x}{\varphi'_x} = \frac{f'_y}{\varphi'_y} = -\lambda,$$

где λ – некоторая константа.

Получили систему уравнений:

$$\begin{cases} f'_x(x, y) + \lambda \cdot \varphi'_x(x, y) = 0, \\ f'_y(x, y) + \lambda \cdot \varphi'_y(x, y) = 0, \\ \varphi(x, y) = 0, \end{cases}$$

откуда находим искомые значения x, y, т.е. точку, подозрительную на экстремум.

Если ввести следующее обозначение:

$$\Phi(x, y) = f(x, y) + \lambda\varphi(x, y),$$

то систему можно записать в более удобной форме:

$$\begin{cases} \Phi'_x = 0, \\ \Phi'_y = 0, \\ \Phi'_\lambda = 0. \end{cases}$$

Заметим, что после определения точки, подозрительной на экстремум, выяснить, является ли эта точка точкой минимума, максимума или стационарной точкой, можно, как правило, после дополнительного анализа поведения целевой функции в окрестности найденной точки.

5.2. Метод штрафных функций

Рассмотрим задачу нелинейного программирования в виде $f(X) \to \min\limits_X$, $X = (x_1, x_2, ..., x_n)$ при ограничениях

$$\varphi_i(X) \geq 0, \ i = 1, 2, ..., m.$$

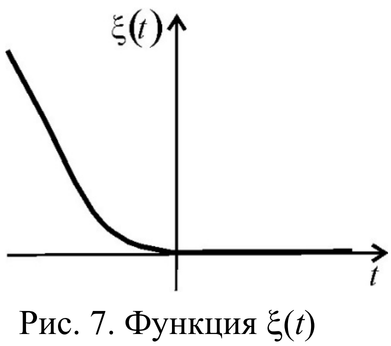


Рис. 7. Функция $\xi(t)$

Введем в рассмотрение функцию $\xi(t)$ (рис. 7), удовлетворяющую следующим условиям:

$$\xi(t) = 0 \quad \text{при } t \geq 0,$$
$$\xi(t) > 0 \quad \text{при } t < 0,$$
$$\xi(t_2) > \xi(t_1) \quad \text{при } t_2 < t_1 < 0,$$
$$\lim_{t \to -\infty} \xi(t) = \infty.$$

Функция $\Phi(X,R)=\sum_{i=1}^{m} r_i \xi(\varphi_i(x))$, где $R=(r_1, r_2, ..., r_m)$, $r_i \geq 0$, называется *штрафной функцией*.

Нетрудно заметить, что $\Phi(X,R)=0$ при выполнении ограничений и $\Phi(X,R)>0$ при их невыполнении.

Идея *метода штрафных функций* заключается в преобразовании задачи нелинейного программирования по минимизации функции с ограничениями к решению последовательности задач безусловного минимума вспомогательной функции

$$\Psi(X)=f(X)+\Phi(X,R).$$

Очевидно, что при достаточно больших значениях величины r_i $\min\Psi(X)$ достигается тогда, когда выполнены все ограничения $\varphi_i \geq 0$ и функция $f(X)$ принимает возможно минимальное значение.

Методы штрафной функции делятся на методы внутренней и внешней точки.

Методы *внутренней точки* используются тогда, когда заранее известна точка $X^{(0)}$, для которой выполняются все ограничения $\varphi_i \geq 0$. В этом случае в качестве $\Phi(X,R)$ можно выбрать одну из функций:

$$\begin{aligned} \Phi_1(X,R) &= -r^{(k)}\sum_{i=1}^{m} \ln \varphi_i(X), \\ \Phi_2(X,R) &= r^{(k)}\sum_{i=1}^{m} \frac{1}{\varphi_i(X)}. \end{aligned} \tag{30}$$

Алгоритм метода внутренней точки

1. Задать начальную точку $x^{(0)}$ из области допустимых значений целевой функции и малое число $\varepsilon > 0$ – точность решения.

2. Выбрать начальное значение вектора $R^{(0)}=\left(r_1^{(0)}, r_2^{(0)}, ..., r_m^{0}\right)$ так, чтобы компоненты его были сравнимы со значением функции $f\left(x^{(0)}\right)$. Принять $k=0$.

3. Составить вспомогательную функцию $\Psi(X)=f(X)+\Phi(X,R)$.

4. Найти точку $x^*(r^{(k)})$ безусловного минимума функции $\psi(X)$ с помощью какого-либо метода минимизации нулевого, первого или второго порядка. В качестве начальной точки взять $x^{(k)}$. Вычислить штрафную функцию $\Phi\left(x^*(r^{(k)}), r^{(k)}\right)$.

5. Проверить условие окончания: если достигнута требуемая точность $\Phi\left(x^*(r^{(k)}), r^{(k)}\right) \le \varepsilon$, процесс поиска закончить:

$$f\left(x^*\right) = f\left(x^*(r^{(k)})\right).$$

В противном случае изменить значения вектора R: $r^{(k+1)} = cr^{(k)}$, где $c > 1$; принять $x^{(k+1)} = x^*(r^{(k)}),\quad k = k + 1$ и перейти к шагу 3.

Метод *внешней точки* необходимо использовать, когда начальные приближения $x_1^{(0)},\quad x_2^{(0)}$, для которых ограничения выполняются, неизвестны. В качестве $\Phi(X,R)$ в этом случае можно принять, например, функцию

$$\Phi(X,R) = r^{(k)} \sum_{i=1}^{m} \left(\varphi_i(X) - |\varphi_i(X)|\right)^2. \tag{31}$$

Алгоритм метода внешней точки аналогичен алгоритму метода внутренней точки.

5.3. Седловая точка для задачи нелинейного программирования

Рассмотрим задачу на отыскание минимума функции $F(x_1, x_2, \ldots, x_n)$ при заданных ограничениях:

$$\begin{cases} \min\limits_{x_1, \ldots, x_n} F(x_1, \ldots, x_n); \\ \varphi_j(x_1, \ldots, x_n) \le 0,\ \ j = 1, \ldots, m; \\ x_{i_k} \ge 0,\ \ k = \{1, 2, \ldots, l\}. \end{cases}$$

Построим функцию Лагранжа аналогично функции Лагранжа для решения задач выпуклого программирования. Для этого введем в рассмотрение вектор

$$Y = (y_1, y_2, \ldots, y_m),\quad y_j \ge 0,\ \ j = 1, 2, \ldots, m$$

и определим функцию

$$L(X,Y) = -F(X) - \sum_{j=1}^{m} y_j \cdot \varphi_j(x). \tag{32}$$

Точка $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ называется точкой Куна – Таккера для задачи нелинейного программирования, если существует точка $Y^* = (y_1^*, y_2^*, \ldots, y_m^*)$, $y_j^* \ge 0,$ для которой выполняются следующие условия:

1) $\dfrac{\partial L}{\partial x_i}(X^*, Y^*) \le 0$ для $x_i \ge 0$;

2) $\dfrac{\partial L}{\partial x_i}(X^*, Y^*) = 0$ для $x_i < 0$;

3) $x_j^* \dfrac{\partial L}{\partial x_i}(X^*, Y^*) = 0$ для $x_i \ge 0$;

4) $\dfrac{\partial L}{\partial y_j}(X^*, Y^*) \ge 0$ для $j = 1, 2, ..., m$;

5) $y_j^* \dfrac{\partial L}{\partial y_j}(X^*, Y^*) = 0$ для $j = 1, 2, ..., m.$

Теорема.

Если точка X^* является оптимальным решением задачи нелинейного программирования, а функции F_j и φ_j дифференцируемы, то точка X^* является точкой Куна – Таккера задачи нелинейного программирования.

Методические рекомендации к теме 5

Знать отличие задачи нелинейного программирования от задачи линейного программирования, различие методов внутренней и внешней точки, уметь составлять условия теоремы Куна – Таккера и решать ЗНП.

Контрольные вопросы и задания

1. Что такое штрафная функция?

2. Чем функция внутреннего штрафа отличается от функции внешнего штрафа? Укажите способы задания штрафных функций.

ЗАКЛЮЧЕНИЕ

Рассмотрены численные методы и алгоритмы решения задач математического программирования, даны методические рекомендации к освоению материала.

Настоящее учебно-методическое пособие предназначено для усвоения лекционного курса по дисциплине «Численные методы и методы оптимизации», читаемого студентам бакалавриата по направлению 09.03.02 «Информационные системы и технологии» Пензенского государственного университета архитектуры и строительства, а также тем, вынесенных на самостоятельное изучение.

Здесь излагаются основные вопросы по элементам линейного программирования, выпуклого программирования, рассматриваются методы безусловной оптимизации, методы отыскания экстремумов функций без ограничений и методы численного решения задачи нелинейного программирования.

Авторы стремились изложить основной материал этой части таким образом, чтобы он мог служить руководством к решению задач, подготовке к коллоквиумам, тестам и зачету.

Предложенные в данном учебном пособии контрольные вопросы и задания призваны закрепить теоретические основы и способствовать развитию практических навыков, полученных в результате изучения курса «Численные методы и методы оптимизации».

Необходимыми предпосылками для успешного усвоения дисциплины являются знания, умения, навыки, компетенции, сформированные у студентов при изучении таких курсов, как математика, дискретная математика, дифференциальные уравнения, информатика, вычислительная математика.

При усвоении данной дисциплины у студентов и слушателей курса формируются знания о численных методах отыскания экстремумов функций и их применимости в профессиональной деятельности, а также умения использовать численные методы и методы оптимизации для отыскания точек экстремума функций без ограничений и при их наличии.

Учебное пособие может быть полезным студентам всех направлений, а также аспирантам, инженерам и экономистам, желающим быстро овладеть основами численного решения задач оптимизации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Акулич, И.Л. Математическое программирование в примерах и задачах [Текст]: учеб. пособ. / И.Л. Акулич. – 2-е изд., испр. и доп.– М.: Высшая школа, 1993. – 336 с.

2. Аттетков А.В. Введение в методы оптимизации [Электронный ресурс]: учеб. пособие/ А.В. Аттетков, В.С. Зарубин, А.Н. Канатников.– Электрон. текстовые данные.– М.: Финансы и статистика, 2014.– 272 с.– Режим доступа: http://www.iprbookshop.ru/18794.– ЭБС «IPRbooks», по паролю.

3. Банди, Б. Методы оптимизации. Вводный курс [Текст]: пер. с англ. / Б. Банди. – М.: Радио и связь, 1988. – 128 с.

4. Бахвалов Н.С. Численные методы [Электронный ресурс]/ Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков.– Электрон. текстовые данные.– М.: БИНОМ. Лаборатория знаний, 2012.– 635 с.– Режим доступа: http://www.iprbookshop.ru/6502.– ЭБС «IPRbooks», по паролю.

5. Бахвалов Н.С. Численные методы в задачах и упражнениях [Электронный ресурс]: учеб. пособие/ Н.С. Бахвалов, А.В. Лапин, Е.В. Чижонков.– Электрон. текстовые данные.– М.: БИНОМ. Лаборатория знаний, 2013.– 240 с.– Режим доступа: http://www.iprbookshop.ru/12282.– ЭБС «IPRbooks», по паролю.

6. Васильков, Ю.В. Компьютерные технологии вычислений в математическом моделировании [Текст]: учеб. пособ. / Ю.В Васильков, Н.Н. Василькова – М.: Финансы и статистика, 1999. – 256 с.

7. Данко, П.Е. Высшая математика в упражнениях и задачах [Текст]: учеб. пособ. для вузов. В 2-х ч. Ч. 1. 7-е изд., испр. / П.Е. Данко, А.Г. Попов, Т.Я. Кожевникова, С.П. Данко. – М.: Оникс; Мир и образование, 2008. – 3688 с.

8. Дьяконов В.П. MATLAB. Полный самоучитель [Электронный ресурс]/ Дьяконов В.П.– Электрон. текстовые данные.– М.: ДМК Пресс, 2014.– 768 с.– Режим доступа: http://www.iprbookshop.ru/7911.– ЭБС «IPRbooks», по паролю.

9. Зангвилл, У. Нелинейное программирование. Единый подход [Текст] / У. Зангвилл; под ред. Е.Г. Гольштейна: пер. с англ. – М.: Сов. радио, 1973. – 312 с.

10. Карманов, В.Г. Математическое программирование [Текст]: учеб. пособие / В.Г. Карманов. – М.: Наука. Гл. ред. физ.-мат. лит., 1975. – 272 с.

11. Кондаков Н.С. Основы численных методов [Электронный ресурс]: практикум/ Кондаков Н.С.– Электрон. текстовые данные.– М.: Московский гуманитарный университет, 2014.– 92 с.– Режим доступа: http://www.iprbookshop.ru/39690.– ЭБС «IPRbooks», по паролю.

12. Контрольно-измерительные материалы по курсу «Численные методы и методы оптимизации» [Текст]: учебно-методическое пособие / В.В. Кузина, А.Н. Кошев. – Пенза: ПГУАС, 2013. – 60 с.

13. Кошев, А.Н. Введение в численные методы [Текст]: учеб. пособие / А.Н. Кошев, С.В. Бакушев, И.Г. Гвоздева. – Пенза: ПГАСА, 2000. – 53 с.

14. Кошев, А.Н. Дискретная математика[Текст]: учеб. пособие в 2 ч. Ч. I. Элементы дискретной математики. Ч.II. Элементы теории графов / А.Н. Кошев, В.В. Кузина. – Пенза: ПГАСА, 2002. – 156 с.

15. Кошев, А.Н. Численные методы и методы оптимизации. В 2-х ч. [Текст]: учеб. пособие: / А.Н. Кошев, В.В. Кузина. – Пенза: ПГУАС, 2004. – 136 с.

16. Кошев А.Н. Численные методы решения задач оптимизации [Текст]: учеб. пособие / А.Н. Кошев, В.В. Кузина. – Пенза: ПГУАС, 2012. – 132 с.

17. Кузнецов, А.В. Высшая математика: Математическое программирование [Текст]: учебник / А.В. Кузнецов, В.А. Сакович, Н.И. Холод; под общ. ред. А. В. Кузнецова. – Мн.: Высш. шк., 1994. – 286 с.

18. Курсовые работы по направлению 230200 «Информационные системы» [Текст]: методические указания для студентов специальности «Информационные системы и технологии» по выполнению курсовых работ / А.Н. Кошев, В.В. Кузина. – Пенза: ПГУАС, 2006.– 28 с.

19. Мышкис, А.Д. Элементы теории математических моделей [Текст] / А.Д. Мышкис. –2-е изд., испр. – М.: Едиториал УРСС, 2004. – 192 с.

20. Охорзин, В.А. Прикладная математика в системе MATHCAD [Текст]: учеб. пособие / В.А. Охорзин. – 2-е изд., испр. и доп. – СПб.: Лань, 2008. – 352 с.

21. Пантелеев, А.В. Методы оптимизации в примерах и задачах: учеб. пособ. / А.В. Пантелеев, Т.А. Летова. – М.: Высшая школа, 2002. – 544 с.

22. Плис, А.И. Mathcad. Математический практикум для инженеров и экономистов: учеб. пособ. / А.И. Плис, Н.А. Сливина. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2003. – 656 с.

23. Пшеничный, Б.Н. Численные методы в экстремальных задачах [Текст]/ Б.Н. Пшеничный, Ю.М. Данилин. – М.: Наука. Гл. ред. физ.-мат. лит., 1975. – 320 с.

24. Ромакин, М.И. Математический аппарат оптимизационных задач [Текст]/ М.И. Ромакин. – М.: Статистика, 1975. – 112 с.

25. Самарский, А.А. Введение в численные методы [Текст]: учеб. пособие для вузов / А.А. Самарский. – 2-е изд., перераб. и доп.– М.: Наука. Гл. ред. физ.-мат. лит., 1987. – 287 с.

26. Самарский, А.А. Задачи и упражнения по численным методам [Текст]: учеб. пособие / А.А. Самарский, П.Н. Вабищевич, Е.А. Самарская. – М.: Эдиториал УРСС, 2000. – 208 с.

27. Седов Е.С. Основы работы в системе компьютерной алгебры Mathematica [Электронный ресурс]/ Седов Е.С.– Электрон. текстовые данные.– М.: Интернет-Университет Информационных Технологий

(ИНТУИТ), 2016.– 401 с.– Режим доступа: http://www.iprbookshop.ru/16717.– ЭБС «IPRbooks», по паролю.

28. Супрун, А.Н. Вычислительная математика для инженеров-экологов [Текст]: методическое пособие / А.Н. Супрун, В.В. Найденко. – М.: Изд-во Ассоциации строительных вузов, 1996. – 391 с.

29. Табак, Д. Оптимальное управление и математическое программирование [Текст] / Д. Табак, Б. Куо; под ред. Я.З. Цыпкина: пер. с англ. – М.: Наука. Гл. ред. физ.-мат. Лит., 1975. – 280 с.

30. Тыртышников, Е.Е. Методы численного анализа [Текст]: учеб. пособие для студ. вузов / Е.Е. Тыртышников – М.: ИЦ «Академия», 2007. – 320 с.

31. Химмельблау, Д. Прикладное нелинейное программирование [Текст] / Д. Химмельблау; И.М. Быховской и Б.Т. Вавилова; под ред. М.Л. Быховского: пер. с англ. – М.: Мир, 1975. – 534 с.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. http://www.intuit.ru/
2. http://www.exponenta.ru/
3. www.mathnet.ru – общероссийский математический портал;
4. http://e.lanbook.com/books/?p_f_1_temp_id=18&p_f_1_65=917&p_f_1_63=&p_f_1_67= – электронно-библиотечная система, издательство «Лань»;
5. www.elibrary.ru – научная электронная библиотека;
6. http://lib.mexmat.ru/ – электронная библиотека механико-математического факультета МГУ;
7. http://www.newlibrary.ru/genre/nauka/matematika/kompyutery_i_matematika/ – электронная библиотека по математике;
8. http://www.edu.ru/modules.php?op=modload&name=Web_Links&file=index&l_op=viewlink&cid=2851 – федеральный портал российского профессионального образования: численные методы;

https://mipt.ru/education/chair/computational_mathematics/study/materials/compmath/ – кафедра вычислительной математики МФТИ: вычислительная математика (3 курс).

СПИСОК ИСПОЛЬЗУЕМЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

R^n – n-мерное евклидово арифметическое пространство

$\Omega \subset R^n$ – множество Ω принадлежит R^n, т.е. является его подмножеством

$x \in \Omega$ – x принадлежит множеству Ω, т.е. является его элементом

$|x|$ – длина (модуль), евклидова норма вектора x

$\|A\|$ – евклидова норма матрицы A

A^T – матрица, транспонированная к матрице A

Δz – приращение функции z

$\nabla\Phi(x)$ или $\operatorname{grad}\Phi(x)$ – градиент функции $\Phi(x_1, x_2, \ldots, x_n)$

$\Phi'_{\bar{e}}(x)$ – производная функции Φ по направлению $\bar{e}$

(a, b) – скалярное произведение векторов a и b

$\min\limits_X f(x)$, $\max\limits_X f(x)$ – минимальное и максимальное значения функции $f(x)$ на множестве X

$f(x) \to \min, \quad x \in \Omega$ – задача минимизации функции $f(x)$ на множестве Ω

$H(x) = \nabla^2 f(x) = \left[\dfrac{\partial^2 f}{\partial x_i \partial x_j}\right]$ – матрица Гёссе

МО – методы оптимизации

ЗВП – задача выпуклого программирования

ЗЛП – задача линейного программирования

ЗМП – задача математического программирования

ЗНП – задача нелинейного программирования

ОГЛАВЛЕНИЕ

Учебное издание

Кузина Валентина Владимировна
Кошев Александр Николаевич

ЧИСЛЕННЫЕ МЕТОДЫ И МЕТОДЫ ОПТИМИЗАЦИИ
Учебное пособие
для направления подготовки 09.03.02
«Информационные системы и технологии»

В авторской редакции
Верстка Н.В. Кучина

Подписано в печать 04.05.16. Формат 60х84/16.
Бумага офисная «Снегурочка». Печать на ризографе.
Усл.печ.л. 3,5. Уч.-изд.л. 3,75. Тираж 80 экз.
Заказ № 327.

Издательство ПГУАС.
440028, г. Пенза, ул. Германа Титова, 28.