Н Н МОИСЕЕВ
Ю П. ИВАНИЛОВ
Е. М. СТОЛЯРОВА

МЕТОДЫ ОПТИМИЗАЦИИ

Допущено Министерством
высшего и среднего специального образования СССР
в качестве учебного пособия
для студентов вузов, обучающихся
по специальности «Прикладная математика»

МОСКВА «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
1978

22 193
М 74
УДК 519.6

Методы оптимизации. Моисеев Н. Н., Иванилов Ю. П., Столярова Е. М. «Наука», Главная редакция физико-математической литературы, М., 1978, 352 стр

Настоящая книга предназначена в качестве учебного пособия для студентов факультетов прикладной математики, факультетов по переподготовке специалистов в области использования вычислительной техники, а также для учащихся математических техникумов. В ней излагается методика составления оптимизационных моделей в прикладных задачах, общие принципы линейного, нелинейного и динамического программирования. Приводится обзор основных методов численного анализа для задач отыскания экстремумов функций.

М $\frac{20204-170}{053(02)-78}$ БЗ 30-89—78

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Предлагаемая книга является изложением первой части курса лекций «Методы оптимизации», который читается студентам факультета прикладной математики МФТИ. Этот курс состоит из трех частей: конечномерные задачи оптимизации, вариационное исчисление, включающее в себя теорию оптимального управления, и теория принятия решений.

Математика, как любая другая научная дисциплина, родилась из потребностей общества, и отвечает вполне определенным запросам людей. Конечно, по мере развития техники, усложнения характера производственной деятельности связь математики и практики становилась все более опосредованной. Постепенно возникла логика собственного развития, порождающая задачи, которые стимулировали появление новых идей и методов дисциплины. Тем не менее, практическая деятельность людей и другие науки всегда оказывали решающее влияние на эволюцию математики, на выбор новых направлений, на формирование шкалы ценностей.

Очень важно, что в процессе развития общества возникла потребность в «культуре мышления», и эту нагрузку, в своей значительной части, приняла на себя математика. Она превратилась в определенную школу мышления и анализа. Математика играет еще одну важную роль — она дает основу того языка, который объединяет различные направления науки, облегчает миграцию идей; проникая в различные дисциплины, она постепенно становится их составной частью: определить, например, где в математической физике кончается физика и начинается математика, — невозможно. И это, наверное, по существу.

Могущество математики в ее единстве, в ее целостности. Вот почему, говоря о прикладной математике, следует иметь в виду не какую-либо специальную дисциплину, а вопросы использования математических методов анализа для решения прикладных задач. Термин «прикладная

математика («mathématique appliquee») возник в прошлом веке именно в связи с использованием математики для решения задач небесной механики, статистики, инженерного дела и т. д. Крупнейшими представителями прикладной математики прошлого были Пуанкаре, Чебышев, Ляпунов и др.

Любая дисциплина, относящаяся к «прикладной математике», должна предстать перед изучающим ее как часть математики Читатель должен видеть связь изучаемого материала с математической классикой Это дает ему не только ощущение возможности черпать идеи и методы из того кладезя сокровищ человеческой мысли, который представляет сегодня математика, но и переносить в новые области человеческой деятельности стиль и культуру мышления, созданные в недрах математики.

Курс лекций, первой части которого посвящена эта книга, носит ярко выраженный прикладной характер. Актуальность всего того направления, которое иногда называется теорией оптимизации, возникла лишь в эпоху ЭВМ, ибо без электронных машин методы оптимизации лишены какой-либо перспективы своего практического использования — реализация алгоритмов отыскания экстремумов чересчур трудоемка Но возникла эта дисциплина как глава анализа, как развитие тех классических подходов, которые связаны с именами Ньютона, Эйлера, Лагранжа Конечно, в процессе развития появились и новые методы и новые идеи, но основа все-таки лежит в анализе

Вот почему изложение и начинается с анализа В первой главе мы фактически повторяем материал, известный студентам, но уже на этом этапе должен увидеть то новое, что ждет его впереди. Это новое связано с машиной, с необходимостью довести исследования до числа. Поэтому постепенно мы начинаем обсуждать практическую возможность реализации идей, их превращения в экономные вычислительные алгоритмы

Среди многочисленных идей, возникших за последние десятилетия в связи с проблемами отыскания экстремальных значений функций и функционалов, следует, вероятно, выделить следующие: идеи линейного программирования, принцип максимума Понтрягина и теорию локальных экстремумов. Конечно, они далеко не равноценны по сложности Линейное программирование и связанные с ним идеи пе-

ребора вершин многогранника, т. е. симплекс-метод и теория двойственности, совершенно элементарны и допускают наглядное геометрическое представление. Что же касается двух других вопросов, то их изложение в рамках учебного курса далеко не просто.

Теория локальных экстремумов дает сегодня, наверное, наиболее общий подход к анализу экстремальных задач. Поэтому изложение методов оптимизации без обсуждения идей этой теории не может быть полноценным. Мы избрали некоторый промежуточный путь: рассказывая об этих идеях, в доказательствах ограничились рассмотрением лишь самого простого случая конечного числа измерений. Тем не менее, изложение даже самого простого варианта этой теории дает возможность продемонстрировать единство изучаемого предмета. Так, например, важная сама по себе (и доказанная ранее независимо) теорема Куна — Таккера оказывается простым следствием теоремы Милютина — Дубовицкого о сопряженных конусах.

Принцип максимума Понтрягина (для непрерывных управляемых систем) в данный раздел курса, разумеется, не входит: он относится к вариационному исчислению. Но изложенная здесь теория нам во многом поможет и в изложении следующей части. С помощью теории локальных экстремумов мы рассматриваем управляемые системы дискретного аргумента — вопросы, важные сами по себе. Эта теория дает возможность установить для таких систем необходимые условия типа принципа максимума, но при дополнительном условии выпуклости функции Гамильтона. Этот пункт нам кажется важным не только в прикладном плане (последний очевиден, поскольку теоремы типа принципа максимума являются инструментом декомпозиции многомерных задач и, следовательно, ключом к построению численных схем). Он подготавливает наших читателей к восприятию того удивительного факта, что при переходе от задач конечномерных к задачам континуальным требование выпуклости исчезает. Кроме того, владение техникой работы с конусами нам позволит изложить в следующей части курса вывод условий трансверсальности в задаче Л. С. Понтрягина.

Заканчивается наша книга главой, которая стоит несколько особняком. Наряду с идеями оптимизации, которые своими корнями уходят в классический анализ, понемногу

стали развиваться и идеи совершенно иной природы. Это идеи последовательного анализа вариантов, их отбраковки, последовательного сужения множества, которому должно принадлежать решение. Они восходят еще к Маркову, получили развитие в работах американского математика Вальда и привели в конце концов к динамическому программированию. Обсуждая свою знаменитую программу, Гильберт высказывает надежду, что в XX веке математики овладеют способами решения оптимизационных задач. Это действительно проблема, ибо в вычислительном плане оптимизационные задачи на порядок более трудоемки, чем все те задачи, с которыми до сих пор сталкивались исследователи. Более того, уже сейчас ясно, что классические методы, даже если в нашем распоряжении окажутся гипотетические вычислительные машины предельного быстродействия, позволят решать только относительно простые задачи. Вот почему так важно, чтобы изучающий методы оптимизации видел и иные пути преодоления «проклятия размерности», кроме тех, которые стали традиционными.

Методы оптимизации — курс, ориентированный на решение задач, возникающих в практической деятельности: в экономике, физике, инженерном деле и т. д. Так же, как и математическая физика, наш курс — это мост, ведущий от математики к анализу конкретных задач. Но нам не кажется разумным начинать курс с изложения этих конкретных задач. Такой стиль изложения чреват опасностью потери у изучающего ощущения общности изучаемого с предшествующим математическим материалом. Наконец, разрозненные примеры, когда они предшествуют изложению математической теории, обычно плохо понимаются читателями. Иное дело, когда у изучающего уже накопился солидный инструментарий, тогда у него появляется органическая потребность пустить этот арсенал средств в дело. Вот почему изложение содержательных примеров мы начинаем далеко не сразу, и роль неформального изложения все время растет. Однако основная содержательная нагрузка лежит на последней (третьей) части курса, так как главным потребителем и «поставщиком» оптимизационных задач является исследование операций — дисциплина, занимающаяся проблемами принятия решений (деятельностью, завершающей любое исследование). В человеческой практике очень редко встречаются «чистые» задачи оптимизации. Стремления инжене-

ров, исследователей, экономистов бывает очень трудно свести к одному критерию. Человек, как правило, всегда оказывается в конфликтной ситуации — он стремится к разным целям, ему мешает стремление других субъектов и т. д. Анализ подобных ситуаций, конечно, лежит за пределами математики, но математик должен ясно видеть свое место в решении подобных проблем.

И, наконец, последнее — проблема строгости изложения. Математика не случайно сделалась эталоном мышления. Этим она обязана представлению о строгости, которое вырабатывалось веками и, конечно, как-то все время деформировалось под натиском нового материала и расширения круга своих задач. Поэтому университетская традиция все и вся доказывать на первых курсах абсолютно необходима: студент должен усвоить эти эталоны. Иначе он не станет математиком. Но все имеет свои разумные пределы. Интуиция, опыт — все то, что обычно называется здравым смыслом или неформальным мышлением, — в такой же мере имеют законное право на существование при анализе математических задач, как и все прочее. Основные трудности в доказательствах обычно связаны со стремлением включить в теорию все возможные патологические случаи, чтобы обеспечить достаточную общность. Именно поэтому прозрачные в своей основе исходные идеи постепенно обрастают тяжелыми и трудными подробностями. Однако иногда даже незначительное сужение класса рассматриваемых задач принципиально упрощает доказательство. Так, например, замена предположения об измеримости решений предположением об их кусочной непрерывности делает доказательство принципа максимума совершенно элементарным. Подобный принцип нами всюду проводится. Если угодно, целый ряд доказательств заменяется их «показательствами», и за этот счет изложение качественно упрощается.

Наконец, в связи с последним еще одно замечание. Сегодня большие усилия направлены на разработку численных алгоритмов и их исследование и, в частности, на то, чтобы понять способ оценки алгоритма. И здесь классическим традициям принадлежит основная роль. Поэтому в качестве главного критерия обычно принимают сходимость алгоритма. Расходящийся алгоритм — это плохой алгоритм; чем быстрее сходится алгоритм, тем лучше — подобные истины почти прописные. А в действительности, с точки зрения

вычислителя-практика все выглядит в несколько ином свете. Теоретики численных методов забывают о том, что машинный нуль — это совсем не нуль, а машинная бесконечность — это вовсе не бесконечность, что важную роль играет время счета и удобство обращения к алгоритму, и многое, многое другое Постепенно становится очевидным, что решение действительно больших задач требует неформальных действий вычислителя, возможности вмешиваться в процесс счета — так называемого диалогового режима и т. д. Другими словами, сходимость алгоритма — это лишь один из многих критериев Вот почему вопросы сходимости алгоритмов почти не рассматриваются

Сказанное означает, что в книге принят «физический уровень строгости», т. е. та разумная степень глубины анализа метода, которая необходима для его использования. Этот выбор стиля изложения сразу делает очевидным круг читателей, на который ориентируются авторы Своими читателями мы видим, прежде всего, студентов факультетов прикладной математики и лиц, имеющих инженерное образование, но желающих углубить свои математические знания, необходимые для решения прикладных задач оптимизации

При написании книги мы широко использовали следующие ротапринтные учебные пособия для студентов МФТИ:

1. *Н Н Моисеев.* Методы оптимизации. Глава I. Задача отыскания экстремума функций многих переменных. ВЦ АН СССР, 1968

2. *Н. Н Моисеев.* Методы оптимизации. Глава II. Нелинейное программирование. ВЦ АН СССР, 1969.

3. *Н. Н Моисеев, А Ф. Кононенко.* Методы оптимизации. Главы II, III Нелинейное программирование. Динамическое программирование. МФТИ, 1972.

4. *И. А. Ватель, Ф. И. Ерешко, Ю. П. Иванилов.* Методы решения экстремальных задач, МФТИ, 1977.

5. *И А Ватель, Ю. П. Иванилов.* Математические методы теории управления МФТИ, 1977.

Теоретические результаты, изложенные в книге, находят широкое применение при решении многих оптимизационных задач В последние годы создаются целые комплексы программ, предназначенных для реализации на ЭВМ различных численных методов. Пример одного такого комплек-

са — диалоговой человеко-машинной системы оптимизации — представлен в приложении, написанном Ю Г Евтушенко

При подготовке рукописи к печати большую работу, далеко выходящую за рамки обычной редакторской, провел В Ю Лебедев Благодаря его советам и помощи изложение ряда разделов заметно улучшилось Авторы выражают ему за это глубокую признательность.

Н. Н. Моисеев
Ю П Иванилов
Е. М. Столярова

Глава I

ЗАДАЧА ОТЫСКАНИЯ ЭКСТРЕМУМА ФУНКЦИЙ МНОГИХ ПЕРЕМЕННЫХ

Введение

Одной из важных задач анализа является задача отыскания экстремума (наибольшего или наименьшего значения) скалярной функции $f(x)$ n-мерного векторного аргумента x при некоторых ограничениях. Эту задачу мы будем записывать следующим образом:

$$\min f(x), \tag{0.1}$$

$$x \in X. \tag{0.2}$$

Здесь X — некоторое подмножество n-мерного евклидова пространства E_n Впредь будем называть X *допустимым множеством* задачи (0.1)—(0.2), а точки, принадлежащие X, — ее *допустимыми точками*. Заметим, что задачу максимизации функции $f(x)$ тоже можно записать в виде (0.1) — (0.2), заменив $f(x)$ на $\tilde{f}(x) = -f(x)$.

В этой главе будут последовательно рассмотрены задача *безусловной минимизации* функции одной переменной ($X = E_1$), задача нахождения безусловного экстремума функции нескольких переменных ($X = E_n$) и, наконец, задача на *относительный экстремум*, т. е. задача минимизации функции нескольких переменных при наличии ограничений типа равенств, когда X — множество решений уравнения

$$g(x) = 0,$$

где $g(x)$ есть m-мерная вектор-функция, $m < n$.

Задача (0.1) — (0.2) является классической и рассматривается во всех курсах анализа. Теория решения таких задач развивалась еще в трудах Эйлера, Лагранжа, Бернулли, Лейбница. Она не потеряла своего значения и в настоящее время, несмотря на то, что с тех пор разработаны более общие методы, включающие классические, как частный случай. Классическая теория содержит значительную часть идей, лежащих в основе современных методов оптимизации. Поэтому изложение этих методов мы начнем с известных фактов анализа.

§ 1. Функция одной переменной. Условия экстремума

1. Предварительные рассмотрения. Возьмем задачу минимизации функции одной переменной $f(x)$ на множестве $X \subset E_1$:

$$\min f(x), \tag{1.1}$$

$$x \in X \subset E_1. \tag{1.2}$$

Ее формулировка нуждается в некоторых уточнениях. Поясним это на следующих примерах.

Пример 1.1. Пусть множество X состоит из четырех точек, значения функции $f(x)$ в которых заданы таблицей:

f	0,9	0,4	1,3	0,8
x	1,2	2,5	2,8	4,7

Перебрав эти точки и сравнив между собой значения функции в них, легко убедиться, что минимум достигается в точке $\hat{x} = 2{,}5$ и его величина есть

$$\hat{f} = f(\hat{x}) = 0{,}4.$$

Пример 1.2. Изменим несколько функцию **примера 1.1** и зададим ее следующей таблицей:

f	0,9	0,4	1,3	0,4
x	1,2	2,5	2,8	4,7

Величина минимума в данном примере остается прежней, $\hat{f} = 0{,}4$, но достигается он теперь не в единственной точке, а в двух $\hat{x}' = 2{,}5$ и $\hat{x}'' = 4{,}7$. Поэтому целесообразно говорить о множестве точек минимума.

Приведенные примеры показывают, что необходимо четко сформулировать понятие решения задачи (1.1) — (1.2).

Определение 1.1. Точка $\hat{x}$ доставляет *глобальный минимум* функции $f(x)$ на множестве X, если $\hat{x} \in X$ и

$$f(\hat{x}) \leqslant f(x) \tag{1.3}$$

для всех $x \in X$.

Определение 1.2. Точка $\hat{x}$ называется точкой *строгого глобального минимума* $f(x)$ на множестве X, если $\hat{x} \in X$ и

$$f(\hat{x}) < f(x) \tag{1.4}$$

для всех $x \in X$, $x \neq \hat{x}$.

Под решением задачи (1.1) — (1.2) часто понимают любую точку, удовлетворяющую условию (1.3). Если

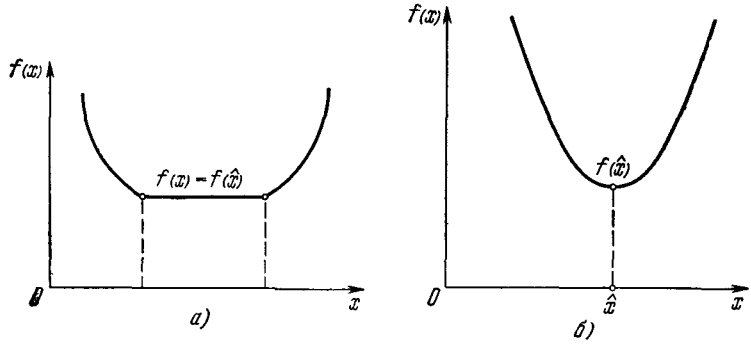


Рис. 1.1.

такая точка единственна, то она и является точкой строгого минимума. Если равенство в формуле (1.3) возможно при $x \neq \hat{x}$, то говорят, что реализуется нестрогий минимум, а под решением в этом случае понимают множество

$$\hat{X} = \{x \in X : f(x) = f(\hat{x})\}.$$

Пример 1.3. График функции, имеющей нестрогий минимум (определение (1.3)), может содержать горизонтальный участок в окрестности точки минимума (рис. 1.1*а*). Для функции со строгим минимумом это исключено (рис. 1.1*б*).

Наряду с задачей определения глобального минимума функции возникает задача поиска локального минимума. Дадим соответствующее определение

Определение 1.3. Точка $\hat{x} \in X$ доставляет *локальный минимум* функции $f(x)$ на множестве X, если при некотором достаточно малом $\varepsilon > 0$ для всех $x \neq \hat{x}$, $x \in X$, удовлетворяющих условию

$$|x - \hat{x}| \leqslant \varepsilon,$$

выполнено неравенство

$$f(\hat{x}) \leqslant f(x). \tag{1.5}$$

Если неравенство (1.5) — строгое, то точку $\hat{x}$ называют точкой *строгого локального минимума* функции $f(x)$. Понятно, что глобальный минимум является и локальным, но не наоборот.

Все определения для максимума функции получаются заменой в выражениях (1.3), (1.4), (1.5) знака неравенства на обратный.

Пример 1.4 Условию (1.5) могут удовлетворять одновременно как точки локального минимума, так и точки локального максимума функции $f(x)$ (рис. 1.2).

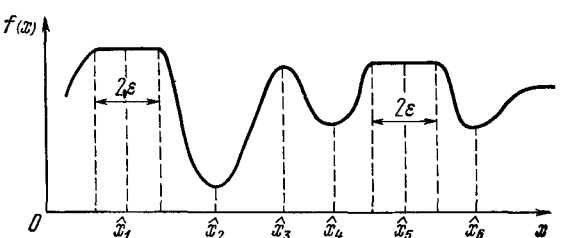


Рис. 1.2

Для функции, изображенной на рис. 1.2, точки $\hat{x}_1$, $\hat{x}_3$, $\hat{x}_5$ являются точками локального максимума, в точках $\hat{x}_4$, $\hat{x}_6$ реализуются локальные минимумы, а точка $\hat{x}_2$ — точка глобального минимума.

Обсудим теперь вопрос о существовании решения задачи (1.1) — (1.2). Рассмотрим некоторые варианты задания множества X. Естественно считать, что это множество непусто и не состоит из единственной точки. Только в этом случае задача минимизации содержательна, так как есть возможность выбора. Если множество X содержит конечное число точек, то решение задачи (1.1) — (1.2)

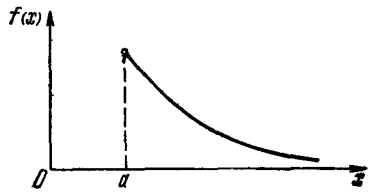


Рис. 1.3.

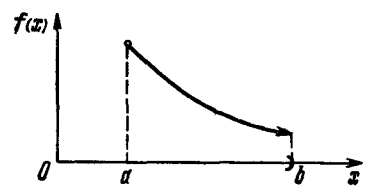


Рис. 1.4.

существует: можно перебрать все точки из X и выбрать среди них точку $\hat{x}$, удовлетворяющую условию (1.3). В случае, когда множество X содержит бесконечное число точек, задача минимизации $f(x)$ на X может не иметь решения.

Пример 1.5. Пусть

$$X = \{x : x \geqslant a\}$$

(рис. 1.3), а функция $f(x)$ монотонно убывает (например, $f(x) = e^{-x}$). Очевидно, что здесь точки $\hat{x}$, удовлетворяющей условию (1.3), не существует.

Пример 1.6. Пусть множество X задано в виде

$$X = \{x \,.\, a \leqslant x < b\},$$

т. е. не замкнуто: точка $x=b$ не содержится во множестве X (на рис. 1.4 это отмечено дужкой в точке b). Если функция $f(x)$ при $x \to b$ монотонно убывает, нижняя грань ее на множестве X не достигается (на рис. 1.4 это показано стрелкой) и точки $\hat{x}$, удовлетворяющей условию (1.3), не существует.

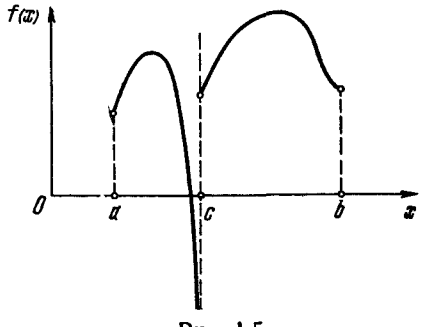


Рис. 1.5

Таким образом, рассмотренные примеры показывают, что в случаях, когда множество X не замкнуто, задача (1.1) — (1.2) может не иметь решения. Разумеется, это не означает, что в подобных случаях решения существовать не может. В самом деле, достаточно взять в примере 1.5 множество $X=\{x : x \leqslant a\}$, а в примере 1.6 $X=\{x : a < x \leqslant b\}$, $f(b)=\lim\limits_{x \to b-} f(x)$, чтобы соответствующие задачи были разрешимы.

Пример 1.7 Пусть

$$X=\{x : a \leqslant x \leqslant b\},$$

а $f(x)$ — неограниченная снизу функция, имеющая вертикальную асимптоту при $x=c$ (рис. 1.5). Здесь не существует точки, удовлетворяющей условию (1.3), т. е. неограниченная снизу на замкнутом ограниченном множестве X функция $f(x)$ глобального минимума на X не имеет.

Сформулируем теперь теорему Вейерштрасса, выделяющую широкий класс задач минимизации (максимизации), заведомо имеющих решение.

Теорема 1.1 *Задача минимизации непрерывной функции $f(x)$ на замкнутом ограниченном множестве X разрешима, т е. непрерывная функция $f(x)$ достигает на замкнутом ограниченном множестве своего минимума (во внутренней или граничной точке)*

Доказательство этой теоремы можно найти в любом курсе анализа

После того как мы сформулировали задачу минимизации функции и обсудили некоторые вопросы, связанные с существованием ее решения, необходимо указать признаки, с помощью которых можно найти точки $\hat{x}$, являющиеся решением задачи (1.1) — (1.2), или проверить, доставляет ли некоторая найденная точка $x \in X$ минимум функции $f(x)$. Эти признаки называются необходимыми условиями и достаточными условиями экстремума.

2. Необходимое условие первого порядка. Выясним условия, которые должны выполняться в точках локального экстремума функции. Рассмотрим те случаи, когда множество X представляет собой вещественную ось. Рассуждения сохраняют силу и для задач, в которых множество X не совпадает с E_1, но открыто, т. е. состоит только из внутренних точек, либо экстремум достигается в его внутренней точке. Изучение случаев, когда экстремум реализуется на границе множества X, требует, как мы увидим ниже, специальных методов.

При выводе условий экстремума будем предполагать, что функция $f(x)$ имеет в окрестности исследуемой точки $\hat{x}$ непрерывные производные до второго порядка включительно.

Теорема 1.2. *Для того чтобы функция $f(x)$, определенная на вещественной оси, имела безусловный локальный экстремум в точке $\hat{x}$, необходимо, чтобы выполнялось условие*

$$\frac{df}{dx}\Big|_{x=\hat{x}} = 0. \tag{1.6}$$

Доказательство. Пусть точка $\hat{x}$ доставляет локальный безусловный минимум функции $f(x)$ (случай максимума рассматривается аналогично). Тогда, согласно определению 1.3, найдется такая окрестность этой точки радиуса ε, что для всех ξ, удовлетворяющих неравенству $|\xi| \leqslant \varepsilon$,

$$f(\hat{x}+\xi) - f(\hat{x}) \geqslant 0. \tag{1.7}$$

По формуле Тейлора имеем

$$f(\hat{x}+\xi) = f(\hat{x}) + \xi f'(\hat{x}) + O(\xi^2).$$

Предположим, что $f'(\hat{x}) \neq 0$, и выберем $\xi = -f'(\hat{x})\rho$, где $\rho > 0$ — любое малое число такое, что $|f'(\hat{x})|\rho < \varepsilon$.

Тогда получим

$$\frac{f(\hat{x}+\xi)-f(\hat{x})}{\rho} = -(f'(\hat{x}))^2 + \frac{O(\rho^2)}{\rho}.$$

Так как

$$\lim_{\rho \to 0} \frac{O(\rho^2)}{\rho} = 0,$$

то найдется такое малое ρ^*, что второе слагаемое в правой части последнего выражения будет по абсолютной величине меньше первого, т. е. $f(\hat{x}+\xi)-f(\hat{x})<0$, что противоречит предположению (1.7). Итак,

$$f'(\hat{x}) = 0,$$

что и требовалось доказать.

Рассмотрим пример функции одной переменной, определенной на всей вещественной оси и удовлетворяющей

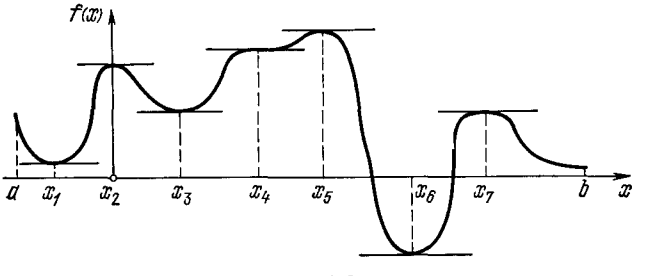


Рис. 1.6.

всем указанным выше условиям (рис. 1.6). Легко видеть, что наименьшего значения функция $f(x)$ достигает в точке x_6, и если, например, известно, что вне интервала $[a, b]$ функция $f(x)$ возрастает, то точка x_6 является точкой ее абсолютного глобального минимума на $X = E_1$. Заметим, что слева от этой точки функция $f(x)$ убывает, а справа от нее — возрастает. В самой же точке x_6 убывание функции приостанавливается. Поэтому она и называется *стационарной*. Вообще, стационарными называются все точки, удовлетворяющие условию (1.6). На рис. 1.6 — это точки, имеющие горизонтальную касательную. Однако, как видно из рис 1.6, не все из стационарных точек будут точками локального минимума (только точки x_1, x_3, x_6) Например,

точки x_2, x_5, x_7 являются точками локального максимума, а точка x_4 — точкой перегиба.

Таким образом, условие (1.6) выделяет стационарные точки, но не определяет их характера: оно одинаково для точек максимума, минимума и перегиба. Чтобы избежать сравнения значений функции $f(x)$ во всех стационарных точках с целью определения глобального минимума (или локальных минимумов), желательно найти некоторые дополнительные условия, которые выполняются только в точках минимума.

3. Необходимые условия второго порядка. Пусть функция $f(x)$ удовлетворяет перечисленным выше условиям. Тогда справедлива

Теорема 1.3. *Для того чтобы функция $f(x)$ имела в стационарной точке $\hat{x}$ безусловный локальный минимум (максимум), необходимо, чтобы ее вторая производная была неотрицательна (неположительна), т. е.*

$$\left.\frac{d^2f}{dx^2}\right|_{x=\hat{x}} \geqslant 0 \qquad \left(\left.\frac{d^2f}{dx^2}\right|_{x=\hat{x}} \leqslant 0\right). \tag{1.8}$$

Доказательство. По теореме 1.2 первая производная в стационарной точке равна нулю. Соответственно, из формулы Тейлора при всех ξ получим

$$f(\hat{x}+\xi)-f(\hat{x})=\frac{1}{2}\xi^2 f''(\hat{x})+o(\xi^2). \tag{1.9}$$

Допустим теперь, что наша теорема неверна, т. е.

$$\left.\frac{d^2f}{dx^2}\right|_{x=\hat{x}} < 0. \tag{1.10}$$

Тогда для достаточно малых ξ второе слагаемое в правой части выражения (1.9) будет по абсолютной величине меньше первого и, следовательно, выполнится неравенство

$$f(\hat{x}+\xi)-f(\hat{x})<0.$$

Это противоречит определению точки $\hat{x}$ как точки локального минимума. Значит,

$$\left.\frac{d^2f}{dx^2}\right|_{x=\hat{x}} \geqslant 0.$$

Теорема доказана.

Условия (1.6), (1.8) называются необходимыми условиями минимума (максимума) второго порядка. Нетрудно

показать, что совокупность условий (1.6), (1.8) не является достаточным условием минимума.

Пример 1.8. Пусть $f(x)=x^3$. Тогда точка $\hat{x}=0$ удовлетворяет необходимым условиям минимума второго порядка:

$$\left.\frac{df}{dx}\right|_{x=0} = 3x^2\,{}_{x=0} = 0,$$
$$\left.\frac{d^2f}{dx^2}\right|_{x=0} = 6x\,{}_{x=0} = 0.$$

При этом она является точкой перегиба, а не минимума функции $f(x)=x^3$.

4. Достаточные условия. Если функция $f(x)$ дифференцируема достаточное число раз, то можно построить аналогичные (1.6), (1.8) необходимые условия любого порядка. Налагая все более и более жесткие ограничения на выбор экстремальных точек, они, тем не менее, не дают окончательного ответа об их характере. Поэтому нужно иметь еще и достаточные условия экстремума.

Теорема 1.4. *Для того чтобы функция $f(x)$ имела в стационарной точке $\hat{x}$ безусловный локальный минимум (максимум), достаточно, чтобы ее вторая производная была в $\hat{x}$ положительна (отрицательна):*

$$\left.\frac{d^2f}{dx^2}\right|_{x=\hat{x}} > 0 \qquad \left(\left.\frac{d^2f}{dx^2}\right|_{x=\hat{x}} < 0\right). \tag{1.11}$$

Доказательство. Поскольку точка $\hat{x}$ — стационарная, разложение функции $f(x)$ в ряд Тейлора в окрестности $\hat{x}$ имеет вид

$$f(\hat{x}+\xi)-f(\hat{x}) = \frac{1}{2}\xi^2 f''(\hat{x}) + o(\xi^2).$$

При $f''(\hat{x}) \neq 0$ для достаточно малых ξ знак правой части этого выражения определяется знаком второй производной $f''(\hat{x})$, т. е. из (1.11) вытекает неравенство

$$f(\hat{x}+\xi)-f(\hat{x}) > 0.$$

Последнее означает, что $\hat{x}$ — точка строгого локального минимума функции $f(x)$, что и требовалось доказать.

Обратимся теперь к случаю, когда в стационарной точке вторая производная функции $f(x)$ обращается в нуль, т. е.

$$\frac{df}{dx}(\hat{x}) = 0 \quad \text{и} \quad \frac{d^2f}{dx^2}(\hat{x}) = 0.$$

Как видно из примера 1.8, полученные нами условия экстремума не определяют в этом случае характера стационарной точки. Очевидно, необходимо рассмотреть производные более высоких порядков и снова воспользоваться разложением функции $f(x)$ в ряд Тейлора.

Пусть функция $f(x)$ имеет в окрестности точки $\hat{x}$ непрерывные производные до k-го порядка включительно, и пусть

$$f'(\hat{x}) = f''(\hat{x}) = \ldots = f^{(k-1)}(\hat{x}) = 0, \quad f^{(k)}(\hat{x}) \neq 0.$$

Тогда, согласно формуле Тейлора,

$$f(\hat{x}+\xi) - f(\hat{x}) = \frac{1}{k!}\xi^k f^{(k)}(\hat{x}) + o(\xi^k).$$

Анализируя это выражение, придем к следующей теореме (достаточному условию общего вида).

Теорема 1.5. *Пусть функция $f(x)$, определенная на множестве $X = E_1$, имеет непрерывные производные до k-го порядка включительно, причем в некоторой точке $\hat{x}$*

$$f'(\hat{x}) = f''(\hat{x}) = \ldots = f^{(k-1)}(\hat{x}) = 0, \quad f^{(k)}(\hat{x}) \neq 0.$$

Тогда, если k — четное число, то функция $f(x)$ имеет в точке $\hat{x}$ локальный максимум при $f^{(k)}(\hat{x}) < 0$ и локальный минимум при $f^{(k)}(\hat{x}) > 0$. Если k нечетно, то $f(x)$ не имеет в точке $\hat{x}$ ни максимума, ни минимума.

Заметим, что необходимые условия экстремума — это уравнения относительно неизвестных величин $\hat{x}$. Корни этих уравнений определяют некоторое множество «претендентов» на экстремум — значений переменной x, среди которых только и могут находиться интересующие нас точки $\hat{x}$, доставляющие максимум или минимум функции $f(x)$. Для того чтобы среди этих точек разыскать, например, точки минимума, мы должны еще в каждой точке множества «претендентов» проверить выполнение достаточных условий.

§ 2. Функция многих переменных

1. Необходимое условие экстремума. Снова рассмотрим задачу безусловной минимизации, но будем теперь считать, что $f(x)$ — скалярная функция векторного аргумента размерности n, т. е. $X = E_n$. Если $\hat{x}$ — точка ее безуслов-

ного локального экстремума, в $\hat{x}^j$ будет достигаться экстремум функции

$$f(\hat{x}^1, \hat{x}^2, \ldots, \hat{x}^{j-1}, x^j, \hat{x}^{j+1}, \ldots, \hat{x}^n)$$

одной переменной x^j, которая получается из функции $f(x)$, если зафиксировать все переменные, кроме x^j, положив $x^i = \hat{x}^i$ для $i \neq j$. Для функции же одной переменной

$$f(\hat{x}^1, \ldots, \hat{x}^{j-1}, x^j, \hat{x}^{j+1}, \ldots, \hat{x}^n)$$

получена теорема 1.2. Проведя это рассуждение для всех $j = 1, \ldots, n$, приходим к следующей теореме.

Теорема 2.1. *Для того чтобы в точке $\hat{x}$ функция $f(x^1, \ldots, x^n)$ имела безусловный локальный экстремум, необходимо, чтобы все ее частные производные обращались в $\hat{x}$ в нуль*:

$$\frac{\partial f}{\partial x^i}\Big|_{x=\hat{x}} = 0, \quad i = 1, 2, \ldots, n. \tag{2.1}$$

Условие стационарности (2.1) мы будем записывать еще в одной из следующих эквивалентных форм:

$$\operatorname{grad} f(\hat{x}) = \nabla f(\hat{x}) = 0, \tag{2.1'}$$

$$f'(\hat{x}) = 0, \tag{2.1''}$$

где $\nabla f(\hat{x}) = f'(\hat{x})$ — n-мерный вектор с компонентами $\frac{\partial f}{\partial x^i}(\hat{x})$, $i = 1, \ldots, n$, который принято называть градиентом функции $f(x)$ в точке $\hat{x}$.

Заметим, что необходимое условие экстремума (2.1) эквивалентно равенству нулю дифференциала функции $f(x)$ в точке $\hat{x}$:

$$df(\hat{x}) = 0.$$

В самом деле, если выполнено условие (2.1), то для любых dx^i, $i = 1, \ldots, n$, имеем

$$df(\hat{x}) = \sum_{i=1}^{n} \frac{\partial f}{\partial x^i}(\hat{x})\, dx^i = 0.$$

Справедливо и обратное утверждение, так как из последнего равенства в силу произвольности независимых приращений dx^i, $i = 1, \ldots, n$, следует, что все частные

производные в точке $\hat{x}$ равны нулю:

$$\frac{\partial f}{\partial x^i}(\hat{x}) = 0, \quad i = 1, \ldots, n.$$

Условия (2.1) образуют систему n уравнений для определения n компонент вектора $\hat{x}$. Эти уравнения могут иметь различную природу и допускать любое количество решений, в частности, не иметь ни одного. Как и выше, точки $\hat{x}$, являющиеся решениями системы уравнений (2.1), будем называть стационарными, а условие (2.1) — необходимым условием экстремума первого порядка.

2. Необходимое условие второго порядка. Достаточные условия. После того как решение $\hat{x}$ системы уравнений (2.1) будет найдено, необходимо еще определить характер стационарной точки $\hat{x}$. Для этого нужно исследовать поведение функции $f(x)$ в окрестности стационарной точки $\hat{x}$. Снова воспользуемся разложением функции $f(x)$ в ряд Тейлора, предполагая ее дважды непрерывно дифференцируемой по всем переменным $x^1, \ldots, x^n$. Тогда получим

$$f(\hat{x}+\xi) = f(\hat{x}) + \frac{1}{2}\sum_{k,i=1}^{n} \frac{\partial^2 f}{\partial x^i \partial x^k}(\hat{x})\,\xi^i\xi^k + o(\|\xi\|^2). \quad (2.2)$$

Здесь через $\frac{\partial^2 f}{\partial x^i \partial x^k}(\hat{x})$ мы обозначили элементы матрицы вторых производных функции $f(x)$ в стационарной точке $\hat{x}$, а через $\|\xi\|$ — какую-нибудь норму вектора ξ, например, $\|\xi\| = \sqrt{(\xi, \xi)}$. Далее матрицу вторых производных мы будем обозначать так:

$$f''(\hat{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^1 \partial x^1}(\hat{x}) & \ldots & \frac{\partial^2 f}{\partial x^1 \partial x^n}(\hat{x}) \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \frac{\partial^2 f}{\partial x^n \partial x^1}(\hat{x}) & \ldots & \frac{\partial^2 f}{\partial x^n \partial x^n}(\hat{x}) \end{bmatrix}. \quad (2.3)$$

Характер стационарной точки $\hat{x}$ функции $f(x)$ связан со знакоопределенностью квадратичной формы

$$\sum_{i,k=1}^{n} \frac{\partial^2 f}{\partial x^i \partial x^k}(\hat{x})\,\xi^i\xi^k = (\xi, f''(\hat{x})\,\xi). \quad (2.4)$$

Напомним, что квадратичная форма называется неотрицательно определенной в точке $\hat{x}$, если

$$(\xi,\ f''(\hat{x})\,\xi) \geqslant 0, \tag{2.5}$$

и положительно определенной, если

$$(\xi,\ f''(\hat{x})\,\xi) > 0 \tag{2.6}$$

для любых векторов $\xi \neq 0$.

Соответственно, симметричная матрица вторых производных $f''(\hat{x})$ называется неотрицательно определенной в точке $\hat{x}$, если выполнено (2.5), и положительно определенной, если выполнено (2.6). Неположительно определенным и отрицательно определенным квадратичным формам и матрицам соответствуют противоположные знаки в неравенствах (2.5), (2.6).

Таким образом, с учетом разложения (2.2), приходим к следующей формулировке условий второго порядка экстремальности функции $f(x^1, \ldots, x^n)$.

Теорема 2.2. *Для того чтобы дважды непрерывно дифференцируемая функция n переменных $f(x)$ имела в стационарной точке $\hat{x}$ безусловный локальный минимум (максимум), необходимо, чтобы матрица ее вторых производных была неотрицательно (неположительно) определенной, и достаточно, чтобы она была положительно (отрицательно) определенной.*

Проверка знакоопределенности матриц может быть осуществлена, например, с помощью критерия Сильвестра. Согласно этому критерию, необходимым и достаточным условием положительной определенности квадратичной формы (x, Ax), где $A = \{a_{ij}\}$ — симметричная $n \times n$ матрица, является выполнение n неравенств:

$$a_{11} > 0, \quad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0, \quad \ldots, \quad \begin{vmatrix} a_{11} & \ldots & a_{1n} \\ \ldots & \ldots & \ldots \\ a_{n1} & \ldots & a_{nn} \end{vmatrix} > 0.$$

Необходимым и достаточным условием отрицательной определенности квадратичной формы (x, Ax) является выполнение цепочки следующих n неравенств:

$$(-1)^n a_{11} > 0, \quad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0, \ \ldots$$

$$\ldots, \ (-1)^n \begin{vmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{vmatrix} > 0.$$

Если квадратичная форма не меняет знака, но обращается в нуль при ненулевых значениях аргумента, то для определения характера стационарной точки $\hat{x}$ требуется исследование производных более высокого порядка.

3. Пример. Проиллюстрируем содержание настоящего параграфа на следующей задаче: определить экстремальные значения функции

$$f(x) = \frac{(x^1)^2}{a} + \frac{(x^2)^2}{b},$$

$$a \neq 0, \quad b \neq 0, \quad X = E_2.$$

Из необходимых условий (2.1) имеем

$$\frac{\partial f}{\partial x^1} = \frac{2x^1}{a} = 0, \quad \frac{\partial f}{\partial x^2} = \frac{2x^2}{b} = 0.$$

Поэтому $\hat{x}^1 = 0$, $\hat{x}^2 = 0$ — стационарная точка. Коэффициенты квадратичной формы (2.4), вычисленные в ней, равны

$$\frac{\partial^2 f}{(\partial x^1)^2} = \frac{2}{a}, \quad \frac{\partial^2 f}{\partial x^1 \partial x^2} = \frac{\partial^2 f}{\partial x^2 \partial x^1} = 0, \quad \frac{\partial^2 f}{(\partial x^2)^2} = \frac{2}{b}.$$

Тогда, согласно теореме 2.2, имеем следующие случаи:

1) $a > 0$, $b > 0$ — функция $f(x)$ имеет в точке $\hat{x} = \{0, 0\}^T$ *) минимум;

2) $a > 0$, $b < 0$ }
3) $a < 0$, $b > 0$ } — экстремума нет;

4) $a < 0$, $b < 0$ — функция $f(x)$ имеет в точке $\hat{x} = \{0, 0\}^T$ максимум.

Отметим, что случаи 1) и 4) соответствуют поверхности, являющейся эллиптическим параболоидом, а случаи 2) и 3) — гиперболическому параболоиду, имеющему стационарную точку типа «седло».

§ 3. Относительный экстремум. Метод множителей Лагранжа

1. Метод исключения. Рассмотрим теперь задачу на относительный экстремум. Как мы видели в § 2, решение задачи об отыскании экстремумов функции n переменных $f(x)$ на всем пространстве E_n может быть сведено с помощью необходимых условий к решению системы урав-

*) Здесь и далее $\{x^1, \ldots, x^n\}^T$ — вектор-столбец.

нений (2.1), в результате чего определяются стационарные точки функции $f(x)$. Оказывается, что аналогичное сведение возможно и для задачи отыскания экстремумов функции $f(x)$ при наличии ограничений типа равенств

$$g_i(x) = 0, \quad i = 1, 2, \ldots, m. \tag{3.1}$$

Условия (3.1) принято еще называть уравнениями связи.

Уточним, что именно мы будем понимать под решением задачи на относительный экстремум. Напомним (см. введение), что точку x, удовлетворяющую условиям (3.1), мы договорились назвать допустимой.

Определение 3.1. Допустимая точка $\hat{x}$ доставляет *относительный локальный минимум* функции $f(x)$, если можно указать такое число $\varepsilon > 0$, что для всех x, удовлетворяющих уравнениям связи (3.1) и условию $\|x - \hat{x}\| < \varepsilon$, имеет место неравенство

$$f(x) \geqslant f(\hat{x}).$$

Определения строгого относительного локального минимума, а также все определения для относительного максимума получаются по аналогии с приведенными в § 1.

Рассмотрим случай, когда уравнения связи (3.1) могут быть разрешены относительно части переменных. Будем предполагать, что функции $g_i(x)$, $i = 1, \ldots, m$, имеют в окрестности рассматриваемой допустимой точки $\hat{x}$ непрерывные частные производные по всем аргументам до второго порядка включительно и, кроме того, ранг матрицы Якоби для функций $g_i(x)$, $i = 1, \ldots, m$, рассматриваемой в точке $\hat{x}$, равен m. Не нарушая общности, предположим, что отличен от нуля определитель (якобиан), составленный из частных производных по первым m аргументам, т. е.

$$\begin{vmatrix} \frac{\partial g_1}{\partial x^1} & \cdots & \frac{\partial g_1}{\partial x^m} \\ \cdots\cdots\cdots & & \\ \frac{\partial g_m}{\partial x^1} & \cdots & \frac{\partial g_m}{\partial x^m} \end{vmatrix} \neq 0. \tag{3.2}$$

Тогда по теореме о неявных функциях в некоторой окрестности точки $\hat{x}$ система уравнений (3.1) разрешима относительно $x^1, \ldots, x^m$, т. е. представима в виде

$$x^j = \varphi_j(x^{m+1}, \ldots, x^n), \quad j = 1, 2, \ldots, m, \tag{3.3}$$

где $\varphi_j(x^{m+1}, \ldots, x^n)$ — непрерывно дифференцируемые в рас-

сматриваемой окрестности функции. Переменные $x^{m+1}, \ldots, x^n$ естественно назвать «независимыми», в отличие от «зависимых» — $x^1, \ldots, x^m$. Подставляя выражения (3.3) в функцию $f(x)$, получим задачу отыскания безусловного экстремума функции $n-m$ переменных

$$f(\varphi_1(x^{m+1}, \ldots, x^n), \ldots, \varphi_m(x^{m+1}, \ldots, x^n), x^{m+1}, \ldots, x^n) = \\ = \tilde{f}(x^{m+1}, \ldots, x^n).$$

Однако провести исключение части компонент вектора x обычно бывает трудно или даже невозможно. Поэтому мы используем другой путь определения точки $\hat{x}$, который не предполагает наличия явных выражений типа (3.3), хотя использует существенно условие (3.2).

2. Метод множителей Лагранжа. Как мы видели в замечании к теореме 2.1, в точке $\hat{x}$, доставляющей безусловный экстремум функции, ее полный дифференциал равен нулю, т. е.

$$df(\hat{x}) = \sum_{j=1}^{m} \frac{\partial f}{\partial x^j}(\hat{x})\, dx^j + \sum_{k=m+1}^{n} \frac{\partial f}{\partial x^k}(\hat{x})\, dx^k = 0, \quad (3.4)$$

где dx^j, $j=1, \ldots, m$, — дифференциалы «зависимых» переменных, связанные с дифференциалами «независимых» переменных dx^k, $k=m+1, \ldots, n$, следующим образом:

$$\sum_{j=1}^{m} \frac{\partial g_i}{\partial x^j}(\hat{x})\, dx^j + \sum_{k=m+1}^{n} \frac{\partial g_i}{\partial x^k}(\hat{x})\, dx^k = 0, \quad i=1, \ldots, m. \quad (3.5)$$

Уравнения (3.5) получены при дифференцировании полным образом уравнений связи (3.1). Исключим теперь дифференциалы «зависимых» переменных из уравнений (3.4), (3.5). Для этого умножим каждое из уравнений системы (3.5) на произвольные множители $\lambda_1, \ldots, \lambda_m$ и результаты сложим с уравнением (3.4), тогда получим следующее равенство:

$$\sum_{j=1}^{m} \left(\frac{\partial f}{\partial x^j}(\hat{x}) + \lambda_1 \frac{\partial g_1}{\partial x^j}(\hat{x}) + \ldots + \lambda_m \frac{\partial g_m}{\partial x^j}(\hat{x}) \right) dx^j + \\ + \sum_{k=m+1}^{n} \left(\frac{\partial f}{\partial x^k}(\hat{x}) + \lambda_1 \frac{\partial g_1}{\partial x^k}(\hat{x}) + \ldots + \lambda_m \frac{\partial g_m}{\partial x^k}(\hat{x}) \right) dx^k = 0. \quad (3.6)$$

Распорядимся множителями $\lambda_1, \ldots, \lambda_m$ таким образом, чтобы обратились в нуль коэффициенты при дифференциалах «зависимых» переменных, т. е.

$$\frac{\partial f}{\partial x^j}(\hat{x}) + \lambda_1 \frac{\partial g_1}{\partial x^j}(\hat{x}) + \ldots + \lambda_m \frac{\partial g_m}{\partial x^j}(\hat{x}) = 0, \quad j = 1, \ldots, m. \tag{3.7}$$

Это можно сделать, так как уравнения (3.7) являются системой линейных алгебраических уравнений относительно множителей $\lambda_1, \ldots, \lambda_m$, которая имеет единственное решение в силу того, что ее определитель (3.2) по условию отличен от нуля. При выбранных таким образом значениях множителей в равенстве (3.6) останутся только члены, содержащие дифференциалы «независимых» переменных. Поэтому коэффициенты при этих дифференциалах должны быть нулями, т. е.

$$\frac{\partial f}{\partial x^k}(\hat{x}) + \lambda_1 \frac{\partial g_1}{\partial x^k}(\hat{x}) + \ldots + \lambda_m \frac{\partial g_m}{\partial x^k}(\hat{x}) = 0, \quad k = m+1, \ldots, n. \tag{3.8}$$

Таким образом, мы получили систему $n+m$ уравнений (3.1), (3.7), (3.8) относительно $n+m$ неизвестных $\hat{x}^1, \ldots, \hat{x}^n, \hat{\lambda}_1, \ldots, \hat{\lambda}_m$. Этот результат представляет собой основное содержание *метода множителей Лагранжа* и позволяет определить множество «претендентов» на решение в задаче на относительный экстремум. Метод Лагранжа состоит из следующих этапов:

1) составляется функция $n+m$ переменных, которая называется функцией Лагранжа:

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x); \tag{3.9}$$

2) вычисляются и приравниваются нулю ее частные производные по x и λ:

$$\begin{aligned} &\frac{\partial L}{\partial x^j} = \frac{\partial f}{\partial x^j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x^j} = 0, \quad j = 1, \ldots, n, \\ &\frac{\partial L}{\partial \lambda_i} = g_i(x) = 0, \qquad\qquad i = 1, \ldots, m; \end{aligned} \tag{3.10}$$

3) решается система (3.10) $n+m$ уравнений относительно $n+m$ неизвестных $x^1, \ldots, x^n, \lambda_1, \ldots, \lambda_m$.

Система уравнений (3.10) представляет собой необходимые условия первого порядка в задаче на относительный экстремум, а ее решения $\hat{x}^1, \ldots, \hat{x}^n$ принято называть *условно-стационарными точками*. Как и в случае задач на безусловный экстремум, необходимые условия первого порядка не определяют характера условно-стационарной точки. Для выяснения этого вопроса следует привлечь производные более высоких порядков функций $f(x)$ и $g(x)$.

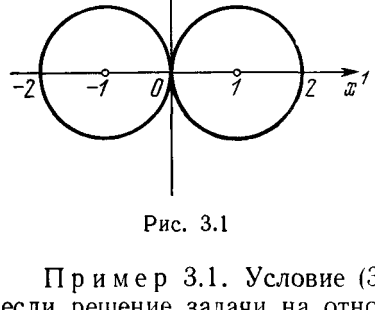


Рис. 3.1

Заметим, что требование неравенства нулю якобиана (3.2) является существенным. Только в этом случае система уравнений (3.7) разрешима, причем единственным образом, относительно множителей Лагранжа $\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_m$.

Пример 3.1. Условие (3.2) может быть не выполнено, если решение задачи на относительный экстремум реализуется, например, в точке касания поверхностей ограничений (3 1) (начало координат на рис. 3.1). Пусть

$$n=2, \quad m=2, \quad f(x)=x^2,$$
$$g_1(x)=(x^1-1)^2+(x^2)^2-1,$$
$$g_2(x)=(x^1+1)^2+(x^2)^2-1.$$

Допустимая точка должна одновременно удовлетворять уравнениям $g_1(x)=0$, $g_2(x)=0$ и является единственной: $x^1=0$, $x^2=0$. Очевидно, что точка $\hat{x}^1=0$, $\hat{x}^2=0$ и будет решением задачи на относительный минимум функции $f(x)=x^2$ при ограничениях $g_1(x)=g_2(x)=0$. Составим для этой задачи функцию Лагранжа:

$$L(x, \lambda)=f(x)+\sum_{i=1}^{2}\lambda_i g_i(x)=$$
$$=x^2+\lambda_1[(x^1-1)^2+(x^2)^2-1]+\lambda_2[(x^1+1)^2+(x^2)^2-1].$$

Метод множителей Лагранжа приводит к уравнениям

$$\frac{\partial L}{\partial x^1} = 2\lambda_1 (x^1 - 1) + 2\lambda_2 (x^1 + 1) = 0,$$

$$\frac{\partial L}{\partial x^2} = 1 + 2\lambda_1 x^2 + 2\lambda_2 x^2 = 0.$$

Этим уравнениям точка относительного минимума $\hat{x}^1 = 0$, $\hat{x}^2 = 0$ не удовлетворяет ни при каких значениях λ_1, λ_2, т. е. в данном случае метод множителей Лагранжа не работает.

Заметим попутно, что метод множителей Лагранжа можно применять всегда для функции Лагранжа более общего вида

$$L(x, \bar{\lambda}) = \lambda_0 f(x) + (\lambda, g(x)), \tag{3.11}$$

причем примеру 3.1 соответствует система множителей Лагранжа $\hat{\lambda}_0 = 0$, $\bar{\lambda}_1 \neq 0$, $\bar{\lambda}_2 \neq 0$. В случае же, когда выполнено условие (3.2), мы получаем единственную систему множителей Лагранжа с $\hat{\lambda}_0 \neq 0$. Поэтому все множители можно разделить на $\lambda_0 \neq 0$ и пользоваться функцией Лагранжа в виде (3.9).

Приведем теперь простую геометрическую интерпретацию метода множителей Лагранжа. На рис. 3.2 изображены линии уровня функции двух переменных $f(x^1, x^2)$ и ограничение $g(x^1, x^2) = 0$ $(c_k > c_{k-1} > \ldots > c_1)$.

Очевидно, что относительные локальные минимумы функции $f(x)$ при ограничении $g(x) = 0$ могут реализоваться только в точках, где линии уровня функции $f(x)$, т. е. кривые, имеющие уравнения $f(x) = \text{const}$, касаются кривой $g(x) = 0$, например, в точке $\hat{x} = \{\hat{x}^1, \hat{x}^2\}^T$. В самом деле, из других точек, в которых линии уровня $f(x)$ не касаются кривой $g(x = 0)$, а пересекают ее, можно, двигаясь вдоль кривой $g(x) = 0$, уменьшить значение функции $f(x)$. Таким образом, в точках локального относительного минимума градиенты функций $f(x)$ и $g(x)$ направлены по одной прямой, т. е.

$$f'(\hat{x}) = -\lambda g'(\hat{x}). \tag{3.12}$$

Уравнения (3.12), в совокупности с уравнением $g(x) = 0$, как нетрудно видеть, совпадают для рассматриваемого простого случая с необходимыми условиями экстремума в виде (3.10).

3. Достаточные условия относительного экстремума. Как в задачах на безусловный экстремум, рассмотренных в §§ 1, 2, так и в задачах с ограничениями типа равенств, которым посвящен настоящий параграф, необходимые условия экстремума не определяют характера стационарной точки. Вопрос о наличии в ней относительного экстремума

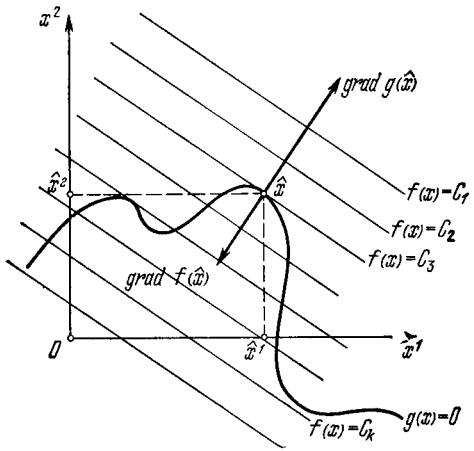


Рис. 3.2

и о выяснении его типа может быть решен с помощью разложений функций $f(x)$, $g_i(x)$ в ряд Тейлора

Пусть пара $\{\hat{x}, \hat{\lambda}\}$ – решение уравнений (3.10) и якобиан (3.2) не равен нулю. Попробуем понять, чем определяются соотношения между значениями функции $f(x)$ в точке $\hat{x}$ и в близких к ней допустимых точках вида $\hat{x}+\xi$. В отличие от случаев, рассмотренных в §§ 1, 2, сравниваемые с $\hat{x}$ точки $\hat{x}+\xi$ должны теперь удовлетворять уравнениям связи (3.1). Заменим при этом приращение функции $f(x)$ приращением функции Лагранжа (3.9) с множителями $\hat{\lambda}_i$, $i=1, 2, \ldots, m$. Тогда получим

$$f(\hat{x}+\xi)-f(\hat{x})=L(\hat{x}+\xi,\ \hat{\lambda})-L(\hat{x},\ \hat{\lambda})=$$
$$=\sum_{i=1}^{n}\frac{\partial L}{\partial x^i}(\hat{x},\ \hat{\lambda})\xi^i+\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial^2 L}{\partial x^i \partial x^j}(\hat{x},\ \hat{\lambda})\xi^i\xi^j+$$
$$+o(\|\xi\|^2), \qquad (3.13)$$

причем первое слагаемое в правой части равно нулю, т. е.

$$f(\hat{x}+\xi)-f(\hat{x})=$$

$$=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial^2 L}{\partial x^i \partial x^j}(\hat{x},\ \hat{\lambda})\xi^i\xi^j+o(\|\xi\|^2). \qquad (3.14)$$

Так как анализируемые смещения ξ из точки $\hat{x}$ не должны нарушать условий связи (3.1), разложение функции $g(x)$ в ряд Тейлора в окрестности $\hat{x}$ приводит к равенству

$$\sum_{i=1}^{n}\frac{\partial g_k}{\partial x^i}(\hat{x})\xi^i+O(\|\xi\|^2)=0,\quad k=1,\ 2,\ \ldots,\ m.$$

Отсюда, пренебрегая вторым слагаемым, в линейном приближении имеем

$$\sum_{i=1}^{n}\frac{\partial g_k(\hat{x})}{\partial x^i}\xi^i=0,\quad k=1,\ 2,\ \ldots,\ m. \qquad (3.15)$$

Это уравнение при каждом k определяет касательную в точке $\hat{x}$ гиперплоскость к поверхности ограничения $g_k(x)=0$, а необходимое условие экстремума второго порядка в задаче на относительный экстремум и достаточное условие однозначно связаны со знакоопределенностью квадратичной формы

$$\sum_{i,\,j=1}^{n}\frac{\partial^2 L}{\partial x^i \partial x^j}(\hat{x},\ \hat{\lambda})\xi^i\xi^j$$

для векторов ξ, удовлетворяющих равенствам (3.15).

Поскольку, по предположению, якобиан (3.2) отличен от нуля, из уравнения (3.15) можно выразить «зависимые» переменные ξ^j, $j=1,\ \ldots,\ m$, через «независимые» ξ^i, $i=m+1,\ \ldots,\ n$. Подставляя соответствующие выражения в формулу (3.14), получим квадратичную форму относительно «независимых» приращений $\xi^{m+1},\ \ldots,\ \xi^n$. По аналогии с теоремой 2.2 заключаем, что, для того чтобы условно-стационарная точка $\hat{x}$ реализовала локальный относительный минимум, необходимо, чтобы эта квадратичная форма была неотрицательно определена, и достаточно, чтобы она была положительно определенной

(соответственно, неположительно определенной и отрицательно определенной для максимума).

4. Пример. Пусть

$$n=2, \quad m=1, \quad f(x)=x^2 \Rightarrow \min,$$
$$g(x)=x^2-(x^1)^2.$$

Функция Лагранжа для этой задачи имеет вид

$$L(x, \lambda)=x^2+\lambda(x^2-(x^1)^2).$$

Соответственно, правило множителей Лагранжа приводит к уравнениям

$$\frac{\partial L}{\partial x^1}=-2\lambda x^1=0, \quad \frac{\partial L}{\partial x^2}=1+\lambda=0, \quad \frac{\partial L}{\partial \lambda}=x^2-(x^1)^2=0,$$

решением которых будет $\hat{\lambda}=-1$, $\hat{x}^1=0$, $\hat{x}^2=0$. Чтобы понять, доставляет точка $\hat{x}=0$ относительный минимум функции $f(x)$ или нет, надо выяснить характер поведения квадратичной формы

$$\sum_{j=1}^{2}\sum_{i=1}^{2}\frac{\partial^2 L}{\partial x^i \partial x^j}(\hat{x}, \hat{\lambda})\xi^i\xi^j=2(\xi^1)^2$$

на прямой

$$\frac{\partial g}{\partial x^1}(\hat{x})\xi^1+\frac{\partial g}{\partial x^2}(\hat{x})\xi^2=-2\hat{x}^1\xi^1+\xi^2=\xi^2=0.$$

При $\xi^2=0$, как функция одной переменной ξ^1, эта форма положительно определена. Значит, в точке $\hat{x}=0$ имеем относительный минимум.

5. Седловая точка функции Лагранжа. Рассмотрим функцию двух переменных $z=\Phi(x, y)$, где x, y — скаляры или векторы.

Определение 3.2 Назовем пару $\{x^*, y^*\}$ *седловой точкой* функции $\Phi(x, y)$, если для любых x, y справедливо неравенство

$$\Phi(x, y^*)\leqslant\Phi(x^*, y^*)\leqslant\Phi(x^*, y). \tag{3.16}$$

Очевидно, что неравенство (3.16) эквивалентно выражению

$$\Phi(x^*, y^*)=\inf_y \Phi(x^*, y)=\sup_x \Phi(x, y^*)=$$
$$=\max_x \inf_y \Phi(x, y)=\min_y \sup_x \Phi(x, y).$$

Снова рассмотрим задачу отыскания относительного экстремума функции $f(x)$ при ограничениях $g(x)=0$. Необходимые условия экстремума (3.10) можно записать в виде

$$\frac{\partial L(\hat{x}, \hat{\lambda})}{\partial x}=0, \quad \frac{\partial L(\hat{x}, \hat{\lambda})}{\partial \lambda}=0, \tag{3.17}$$

т. е. пара $\{\hat{x}, \hat{\lambda}\}$ является стационарной точкой функции Лагранжа

$$L(x, \lambda)=f(x)+(\lambda, g(x)).$$

Однако в этой точке функция $L(x, \lambda)$ не может достигать максимума или минимума по x и λ одновременно. В самом деле, пусть в точке $\{\hat{x}, \hat{\lambda}\}$ достигается максимум функции $L(x, \lambda)$ по x и λ. Так как условия связи в точке $\{\hat{x}, \hat{\lambda}\}$ выполнены, то $L(\hat{x}, \hat{\lambda})=f(\hat{x})$. Пусть, далее, в некоторой точке $\tilde{x}$ нарушено одно из ограничений, например $g_k(\tilde{x}) \neq 0$. Тогда в силу линейности функции L по λ мы можем за счет выбора λ_k добиться бесконечно большого значения L (число λ_k имеет знак, противоположный знаку $g_k(\tilde{x})$). Следовательно, в точке $\{\hat{x}, \hat{\lambda}\}$ функция Лагранжа не может иметь максимума по λ. Аналогично можно показать, что в точке $\{\hat{x}, \hat{\lambda}\}$ не может одновременно достигаться минимум функции Лагранжа по x и λ.

Покажем теперь, что в точке $\{\hat{x}, \hat{\lambda}\}$ достигается либо $\max\limits_{x} \inf\limits_{\lambda} L(x, \lambda)$, либо $\min\limits_{x} \sup\limits_{\lambda} L(x, \lambda)$ в зависимости от того, является $\hat{x}$ точкой максимума или минимума. В самом деле, при каждом фиксированном x

$$\inf_{\lambda} L(x, \lambda)=\begin{cases} -\infty, & \text{если не выполнено хотя бы одно из ограничений,} \\ f(x), & \text{если все } g_i(x)=0,\ i=1, \ldots, m. \end{cases}$$

Следовательно,

$$\max_{x} \inf_{\lambda} L(x, \lambda)=\max_{x \in X} f(x),$$

где

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

Таким образом, по x и λ функция Лагранжа имеет экстремум противоположного характера. Если при этом

оказывается, что

$$\max_x \inf_\lambda L = \min_\lambda \sup_x L,$$

то точка $\{\hat{x}, \lambda\}$, по определению, является седловой точкой функции Лагранжа. В четвертой главе мы вернемся к этому вопросу и убедимся, что при известных условиях этот факт действительно имеет место, что позволяет развивать эффективные численные методы ее отыскания.

6. Заключение. Подведем некоторые итоги проведенного анализа экстремальных задач. Мы сделали большой шаг в понимании природы задач оптимизации — вывели необходимые и достаточные условия, которым должна удовлетворять функция в экстремальных точках. При этом мы рассмотрели те случаи, когда выбор значений аргумента x либо вообще не стеснен никакими ограничениями, либо подчиняется ограничениям типа равенств. Как уже неоднократно подчеркивалось, полученные необходимые условия экстремума не только позволяют проверить, доставляет ли некоторая точка максимум или минимум изучаемой функции, но служат, кроме того, инструментом для эффективного отыскания экстремумов.

В самом деле, условие стационарности

$$f'(\hat{x}) = 0 \tag{3.18}$$

сводит задачу отыскания экстремумов функции $f(x)$ к задаче отыскания корней трансцендентного уравнения (3.18). Если это уравнение имеет конечное число корней, то остается только найти их и проверить с помощью достаточных условий, какой из этих корней является максимумом или минимумом. Конечно, трансцендентное уравнение (3.18) может оказаться сложным. Поэтому реализация такого пути решения задачи далеко не всегда проста. Тем не менее, изложенный подход открывает определенную перспективу для решения разнообразных практических оптимизационных задач.

Прикладные задачи оптимизации, как правило, имеют одну особенность, которая качественно осложняет процедуру нахождения экстремальных значений. Помимо ограничений типа равенств эти задачи могут содержать условия типа неравенств

$$h(x) \leqslant 0. \tag{3.19}$$

Например, мы собираемся приобретать некоторое количество оборудования (вектор x) для того, чтобы увеличить выпуск продукции $f(x)$, т. е. мы должны максимизировать функцию $f(x)$. При этом, если учесть тот реальный факт, что средства на покупку оборудования ограничены, то мы придем к задаче максимизации функции $f(x)$ при ограничениях вида (3.19).

В технике мы также непрерывно сталкиваемся с ограничениями подобного рода. Параметры любой конструкции, которыми мы можем распоряжаться — будь то угол поворота руля самолета или величина тяги ракеты,— всегда ограничены, т. е. подчинены условиям типа неравенств. Другими словами, мы все время сталкиваемся с задачами, в которых требуется отыскать экстремум функции в некоторой ограниченной замкнутой области, и при этом нет никаких оснований считать, что экстремум достигается во внутренней точке этой области. Могут ли классические подходы, которые были изложены в настоящей главе, являться основой для построения эффективных численных процедур в этих, более сложных ситуациях?

Последующее изложение покажет, что более или менее сложные задачи отыскания экстремума при наличии ограничений типа равенств и неравенств требуют специальных подходов, и методы, которые будут получены для решения таких задач, окажутся очень мало похожими на те способы решения, которые мы рассматривали в этой главе. Тем не менее, даже тот пока еще достаточно скудный арсенал средств, которым мы располагаем, опираясь только на классические методы, позволяет в некоторых простых случаях довести решения экстремальной задачи до конца.

Рассмотрим, например, задачу определения максимума функции $f(x)$ скалярного аргумента на отрезке $a \leqslant x \leqslant b$ (рис. 3.3). Используя необходимое условия экстремума

$$f'(x) = 0, \tag{3.20}$$

мы находим все корни этого уравнения, лежащие внутри отрезка $[a, b]$. В данном примере таких корней будет два: $x = c$ и $x = d$. Проверяя достаточные условия, мы убеждаемся, что локальный максимум будет в точке $x = c$. При выводе необходимых и достаточных условий экстремума мы использовали некоторую малую окрестность экстремальной точки $\hat{x}$. Поэтому внутри отрезка $[a, b]$ эти усло-

вия сохраняются, а на концах отрезка они, вообще говоря, не верны. С другой стороны, вычисляя значения функции на концах отрезка (в точках a и b) и сравнивая их со значением локального максимума $f(c)$, мы обнаруживаем, что максимальное значение функции $f(x)$ на отрезке $[a, b]$ достигается как раз в точке $x=b$, т. е.

$$\max\{f(a),\ f(c),\ f(b)\}=f(b).$$

Таким образом, граничные точки множества X, в которых не выполняются необходимые условия (3.20), нужно исследовать отдельно.

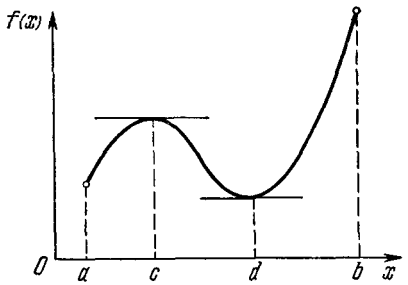


Рис. 3.3

Этот подход нетрудно распространить и на случай большего числа измерений.

Пусть теперь x — вектор, а X — некоторое замкнутое множество, граница которого удовлетворяет уравнению

$$g(x)=0.$$

Тогда схема решения остается прежней. Сначала мы находим все корни уравнения

$$f'(x)=0,$$

лежащие внутри рассматриваемой области. Пусть это будут точки $x_1, x_2, \ldots, x_k$. Затем решаем следующую задачу на относительный экстремум:

$$\max f(x),$$
$$g(x)=0.$$

Для этого составляем функцию Лагранжа

$$L(x, \lambda)=f(x)+(\lambda, g(x))$$

и, применяя метод множителей, выписываем систему уравнений

$$f'(x)+(\lambda,\ g'(x))=0,$$
$$g(x)=0.$$

Пусть корнями этой системы уравнений являются точки $x_{k+1}, \ldots, x_s$. Проверяем в точках $x_1, \ldots, x_k$ достаточные условия экстремума и отбрасываем те точки, в которых нужные нам условия не удовлетворяются. И, наконец, сравнивая значения функции в оставшихся точках со значениями функции в точках $x_{k+1}, \ldots, x_s$, находим максимальное значение функции $f(x)$ на множестве X.

Глава II

ЧИСЛЕННЫЕ МЕТОДЫ ОТЫСКАНИЯ БЕЗУСЛОВНОГО ЭКСТРЕМУМА

Введение

В этой главе мы будем изучать численные методы решения задачи поиска безусловного минимума функции $f(x)$, заданной на всем пространстве E_n. Эта задача представляет не только самостоятельный интерес. Многие алгоритмы решения задач с ограничениями включают минимизацию без ограничений как некоторый этап, процедуру.

Принято различать два подхода к решению задачи об отыскании экстремума. С одним из них мы уже познакомились в первой главе. Этот подход заключается в замене задачи на экстремум задачей поиска решений системы трансцендентных уравнений вида

$$f'(x) = 0 \tag{0.1}$$

в случае, когда речь идет о безусловном экстремуме, или задачей

$$\begin{aligned} \frac{\partial L}{\partial x} &\equiv f'(x) + (\lambda,\ g'(x)) = 0, \\ \frac{\partial L}{\partial \lambda} &\equiv g(x) = 0, \end{aligned} \tag{0.2}$$

если есть ограничения типа равенств. Однако для поиска экстремума можно и не использовать необходимые условия (0.1) или (0.2). В самом деле, если мы найдем способ пошагового определения точек x_1, x_2, x_3 и т. д., значения целевой функции в которых образуют убывающую сходящуюся последовательность, можно надеяться, что тем самым удается найти минимум функции, не прибегая к необходимым условиям. Такие способы обычно называют прямыми методами решения задач оптимизации.

Примечание. Как мы увидим ниже, использование необходимых условий часто приводит к построению некоторой последовательности $\{x_k\}$. Поэтому различие между

двумя указанными подходами оказывается не очень существенным.

Итак, нам нужно научиться строить последовательность векторов $x_0, x_1, \ldots, x_n$, удовлетворяющих условию

$$f(x_0) > f(x_1) > \ldots > f(x_n). \tag{0.3}$$

Такие последовательности $\{x_k\}$ будем называть *релаксационными*, а методы их построения принято называть *методами спуска*.

Различные методы спуска отличаются друг от друга способами выбора направления спуска и длины шага вдоль этого направления. В этих методах точки последовательности $\{x_k\}$ вычисляются по формуле

$$x_{k+1} = x_k + \alpha_k p_k, \tag{0.4}$$

где p_k — направление спуска, а α_k — длина шага вдоль этого направления.

Важнейшей характеристикой методов спуска является их скорость сходимости. При оценке качества метода говорят о линейной скорости сходимости (или о сходимости со скоростью геометрической прогрессии), если

$$\| x_{k+1} - x_* \| \leqslant q \| x_k - x_* \|,$$

где x_* — точка минимума функции $f(x)$, а $0 < q < 1$ — некоторая константа. Скорость сходимости сверхлинейна, если

$$\| x_{k+1} - x_* \| \leqslant q_k \| x_k - x_* \|,$$

где $q_k \to 0$ при $k \to \infty$, и квадратична, если

$$\| x_{k+1} - x_* \| \leqslant C \| x_k - x_* \|^2, \quad C \geqslant 0.$$

Алгоритмы безусловной минимизации принято делить на классы, в зависимости от максимального порядка производных минимизируемой функции, вычисление которых предполагается. Так, методы, использующие только значения самой целевой функции, относят к методам нулевого порядка (иногда их называют также *методами поиска*); если, кроме того, требуется вычисление первых производных минимизируемой функции, то мы имеем дело с методами первого порядка и т. д.

Среди всех наиболее употребительных методов методы второго порядка требуют для получения результата

с заданной точностью наименьшего числа шагов (итераций). Однако это не означает, что они являются наиболее эффективными для минимизации любых функций, если под эффективностью понимать необходимое количество машинных операций (т. е. в конечном счете необходимые затраты машинного времени). Вычисление вторых производных для достаточно сложной функции часто представляет собой очень громоздкую и дорогостоящую с точки зрения затрат машинного времени процедуру. Поэтому на практике метод, который сходится медленнее, но не требует большого количества промежуточных вычислений, может оказаться предпочтительнее. Все зависит от природы функции $f(x)$. Выделить заранее какой-либо метод спуска, пригодный в любом случае, невозможно. Для отыскания наиболее приемлемого метода обычно используют опыт, интуицию и предварительное исследование задачи.

Настоящую главу мы начнем с методов первого порядка, которые обычно называются градиентными.

§ 1. Градиентные методы

1. Общая схема градиентного спуска. Как известно из курсов анализа, градиент скалярной функции $f(x)$ в некоторой точке x_k направлен в сторону наискорейшего возрастания функции и ортогонален линии уровня (поверхности постоянного значения функции $f(x)$, проходящей через точку x_k). Вектор, противоположный градиенту $f'(x_k)$, антиградиент, направлен в сторону наискорейшего убывания функции $f(x)$. Выбирая в качестве направления спуска p_k в (0.4) антиградиент функции $f(x)$ в точке x_k, мы приходим к итерационному процессу вида

$$x_{k+1} = x_k - \alpha_k f'(x_k), \quad \alpha_k > 0, \quad k = 1, 2, \ldots \qquad (1.1)$$

В координатной форме этот процесс записывается следующим образом:

$$x^i_{k+1} = x^i_k - \alpha_k \frac{\partial f}{\partial x^i}(x_k), \quad i = 1, 2, \ldots, n. \qquad (1.2)$$

Все итерационные процессы, в которых направление движения на каждом шаге совпадает с антиградиентом (градиентом) функции, называются *градиентными методами* и отличаются друг от друга способами выбора шага α_k.

Существует много различных способов выбора α_k, но наиболее распространены два: первый называется методом с дроблением шага и связан с проверкой на каждой итерации некоторого неравенства (см. ниже неравенство (1.4)); во втором при переходе из точки x_k в точку x_{k+1} функция $f(x_k - \alpha f'(x_k))$ минимизируется по α — *метод наискорейшего спуска*. Соответствующие итерационные процессы рассматриваются в пп. 2 и 3 настоящего параграфа.

2. Градиентные методы с дроблением шага. Методы с постоянным шагом. Рассмотрим процесс (1.1). Первая проблема, с которой мы сталкиваемся при его реализации, — это выбор шага α_k. Достаточно малый шаг α_k обеспечит убывание функции, т. е. выполнение неравенства

$$f(x_k - \alpha_k f'(x_k)) < f(x_k), \tag{1.3}$$

но может привести к неприемлемо большому количеству итераций, необходимых для достижения точки минимума. С другой стороны, слишком большой шаг может вызвать неожиданный рост функции (невыполнение условия (1.3)) либо привести к колебаниям около точки минимума. Проиллюстрируем эти обстоятельства на простом примере.

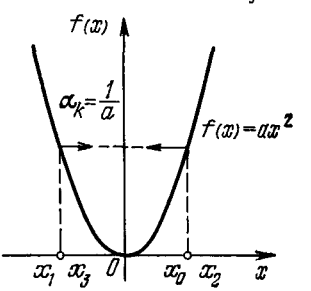


Рис. 1.1.

Пример 1.1. Рассмотрим задачу минимизации функции $f(x) = ax^2$, где a — некоторое положительное число (рис. 1.1). Тогда формула (1.1) принимает вид

$$x_{k+1} = (1 - 2\alpha_k a)\, x_k.$$

Очевидно, что при постоянном шаге α_k соответствующий процесс будет сходиться, если $0 < \alpha_k < \frac{1}{a}$, и расходиться для $\alpha_k > \frac{1}{a}$. Если принять $\alpha_k = \frac{1}{a}$, то $x_1 = -x_0$, $x_2 = x_0$, $x_3 = -x_0$ и т. д. Процесс будет расходящимся, но при этом значения аргумента (а, значит, и функции) повторяются. Расходимость такого рода обычно называют зацикливанием.

В методе градиентного спуска с дроблением шага величина α_k выбирается так, чтобы было выполнено следующее неравенство:

$$f(x_k - \alpha_k f'(x_k)) - f(x_k) \leqslant -\varepsilon\alpha_k \|f'(x_k)\|^2, \qquad (1.4)$$

где $0 < \varepsilon < 1$ — произвольно выбранная постоянная (одна и та же для всех итераций). Очевидно, что требование (1.4) на выбор шага более жесткое, чем условие (1.3), но имеет тот же смысл: функция должна убывать от итерации к итерации. Процесс (1.1) с выбором шага, удовлетворяющего неравенству (1.4), протекает следующим образом.

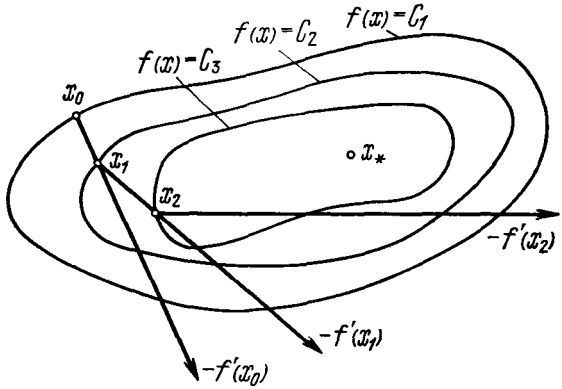


Рис. 1.2.

Выбираем число $\alpha > 0$, одно и то же для всех итераций. На k-й итерации проверяем выполнение неравенства (1.4) при $\alpha_k = \alpha$. Если оно выполнено, полагаем $\alpha_k = \alpha$ и переходим к следующей итерации. Если нет, то шаг α_k дробим до тех пор, пока оно не выполнится.

Геометрически градиентный спуск с дроблением шага изображен на рис. 1.2 Здесь изображены линии уровня функции $f(x)$, имеющей минимум в точке x_*, причем $c_1 > c_2 > c_3 \ldots$, и некоторая зигзагообразная траектория $x_0x_1 \ldots x_k$, ортогональная в каждой точке x_0, x_1, ..., x_k соответствующим линиям уровня и приводящая из начальной точки x_0 в точку минимума x_*. Ломаная $x_0x_1 \ldots x_k$ аппроксимирует так называемую градиентную кривую,

которая удовлетворяет дифференциальному уравнению

$$\frac{dx}{d\alpha} = -f'(x). \tag{1.5}$$

Это уравнение, в свою очередь, получается из дискретного соотношения (1.1) при $\alpha \to 0$, а сама итерационная схема (1.1) может рассматриваться как схема метода Эйлера для решения дифференциального уравнения (1.5).

Заметим, что процедура проверки неравенства (1.4) на каждой итерации является довольно трудоемкой. Если известны некоторые параметры, характеризующие функцию $f(x)$, можно использовать вариант метода (1.1) с постоянным на всех итерациях шагом, при котором функция $f(x)$ заведомо монотонно убывает. Пусть, например, существует константа R такая, что неравенство

$$\|f'(x) - f'(y)\| \leqslant R\|x - y\| \tag{1.6}$$

выполнено для любых $x, y \in E_n$ *). Тогда достаточно взять

$$\alpha_k \equiv \alpha = \frac{1-\varepsilon}{R}. \tag{1.7}$$

Если известна равномерная по x оценка M сверху максимального собственного числа матрицы $f''(x)$, выполнение неравенства (1.4) обеспечивается выбором шага по формуле

$$\alpha_k \equiv \alpha = \frac{2(1-\varepsilon)}{M}. \tag{1.8}$$

При спуске с постоянным шагом трудоемкость каждой итерации минимальна (нужно вычислять только градиент $f'(x_k)$). Однако значения постоянных R, M обычно заранее неизвестны. Утверждения (1.7), (1.8) доказаны в [9].

3. Метод наискорейшего спуска. Как мы видели в п. 2 настоящего параграфа, можно выбрать некоторую постоянную для всех итераций величину шага, обеспечивающую убывание функции $f(x)$ от итерации к итерации. Однако обычно шаг при этом оказывается очень малым, что приводит к необходимости проводить большое количество итераций для достижения точки минимума. Поэтому методы спуска с переменным шагом являются более экономными. Процесс, на каждой итерации которого шаг α_k

*) Сформулированное условие называется условием Липшица, а постоянная R — постоянной Липшица.

выбирается из условия минимума функции $f(x)$ в направлении движения, т. е.

$$f(x_k - \alpha_k f'(x_k)) = \min_{\alpha \geqslant 0} f(x_k - \alpha f'(x_k)), \qquad (1.9)$$

называется методом наискорейшего спуска. В этом варианте градиентного спуска на каждой итерации требуется решать задачу одномерной минимизации (1.9). Разумеется, этот способ выбора α_k сложнее, чем рассмотренные в предыдущем пункте.

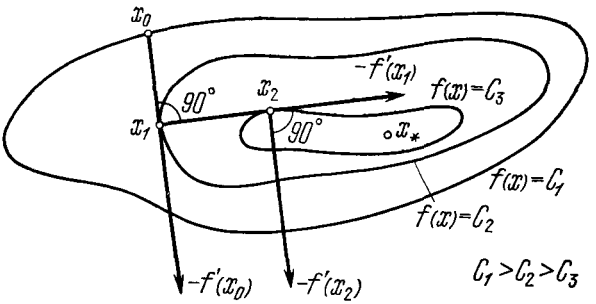


Рис. 1.3.

Геометрическая интерпретация метода наискорейшего спуска представлена на рис. 1.3.

В этом методе, в отличие от обычного градиентного спуска, направление движения из точки x_k касается линии уровня в точке x_{k+1}. Последовательность точек x_0, x_1, x_2, ..., x_k, ... зигзагообразно приближается к точке минимума x_*, причем звенья этого зигзага ортогональны между собой. В самом деле, шаг α выбирается из условия минимизации по α функции

$$\varphi(\alpha) = f(x_k - \alpha f'(x_k)),$$

и поэтому

$$\frac{d\varphi(\alpha_k)}{d\alpha} = -f'(x_{k+1}) f'(x_k) = 0.$$

Таким образом, направления спуска на двух последовательных итерациях взаимно ортогональны (рис. 1.4).

Реализация метода наискорейшего спуска предполагает решение на каждом шаге довольно трудоемкой вспомога-

тельной задачи одномерной минимизации (1.9). Как правило, метод наискорейшего спуска, тем не менее, дает выигрыш в числе машинных операций, поскольку обеспечивает движение с самым выгодным шагом. В то же время

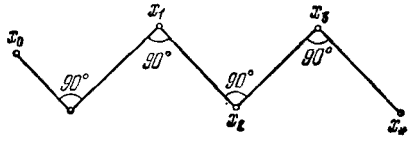


Рис. 1.4.

решение задачи (1.9) связано с дополнительными вычислениями только самой функции $f(x)$, тогда как основное машинное время тратится на вычисление ее градиента $f'(x)$.

4. О сходимости градиентных методов. Во всех рассмотренных выше градиентных методах последовательность точек $\{x_k\}$ сходится к стационарной точке функции $f(x)$ при достаточно общих предложениях относительно свойств этой функции. В частности, справедлива

Теорема 1.1. *Если функция $f(x)$ ограничена снизу, ее градиент удовлетворяет условию Липшица* (1.6) *и выбор значения α_k производится одним из описанных выше способов, то, какова бы ни была начальная точка x_0:*

$$\|f'(x_k)\| \to 0 \quad при \quad k \to \infty.$$

Доказательство этой теоремы можно найти, например, в [9].

При практической реализации схемы (1.1) итерации прекращаются, если для всех i, $i = 1, 2, \ldots, n$, выполнены условия типа

$$\left\| \frac{\partial f}{\partial x^i}(x_k) \right\| \leqslant \delta,$$

где δ — некоторое заданное число, характеризующее точность нахождения минимума.

В условиях теоремы 1.1 градиентный метод обеспечивает сходимость по функции либо к точной нижней грани $\inf\limits_x f(x)$ (если функция $f(x)$ не имеет минимума; рис. 1.5), либо к значению функции в некоторой стационарной точке, являющейся пределом последовательности $\{x_k\}$. Нетрудно придумать примеры, когда в этой точке реализуется

седло, а не минимум. Однако подобные примеры патологичны и не должны смущать читателя. На практике методы градиентного спуска уверенно обходят седловые точки и находят минимумы целевой функции (в общем случае — локальные).

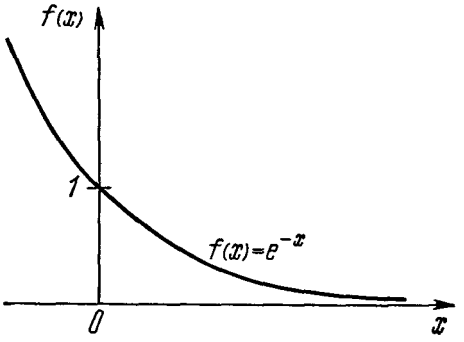


Рис. 1.5.

В предположениях теоремы 1.1 оценить скорость сходимости градиентного спуска не представляется возможным. Мы сделаем это в случае, когда $f(x)$ — сильно выпуклая функция. Напомним, что *сильно выпуклой* называется дважды непрерывно дифференцируемая функция, матрица вторых производных которой при любых $x,\ y \in E_n$ удовлетворяет условию

$$m\|y\|^2 \leqslant (f''(x)\,y,\ y) \leqslant M\|y\|^2, \tag{1.10}$$

где $M \geqslant m > 0$ — некоторые числа.

Теорема 1.2. *Пусть $f(x)$ — сильно выпуклая функция, а последовательность $\{x_k\}$ строится по методу* (1.1) *с выбором шага по схеме* (1.9). *Тогда последовательность $\{x_k\}$ сходится к точке минимума со скоростью геометрической прогрессии со знаменателем $q = \dfrac{M-m}{m+M}$, т. е. при достаточно больших k выполнено неравенство*

$$\|x_{k+1} - x_*\| \leqslant \frac{M-m}{M+m}\|x_k - x_*\|. \tag{1.11}$$

Доказательство можно найти в уже цитированной книге [9].

5. Эффект оврагов. Релаксационные методы. Мы установили, что в предположениях теоремы 1.2 градиентные методы сходятся со скоростью геометрической прогрессии со знаменателем q, зависящим от M и m — равномерных по x оценок сверху и снизу соответственно максимального и минимального собственных чисел матрицы $f''(x)$. (В действительности, в качестве M и m можно взять максимальное и минимальное собственные числа матрицы $f''(x_*)$.) Если M и m мало отличаются друг от друга — матрица $f''(x)$ хорошо обусловлена, — то число q мало и, следовательно, сходимость методов достаточно высокая. Если же $\frac{m}{M} \ll 1$, то q близко к единице, и градиентные методы начинают сходиться плохо. Этот факт хорошо интерпретируется геометрически и известен в литературе как «эффект оврагов». Если числа M и m сильно отличаются, то топография поверхностей уровня $f(x) = \text{const}$ имеет овражную структуру (в задачах максимизации $f(x)$ имеются, соответственно, крутые хребты).

Для иллюстрации сказанного рассмотрим пример.

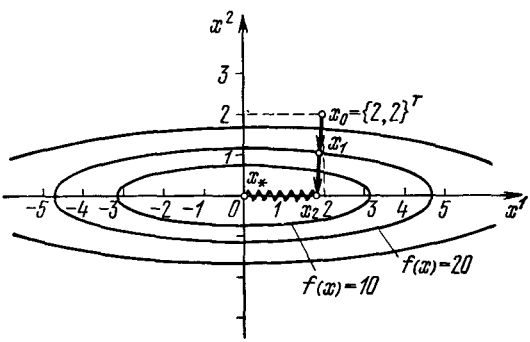


Рис. 1.6.

Пример 1.2. Возьмем функцию

$$f(x) = (x^1)^2 + 16\,(x^2)^2$$

(рис. 1.6). Ее линии уровня сильно вытянуты вдоль оси x^1. Матрица вторых производных в данном случае постоянна и равна

$$f''(x) = H = \begin{bmatrix} 2 & 0 \\ 0 & 32 \end{bmatrix}.$$

Наименьшее и наибольшее собственные числа ее равны, соответственно, 2 и 32, т. е. сильно различаются между собой. Траектория градиентного метода, как видно из рис. 1.6, характеризуется довольно быстрым спуском на «дно» оврага и затем медленным зигзагообразным движением в точку минимума.

Одним из выходов в создавшейся ситуации является изменение масштабов независимых переменных целевой функции. Поясним этот способ на следующем примере.

Пусть функция $f(x)$ имеет вид

$$f(x) = \sum_{i=1}^{n} \alpha_i (x^i)^2, \tag{1.12}$$

где величины $\alpha_i > 0$ сильно различаются между собой. Поверхности уровней функции (1.12) вытянуты вдоль тех осей x^i, которым соответствуют малые α_i. Заменой переменных

$$x^i = \mu_i y^i$$

можно добиться того, чтобы в новых переменных y^i линии уровня стали сферами. Для этого достаточно принять

$$\mu_i = \alpha_i^{-1/2}.$$

Тогда получим преобразование

$$x^i = \alpha_i^{-1/2} y^i. \tag{1.13}$$

Вернемся к нашему примеру 1.2. Мы начали движение из точки $x_0 = \{2, 2\}^T$, и так как составляющие градиента в этой точке

$$\frac{\partial f}{\partial x^1}(x_0) = 4, \quad \frac{\partial f}{\partial x^2}(x_0) = 64$$

сильно различаются, мы получили направление спуска, существенно отклоняющееся от направления в точку минимума $x_* = \{0, 0\}^T$. Замена переменных (1.13) в данном случае имеет вид

$$y^1 = x^1, \quad y^2 = 4x^2.$$

Минимизируемая функция в новых координатах выглядит так:

$$f(y) = (y^1)^2 + (y^2)^2.$$

Вектор градиента в точке $y_0^1 = 2$, $y_0^2 = 4x_0^2 = 8$ действительно направлен в точку минимума, а линии уровня стали окружностями (рис. 1.7).

В случае, когда $f(x)$ не квадратичная, а достаточно гладкая функция общего вида, выбирают

$$\mu_i = \left(\frac{\partial^2 f}{\partial x^i \partial x^i} (x) \right)^{-1/2} \qquad (1.14)$$

в точке x одномерного минимума вдоль направления x^i. Множитель μ_i пропорционален радиусу кривизны линии уровня в точке x. Преобразование (1.14), конечно, не превратит поверхностей уровня в сферы, но в некоторых случаях уменьшит их вытянутость. Гарантированно же исправить топографию функции $f(x)$ можно, если учесть все, а не только диагональные (формулы (1.13), (1.14)) элементы матрицы вторых производных $f''(x)$, и применить преобразование координат вида

$$y = (f''(x))^{1/2} x.$$

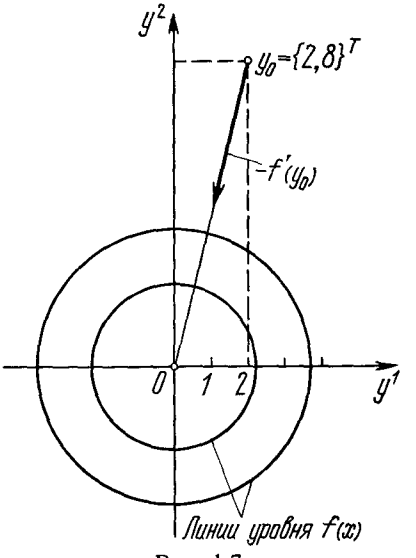


Рис. 1.7.

Здесь под $(f''(x))^{1/2}$ нужно понимать симметричную матрицу, при возведении которой в квадрат получается матрица $f''(x)$ (при этом $f''(x)$ должна быть положительно определена). Нетрудно показать, что указанное преобразование приводит к методу Ньютона (см. § 2 настоящей главы). Иногда функция $f(x)$ слишком сложна, чтобы ее продифференцировать аналитически, тогда вторые производные в (1.14) аппроксимируются их конечно-разностными аналогами.

Масштабирование переменных в общем случае приводит к итерационному процессу вида

$$x_{k+1} = x_k - \alpha_k B_k f'(x_k), \qquad (1.15)$$

где матрица B_k зависит от номера итерации. В примере 1.2 эта матрица постоянна:

$$B = \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1/16 \end{bmatrix}.$$

Методы вида (1.15) часто называют релаксационными. При $B_k = E$ имеем обычный градиентный метод (1.1).

6. Эвристические схемы. Иногда, используя градиентный спуск для минимизации функций со сложной топографической структурой, применяют некоторые эвристические схемы, которые идейно близки к релаксационному методу (1.15). Мы рассмотрим две такие процедуры. Первая из них заключается в следующем.

а) Пусть в точке x_k вычислены все частные производные $\partial f/\partial x^i$, $i = 1, \ldots, n$. Задаем малое число $\varepsilon_1 \ll 1$ и полагаем $\frac{\partial f}{\partial x^i} = 0$, если $\left|\frac{\partial f}{\partial x^i}\right| \leqslant \varepsilon_1$. Таким образом, спуск производится лишь по тем переменным, в направлении которых производная функции достаточно велика. Это позволяет быстро спуститься на «дно оврага».

б) Задаем некоторое большое число $\varepsilon_2 \gg 1$ и используем градиентный метод, полагая $\frac{\partial f}{\partial x^i} = 0$, если $\left|\frac{\partial f}{\partial x^i}\right| \geqslant \varepsilon_2$. В этом случае перемещение происходит по «берегу» оврага вдоль его «дна».

Комбинируя процедуры а) и б), можно поступать следующим образом. В подпространстве «быстрых переменных», т. е. с помощью алгоритма а), мы спускаемся до тех пор, пока метод не зацикливается, т. е. до тех пор, пока каждая следующая итерация позволяет найти точку, в которой значение функции меньше, чем значение, найденное в предыдущей итерации. После этого мы «включаем» аналогичную процедуру б) в пространстве «медленных переменных».

Создание диалоговых систем человек-машина и использование их в системах управления делает подобные комбинированные методы весьма эффективным средством использования идей оптимизации.

Другие идеи лежат в основе так называемого овражного метода, предложенного И. М. Гельфандом в начале 60-х годов.

Пусть x_0 и $\tilde{x}_0$ — две произвольные близкие точки (рис. 1.8). Из точки x_0 совершаем обычный градиентный

спуск и после нескольких итераций с малым шагом τ попадем в точку x_1. Тоже самое делаем для точки $\tilde{x}_0$, получая точку $\tilde{x}_1$. Две точки x_1, $\tilde{x}_1$ лежат в окрестности «дна оврага». Соединяя их прямой, делаем «большой шаг» λ в полученном направлении, перемещаясь «вдоль оврага». (Шаг λ называют овражным шагом.) В результате получаем точку x_2. В ее окрестности выбираем точку $\tilde{x}_2$ и повторяем процедуру.

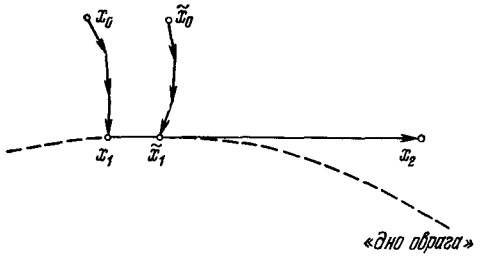


Рис. 1.8.

Многие из релаксационных методов, и метод овражного шага в том числе, являются эвристическими и их сходимость строго не установлена. Значение подобных методов с увеличением мощности ЭВМ, пакетов прикладных программ и диалоговых средств пользования ими непрерывно растет.

7. Методы покоординатного спуска. Стремление уменьшить объем вычислительной работы, требуемой для осуществления одной итерации метода наискорейшего спуска, привело к созданию ряда других методов. Одним из них является *метод покоординатного спуска.*

Пусть

$$x_0 = \{x_0^1, \ \ldots, \ x_0^n\}^T$$

— начальное приближение. Вычислим частную производную по первой координате $\frac{\partial f(x_0)}{\partial x^1}$ и примем

$$x_1 = x_0 - \alpha_0 \frac{\partial f(x_0)}{\partial x^1} e_1,$$

где $e_1 = \{1, 0, \ldots, 0\}^T$ — единичный вектор оси x^1.

Следующая итерации состоит в вычислении точки x_2 по формуле

$$x_2 = x_1 - \alpha_1 \frac{\partial f(x_1)}{\partial x^2} e_2,$$

где $e_2 = \{0, 1, 0, \ldots, 0\}^T$ — единичный вектор оси x^2 и т. д.

Таким образом, в методе покоординатного спуска мы спускаемся по ломаной, состоящей из отрезков прямых, параллельных координатным осям (рис. 1.9).

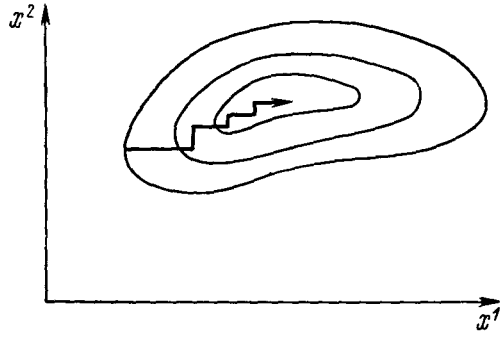


Рис. 1 9.

Спуск по всем n координатам составляет одну «внешнюю» итерацию. Пусть k — номер очередной внешней итерации, а s — номер той координаты, по которой производится спуск. Тогда рекуррентная формула, определяющая следующее приближение к точке минимума, запишется в следующем виде:

$$x_{kn+s+1} = x_{kn+s} - \alpha_{kn+s} \frac{\partial f(x_{kn+s})}{\partial x^s} e_s,$$

$$k = 1, 2, \ldots, \quad s = 1, 2, \ldots, n,$$

или, в координатной форме,

$$x^i_{kn+s+1} = x^i_{kn+s}, \qquad i \neq s,$$

$$x^i_{kn+s+1} = x^i_{kn+s} - \alpha_{kn+s} \frac{\partial f(x_{kn+s})}{\partial x^s}, \qquad i = s;$$

$$k = 1, 2, \ldots, \quad s = 1, 2, \ldots, \quad n.$$

После $s = n$ счетчик числа больших итераций k увеличивается на единицу, а s принимает значение, равное единице.

Величина шага α_k выбирается на каждой итерации аналогично тому, как это делалось в схемах, изложенных в пп. 2, 3. Если $\alpha_k = \alpha$ постоянно, то имеем *покоординатный спуск с постоянным шагом*. Если же шаг α_k выбирается из условия минимума функции

$$\varphi(\alpha) = f\left(x_{kn+s} - \alpha \frac{\partial f}{\partial x^s} e_s\right),$$

то мы получаем координатный аналог метода наискорейшего спуска, называемый обычно *методом Гаусса* (или *Гаусса — Зейделя*). Заметим, что метод Гаусса — Зейделя может рассматриваться как частный случай общего релаксационного процесса (1.15), когда на каждой итерации номера k все элементы матрицы B_k, кроме одного, стоящего на диагонали, равны нулю. Положение ненулевого элемента на диагонали определяется номером итерации.

8. Заключительные замечания. Как мы видели, градиентные методы достаточно просты в реализации и могут использоваться для минимизации различных по характеру функций. В этом их несомненное достоинство. Однако они плохо работают (медленно сходятся), если матрица вторых производных минимизируемой функции $f''(x)$ плохо обусловлена. Это соответствует сложной топографической структуре функции $f(x)$ — наличию «оврагов» или «хребтов». Возможные пути преодоления этой трудности — способы «исправления» поверхности $f(x)$, рассмотренные в пп. 5 и 6 этого параграфа. Другой выход — использование вторых производных минимизируемой функции $f(x)$ для построения методов, которые не «реагируют» на овражную структуру функции $f(x)$.

§ 2. Метод Ньютона

1. Схема метода. Геометрическая интерпретация. Мы переходим к изложению методов второго порядка, использующих вторые частные производные минимизируемой функции $f(x)$. Все они являются прямым обобщением известного метода Ньютона отыскания корня уравнения

$$\varphi(x) = 0, \tag{2.1}$$

где $\varphi(x)$ — скалярная функция скалярного аргумента x. Напомним содержание этого метода. Разложение

функции $\varphi(x)$ в окрестности некоторой точки x_k позволяет переписать уравнение (2.1) в следующем виде:

$$\varphi(x) = \varphi(x_k) + (x - x_k)\,\varphi'(x_k) + o(|x - x_k|) = 0. \quad (2.2)$$

Отбрасывая в (2.2) малые высшего порядка, получим линейное уравнение относительно неизвестной величины x, решением которого будет

$$x = x_k - \frac{\varphi(x_k)}{\varphi'(x_k)}. \quad (2.3)$$

Значение аргумента x, определяемое по этой формуле, дает новое приближение корня уравнения (2.1). Таким образом, метод Ньютона отыскания решения уравнения (2.1) описывается следующей рекуррентной формулой:

$$x_{k+1} = x_k - \frac{\varphi(x_k)}{\varphi'(x_k)}. \quad (2.4)$$

Процесс (2.4) называют также методом касательных для решения уравнения (2.1). Это название возникло из его геометрической интерпретации (рис. 2.1).

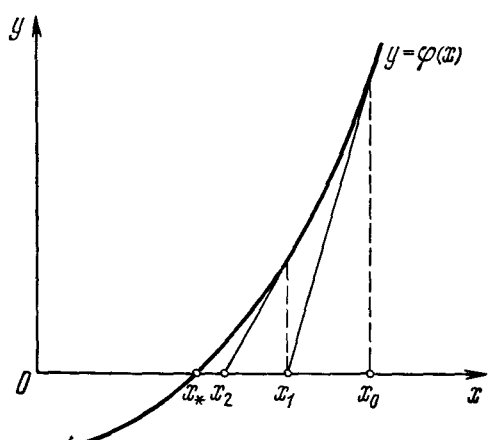


Рис. 2.1.

Проведем касательную к кривой $\varphi(x)$ в точке x_k. Точка пересечения этой касательной с осью абсцисс, x_{k+1}, определяется по формуле (2.4). Таким образом, метод Ньютона для решения уравнения $\varphi(x) = 0$ состоит в после-

довательном построении точек пересечения касательных к кривой $\varphi(x)$ с осью абсцисс.

Пусть теперь $\varphi(x)$ — n-мерная вектор-функция векторного аргумента x той же размерности. Тогда для решения системы уравнений $\varphi(x)=0$ мы можем использовать итерационный процесс, аналогичный (2.4). В самом деле, раскладывая покоординатно вектор-функцию $\varphi(x)$ в ряд Тейлора, получим в линейном приближении систему уравнений

$$\varphi_i\left(x_k^1, x_k^2, \ldots, x_k^n\right)+\sum_{s=1}^{n}\left(x^s-x_k^s\right)\frac{\partial\varphi_i}{\partial x^s}(x_k)=0,$$
$$i=1, 2, \ldots, n.$$

Ее решение имеет вид

$$x_{k+1}=x_k-(\varphi'(x_k))^{-1}\varphi(x_k), \tag{2.5}$$

где

$$\varphi'(x_k)=\left\{\frac{\partial\varphi_i}{\partial x^s}(x_k)\right\}$$

— квадратная $n\times n$ матрица.

Рассмотрим теперь случай, когда вектор-функция $\varphi(x)$ является градиентом некоторой скалярной функции $f(x)$, т. е.

$$\varphi(x)=f'(x).$$

Приравнивая ее нулю, приходим к системе уравнений, определяющей координаты стационарных точек функции $f(x)$. Формула метода Ньютона для решения этой системы выглядит так:

$$x_{k+1}=x_k-(f''(x_k))^{-1}f'(x_k), \tag{2.6}$$

и получается заменой в (2.5) $\varphi(x_k)$ на $f'(x_k)$.

Итерационный процесс (2.6) строит последовательность точек $\{x_k\}$, которая при определенных предположениях сходится к некоторой стационарной точке x_* функции $f(x)$, т. е. к точке, в которой $f'(x_*)=0$. Если матрица вторых производных $f''(x_*)$ положительно определена, эта точка будет точкой строгого локального минимума функции $f(x)$.

Процесс (2.6) допускает также отличную от рассмотренной выше интерпретацию. Предположим, что матрица вторых производных $f''(x)$ удовлетворяет условию

$$m\|y\|^2\leqslant(f''(x)y, y)\leqslant M\|y\|^2, \quad M\geqslant m>0,$$

при любых $x,\ y \in E_n$, и аппроксимируем функцию $f(x)$ в окрестности точки x_k квадратичной формой

$$\tilde{f}(x) = f(x_k) + (f'(x_k),\ x - x_k) + \frac{1}{2}(f''(x_k)(x - x_k), x - x_k). \quad (2.7)$$

Эта форма имеет единственную точку минимума, которая является корнем уравнения

$$\tilde{f}'(x) = 0 = f'(x_k) + f''(x_k)(x - x_k),$$

т. е. совпадает с x_{k+1} (см. (2.6)). Таким образом, метод Ньютона интерпретируется как последовательный поиск точек минимума квадратичных аппроксимирующих функций вида (2.7). Геометрически итерационный процесс (2.6) для минимизации функции $f(x)$ выглядит так (рис. 2.2): в точке x_k функция $f(x)$ аппроксимируется параболой $\tilde{f}(x)$, а в качестве приближения x_{k+1} принимается точка, соответствующая минимальной ординате этой параболы.

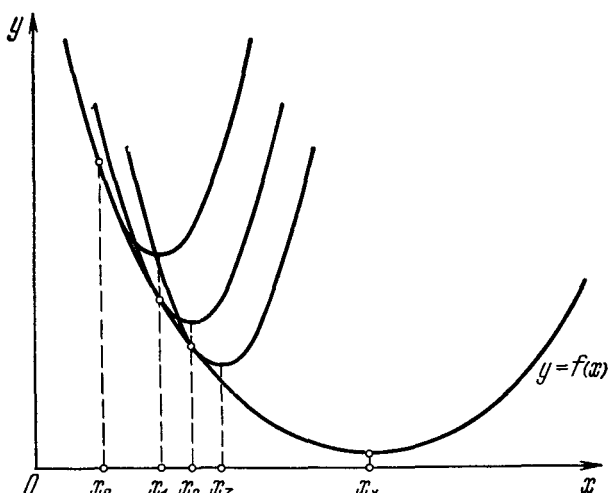


Рис. 2.2.

Следует ожидать, что процесс отыскания минимума с помощью метода Ньютона окажется более эффективным (т. е. потребует меньшего числа итераций), чем градиентные методы, так как квадратичная функция локально точ-

нее аппроксимирует минимизируемую функцию, чем линейная, по сути дела лежащая в основе градиентных методов. Сравнение метода Ньютона и градиентного спуска с точки зрения аппроксимации целевой функции представлено на рис. 2.3.

Заметим попутно, что при использовании градиентных методов приходится выбирать определенную длину шага α_k вдоль направления антиградиента в точке x_k, $-f'(x_k)$, так как линейная аппроксимирующая функция не имеет конечных точек экстремума. В формуле же (2.6) метода Ньютона

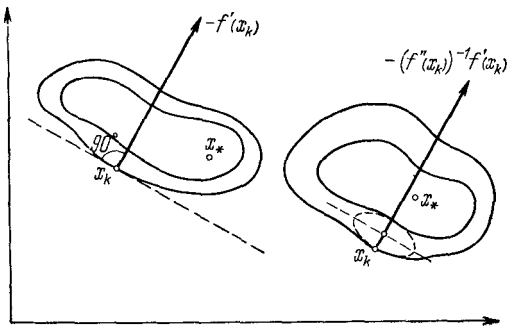


Рис. 2.3.

аппроксимирующая квадратичная функция (2.7) имеет конечную точку минимума. Поэтому шаг вдоль направления на эту точку, $-(f''(x_k))^{-1}f'(x_k)$, не выбирается, а полагается равным единице.

Рассмотрим теперь метод Ньютона в двумерном случае и продемонстрируем важность положительной определенности матрицы вторых производных (матрицы Гессе) целевой функции. Квадратичная аппроксимация (2.7) для функции двух переменных x^1, x^2 выглядит так:

$$\tilde{f}(x_k) = b_0 + b_1 x_k^1 + b_2 x_k^2 + b_{11}(x_k^1)^2 + b_{22}(x_k^2)^2 + b_{21} x_k^2 x_k^1 + b_{12} x_k^1 x_k^2. \tag{2.8}$$

Форму (2.8) можно преобразовать к каноническому виду

$$\tilde{f}(x_k) - \tilde{f}(x_*) = \tilde{b}_{11}(\tilde{x}_k^1)^2 + \tilde{b}_{22}(x_k^2)^2, \tag{2.9}$$

где x_* — стационарная точка функции $\tilde{f}(x)$. Преобразование квадратичной формы (2.8) к виду (2.9) включает перенос начала координат, при котором исчезают линейные члены, и поворот осей, в результате чего исключаются перекрестные члены. Коэффициенты формы (2.8) очевидным образом связаны с элементами матрицы Гессе функции $f(x)$:

$$\frac{1}{2}\frac{\partial^2 f}{(\partial x^1)^2}(x_k) = b_{11}, \quad \frac{1}{2}\frac{\partial^2 f}{\partial x^1 \partial x^2}(x_k) = b_{12} = \frac{1}{2}\frac{\partial^2 f}{\partial x^2 \partial x^1}(x_k),$$

$$\frac{1}{2}\frac{\partial^2 f}{(\partial x^2)^2}(x_k) = b_{22},$$

а коэффициенты $\tilde{b}_{11}$, $\tilde{b}_{22}$ канонического представления (2.9) являются собственными значениями матрицы $\frac{1}{2}f''(x_k)$. Если эти собственные значения положительны, то аппроксимирующая квадратичная функция имеет вид круговой или эллиптической впадины (рис. 2.4).

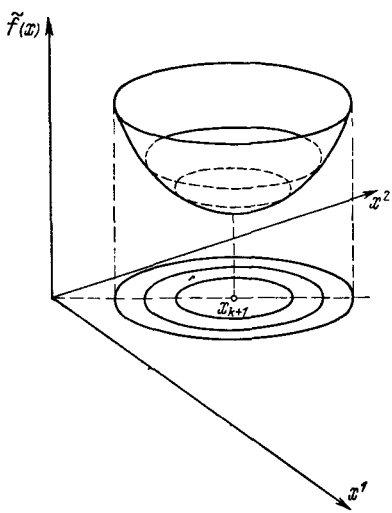


Рис. 2.4.

Если знаки коэффициентов $\tilde{b}_{11}$, $\tilde{b}_{22}$ противоположны, функция $\tilde{f}(x)$ представляет собой гиперболический параболоид (рис. 2.5) — поверхность с седловой точкой. Эта точка и будет взята в качестве следующего приближения x_{k+1} в методе Ньютона, хотя величина $\tilde{f}(x_{k+1})$ может оказаться больше, чем $\tilde{f}(x_k)$ (рис. 2.5). Последнее, скорее всего, приведет к тому, что значение функции $f(x)$ в точке x_{k+1} также будет больше, чем в x_k, и мы будем удаляться от искомой точки минимума x_* вместо того, чтобы приближаться к ней. Таким образом, на сходимость метода Ньютона к x_* можно рассчитывать только в том случае, когда матрица Гессе целевой функции положительно определена на каждой итерации.

Как мы видели в § 1, если числа $\tilde{b}_{11}$, $\tilde{b}_{22}$ одного знака, но сильно отличаются по величине, квадратичная функция (2.8) имеет овраг. Градиентные методы для минимизации такой функции работают очень плохо. Метод же Ньютона находит минимум квадратичной функции за один

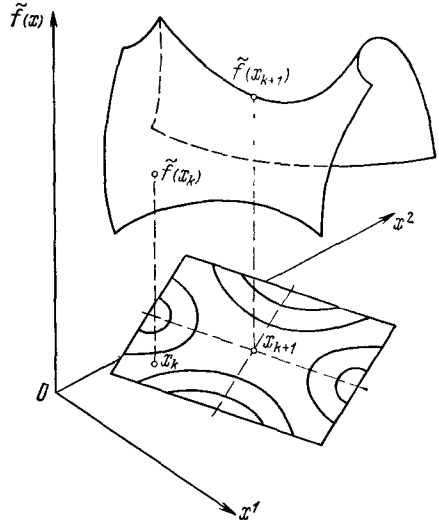


Рис. 2.5.

шаг, независимо от начального приближения x_0 и степени овражности. Вернемся к примеру 1.2 § 1 настоящей главы. В этом примере

$$f(x) = (x^1)^2 + 16\,(x^2)^2,$$

и овраг вытянут вдоль оси x^1. Легко убедиться, что направление спуска

$$-(f''(x_0))^{-1} f'(x_0),$$

вычисленное в различных точках x_0, всегда совпадает с направлением в точку минимума $x_* = \{0,\ 0\}^T$ (рис. 2.6). В самом деле, пусть снова $x_0 = \{2,2\}^T$. Тогда

$$f'(x_0) = \{4,64\}^T,$$

$$-(f''(x_0))^{-1} f'(x_0) = -1 \begin{bmatrix} 1/2 & 0 \\ 0 & 1/32 \end{bmatrix} \begin{Bmatrix} 4 \\ 64 \end{Bmatrix} = \begin{Bmatrix} -2 \\ -2 \end{Bmatrix}.$$

Возьмем теперь другую начальную точку $x_0 = \{2, 0\}^T$. При этом

$$-(f''(x_0))^{-1} f'(x_0) = \begin{Bmatrix} -2 \\ 0 \end{Bmatrix},$$

и мы снова за один шаг попадаем в точку минимума $x_* = \{0, 0\}^T$.

В общем случае, когда минимизируемая функция не квадратична, вектор

$$-(f''(x_k))^{-1} f'(x_k)$$

не указывает в точку ее минимума, однако имеет большую составляющую вдоль оси оврага и значительно ближе к направлению на минимум, чем антиградиент. Этим и

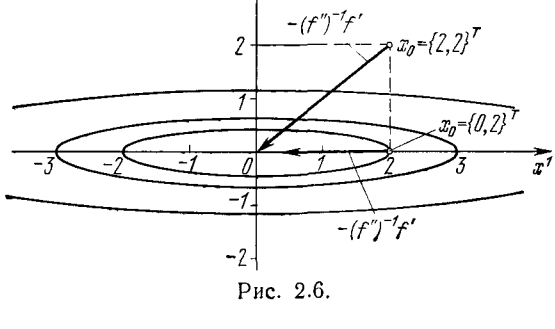


Рис. 2.6.

обусловлена более высокая сходимость метода Ньютона по сравнению с градиентным при минимизации овражных функций, которые, к сожалению, встречаются довольно часто.

Основные недостатки метода Ньютона состоят в том, что он, во-первых, предполагает вычисление вторых производных, что может быть связано с существенными трудностями, и, во-вторых, может расходиться, если целевая функция не является сильно выпуклой и начальное приближение находится достаточно далеко от минимума.

2. Сходимость метода Ньютона. Метод с регулировкой шага (Ньютона — Рафсона). На рис. 2.1 и 2.2 показаны функции, для которых метод Ньютона сходится. Нетрудно, однако, построить примеры, в которых он разойдется.

Пример 2.1. Рассмотрим задачу отыскания корня уравнения $\varphi(x) = \operatorname{arctg} x = 0$. Применим для ее решения

метод Ньютона (2.4). Как видно из рис. 2.7, неудачный выбор начального приближения x_0 ($|x_0| > 1{,}57$ *)) приводит к расходящемуся процессу. Таким образом, сходимость метода Ньютона существенно зависит от начального приближения x_0. Покажем, что если оно выбрано достаточно близким к решению x_*, метод Ньютона (2.4) сходится, причем с квадратичной скоростью.

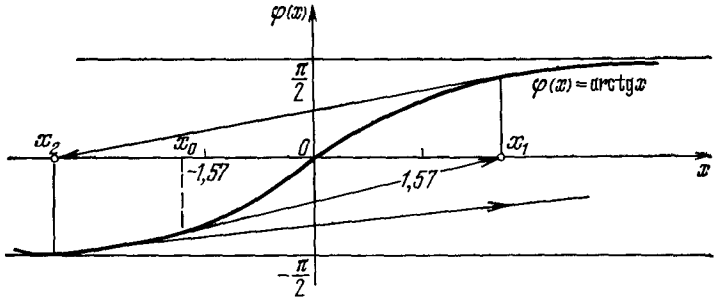


Рис. 2.7.

Теорема 2.1. *Пусть $\varphi(x)$ — дважды непрерывно дифференцируемая вектор-функция, а $\varphi'(x_*)$ — невырожденная матрица. Тогда существует окрестность точки x_* такая, что для любого начального приближения x_0 из этой окрестности метод Ньютона* (2.5) *сходится к x_* с квадратичной скоростью, т. е. справедлива оценка*

$$\|x_{k+1} - x_*\| \leqslant c\|x_k - x_*\|^2, \qquad (2.10)$$

где c — некоторая неотрицательная постоянная.

Доказательство. Заметим, прежде всего, что в силу невырожденности матрицы $\varphi'(x_*)$ матрица $\varphi'(x_k)$ при x_k, близких к x_*, также невырождена. Разложим $\varphi(x)$ в ряд Тейлора в окрестности точки x_*:

$$\varphi(x) = \varphi(x_*) + \varphi'(x_*)(x - x_*) + O(\|x - x_*\|^2).$$

При этом $\varphi(x_*) = 0$ и

$$\varphi'(x) = \varphi'(x_*) + O(\|x - x_*\|),$$

откуда

$$\varphi(x) = \varphi'(x)(x - x_*) + O(\|x - x_*\|^2)$$

*) $x_0 = 1{,}57$ — корень уравнения $2x = (1 + x^2)\operatorname{arctg} x$.

и

$$(\varphi'(x))^{-1}\varphi(x) = (x - x_*) + \varphi'(x)^{-1} O(\|x - x_*\|^2).$$

Поэтому для x, близких к x_*, имеем

$$\|x - \varphi'(x)^{-1}\varphi(x) - x_*\| = O(\|x - x_*\|^2),$$

т. е. существуют числа $\delta > 0$, $c > 0$ такие, что при x, удовлетворяющих неравенству

$$\|x - x_*\| < \delta,$$

будет

$$\|x - \varphi'(x)^{-1}\varphi(x) - x_*\| \leqslant c\|x - x_*\|^2. \qquad (2.11)$$

Возьмем теперь x_0 такое, что

$$\|x_0 - x_*\| \leqslant \hat{\delta},$$

где $\hat{\delta} = \min\left\{\delta, \frac{1}{(1+\beta)c}\right\}$, $\beta > 0$, — некоторое число. Тогда из (2.11) следует, что метод Ньютона (2.4) сходится, начиная из x_0, причем выполнена оценка (2.10). Теорема доказана.

Совершенно аналогично доказывается сходимость с квадратичной скоростью метода Ньютона (2.6) для минимизации функции $f(x)$ (начальное приближение x_0 должно быть достаточно близким к стационарной точке x_*).

Итак, удачный выбор начального приближения x_0 гарантирует сходимость методов Ньютона (2.4), (2.6). Однако отыскание подходящего начального приближения — далеко не тривиальная задача. Поэтому необходимо как-то изменить формулы (2.4), (2.6), чтобы добиться сходимости соответствующих процессов независимо от начального приближения. Можно показать, что в некоторых предположениях для этого достаточно в методе Ньютона кроме направления движения $(f''(x_k))^{-1} f'(x_k)$ (или $\varphi'(x_k)^{-1}\varphi(x_k)$) выбирать и длину шага вдоль него. Соответствующие алгоритмы называются методами Ньютона с регулировкой шага (методами Ньютона — Рафсона) и выглядят так:

$$x_{k+1} = x_k - \alpha_k (\varphi'(x_k))^{-1}\varphi(x_k), \qquad (2.12)$$

$$x_{k+1} = x_k - \alpha_k (f''(x_k))^{-1} f'(x_k). \qquad (2.13)$$

Возьмем для определенности метод (2.13) минимизации функции $f(x)$. Как и в градиентных методах, в нем величина α_k выбирается так, чтобы обеспечить убывание целе-

вой функции на каждой итерации. Мы рассмотрим два способа выбора длины шага α_k. Первый из них связан с проверкой неравенства вида (1.4)

$$f(x_k - \alpha_k (f''(x_k))^{-1} f'(x_k)) - f(x_k) \leqslant +\varepsilon \alpha_k (f'(x_k), p_k), \quad (2.14)$$

где

$$p_k = -(f''(x_k))^{-1} f'(x_k)$$

— направление спуска, а $0 < \varepsilon < \frac{1}{2}$ — некоторое число. Если это неравенство выполнено при $\alpha_k = 1$, то шаг принимается равным единице и осуществляется следующая итерация. Если нет — шаг дробится до тех пор, пока оно не выполнится. Второй метод определения шага α_k в схеме (2.13), как и в методе наискорейшего спуска, состоит в минимизации функции

$$f(x_k - \alpha (f''(x_k))^{-1} f'(x_k))$$

по α в направлении движения:

$$f(x_k - \alpha_k (f''(x_k))^{-1} f'(x_k)) = \min_{\alpha \geqslant 0} (f(x_k - \alpha (f''(x_k))^{-1} f'(x_k)). \quad (2.15)$$

Проиллюстрируем сказанное на функции из примера 2.1. Возьмем снова $|x_0| > 1{,}57$, но искать решение теперь будем по схеме (2.12) с выбором шага по способу (2.14). Тогда, как видно из рис. 2.8, последовательные итерации быстро сходятся при том же, что и прежде, выборе начального приближения x_0.

Сходимость метода Ньютона (2.13) с регулировкой шага устанавливается следующей теоремой.

Теорема 2.2. *Пусть дважды непрерывно дифференцируемая функция сильно выпукла и матрица ее вторых производных удовлетворяет условию Липшица*

$$\|f''(x) - f''(y)\| \leqslant R_1 \|x - y\|, \quad x, y \in E_n. \quad (2.16)$$

Тогда последовательность (2.13), в которой α_k выбирается из условий (2.14) или (2.15), сходится, независимо от начальной точки x_0, к точке минимума x_* с квадратичной скоростью

$$\|x_{k+1} - x_*\| \leqslant \frac{R_1}{m} \|x_k - x_*\|^2.$$

Здесь, как и выше, m — оценка наименьшего собственного числа матрицы $f''(x)$. Доказательство теоремы можно найти в [9]. Если матрица $f''(x)$ не удовлетворяет условию Липшица, сходимость метода Ньютона сверхлинейная.

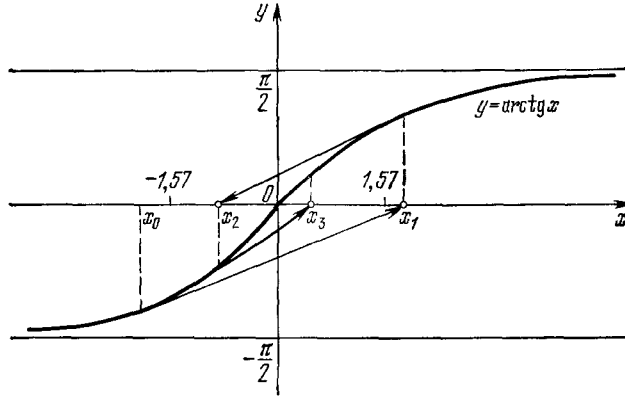


Рис. 2.8.

3. Модификации метода Ньютона. Все трудности, возникающие при практической реализации метода Ньютона, можно условно разбить на две группы. Первые связаны с необходимостью вычислять матрицу $f''(x)$. (Заметим, что, хотя в изложенных ранее схемах метода Ньютона фигурирует обратная к ней матрица $(f''(x_k))^{-1}$, на практике нет необходимости вычислять последнюю, так как направление спуска

$$p_k = -(f''(x_k))^{-1} f'(x_k)$$

можно найти как решение системы линейных уравнений

$$f''(x_k)\, p_k = -f'(x_k)$$

каким-нибудь из методов исключения.) Мы рассмотрим ниже две модификации метода Ньютона, которые используют не точные значения, а некоторые приближенные аналоги матрицы вторых производных. В результате уменьшается трудоемкость методов, но, конечно, ухудшается их сходимость. Ко второй группе можно отнести все осложнения, возникающие в связи с нарушением в процессе

счета положительной определенности матрицы вторых производных. Для преодоления этих трудностей предназначена третья из предлагаемых ниже модификаций

В качестве первой модификации метода Ньютона рассмотрим следующий алгоритм:

$$x_{k+1} = x_k - \alpha_k (f''(x_0))^{-1} f'(x_k), \quad \alpha_k \geqslant 0. \qquad (2.17)$$

Здесь для построения направления спуска используется один раз вычисленная и обращенная матрица вторых производных $f''(x_0)$. Если интерпретировать схему (2.17) как метод касательных для решения уравнения $\frac{df}{dx} = 0$, то геометрически мы в каждой точке последовательности $\{x_k\}$, начиная с x_1, проводим для кривой $\frac{df}{dx}$ прямые, параллельные первой касательной (рис. 2.9).

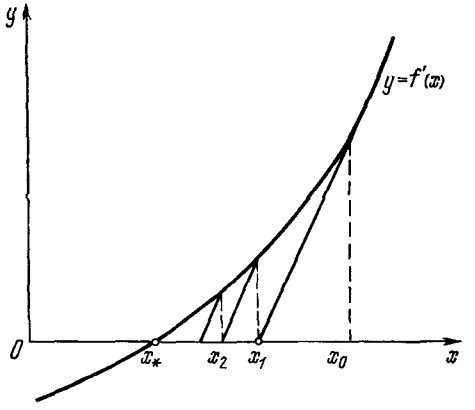


Рис. 2.9.

Очевидно, что если матрица $f''(x)$ положительно определена, итерационный процесс (2.17) является одной из модификаций градиентного спуска (см. релаксационные методы) и сходится, независимо от начального приближения x_0, со скоростью геометрической прогрессии, причем знаменатель прогрессии q тем меньше, чем ближе точка x_0 к x_*.

Другая известная модификация метода Ньютона связана с обновлением матрицы вторых производных через определенное количество шагов.

Итерационный процесс имеет вид

$$x_{\xi t+i+1} = x_{\xi t+i} - \alpha_{\xi t+i} (f''_{\xi t})^{-1} f'_{\xi t+i}, \quad \alpha_{\xi t+i} \geqslant 0,$$
$$k = \xi t + i, \quad \xi = 0, 1, \ldots, \quad i = 0, 1, \ldots, t-1, t, \tag{2.18}$$
$$f''_{\xi t} = f''(x_{\xi t}), \quad f'(x_{\xi t+i}) = f'_{\xi t+i}.$$

Здесь $t > 0$ — произвольное целое число. Очевидно, метод занимает промежуточное положение между обычным методом Ньютона и методом (2.17).

Для скорости сходимости метода в условиях теоремы 2.2 справедлива оценка (см., например, [9]):

$$\| x_{(\xi+1)\, t} - x_* \| \leqslant c \| x_{\xi t} - x_* \|^{t+1}, \quad c \geqslant 0. \tag{2.19}$$

При $t = 1$ метод (2.18) переходит в обычный метод Ньютона (2.13), а скорость сходимости (2.19) квадратична.

Предположим теперь, что матрица Гессе минимизируемой функции не является положительно определенной. В этом случае, как мы знаем, последовательность точек $\{x_k\}$, вырабатываемая по методу Ньютона, может расходиться. Левенберг (1944) и Маркардт (1963) предложили добавлять к матрице вторых производных на каждом шаге величину $\lambda_k E$, где λ_k — некоторое число, а E — единичная матрица. Соответствующий итерационный процесс имеет вид

$$x_{k+1} = x_k - (f''(x_k) + \lambda_k E)^{-1} f'(x_k). \tag{2.20}$$

Чтобы избежать вычисления длины шага α_k, здесь ее полагают равной единице, а величины λ_k выбирают так, чтобы выполнялись условия

$$\cos \widehat{((f''(x_k) + \lambda_k E)^{-1} f'(x_k),\ f'(x_k))} \geqslant \varepsilon_1 > 0,$$
$$f(x_{k+1}) - f(x_k) \leqslant -\varepsilon_2 ((f''(x_k) + \lambda_k E)^{-1} f'(x_k),\ f'(x_k)), \tag{2.21}$$
$$0 < \varepsilon_2 < \frac{1}{2},$$

обеспечивающие сходимость. В этих неравенствах ε_1, ε_2 — заданные постоянные. Первое неравенство означает, что угол между направлением спуска и антиградиентом в точке x_k должен быть острым, а выполнение второго гарантирует существенное убывание функции на каждой итерации. Параметры λ_k не определяются условиями (2.21) однозначно, и для их выбора существуют различные эвристические схемы, на которых мы останавливаться не

будем. Отметим только, что при правильном способе выбора λ_k метод (2.20) вдали от точки минимума ведет себя как градиентный, а при приближении к x_* должен переходить в обычный метод Ньютона (2.6).

4. Метод секущих. Говоря о скорости сходимости того или иного метода, обычно считают, что она тем лучше, чем меньшее количество итераций необходимо произвести для получения приближенного решения с заданной точностью. С этой точки зрения самым эффективным является метод Ньютона, скорость сходимости которого квадратична. Большой практический интерес представляет другое понимание эффективности метода, когда в качестве критерия принимают число машинных операций, необходимых для достижения заданной точности. При такой оценке более предпочтительным может оказаться метод, который хотя и сходится медленнее, т. е. требует для достижения заданной точности большего числа итераций, но расчет каждой из итераций занимает меньше времени. К сожалению, для подобной оценки метода нет иных эффективных способов, кроме экспериментальной проверки. Именно поэтому для решения трудных оптимизационных задач обычно используют на разных этапах расчета различные методы. Одним из методов, который сходится медленнее метода Ньютона, однако менее трудоемок на каждой итерации и поэтому в отдельных случаях может оказаться более выгодным, является метод секущих. Он представляет собой конечно-разностный аналог метода Ньютона. Нам будет удобнее проиллюстрировать его на задаче поиска корня уравнения $\varphi(x)=0$.

Пусть сначала x — скаляр Обозначим через x_0, x_1 две точки, в которых функция $\varphi(x)$ принимает значения разных знаков. Пусть, далее, на отрезке $[x_0, x_1]$ расположен единственный корень x_* уравнения $\varphi(x)=0$ (рис. 2.10).

Уравнение прямой, проходящей через точки $\{x_0, \varphi(x_0)\}$ и $\{x_1, \varphi(x_1)\}$, имеет вид

$$y=\frac{\varphi(x_1)-\varphi(x_0)}{x_1-x_0}x+\frac{\varphi(x_0)x_1-\varphi(x_1)x_0}{x_1-x_0}=ax+b.$$

В качестве приближения корня выберем точку ее пересечения с осью абсцисс. Эта точка, x_2, определяется по формуле

$$x_2=\frac{\varphi(x_1)x_0-\varphi(x_0)x_1}{\varphi(x_1)-\varphi(x_0)},$$

Теперь проведем прямую, проходящую через точки $\{x_0, \varphi(x_0)\}$, $\{x_2, \varphi(x_2)\}$. Ее пересечение с осью абсцисс принимаем за очередное приближение x_3 и т. д. Общая формула итерационного процесса имеет вид

$$x_{k+1} = \frac{\varphi(x_k)\, x_0 - \varphi(x_0)\, x_k}{\varphi(x_k) - \varphi(x_0)}. \qquad (2.22)$$

Рассмотрим теперь метод секущих для решения векторного уравнения

$$F(x) = 0, \qquad (2.23)$$

где $F(x)$ — n-мерная вектор-функция векторного аргумента x

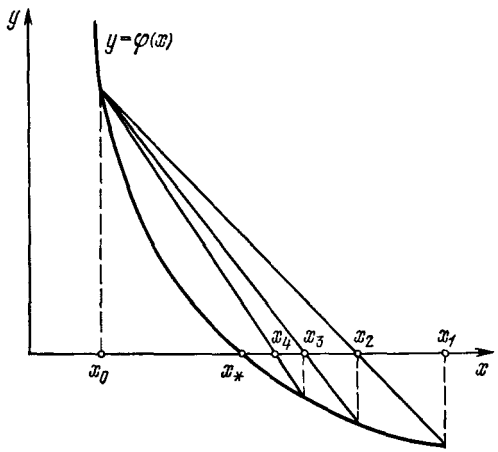


Рис. 2.10.

размерности n. Уравнение (2.23) эквивалентно n скалярным

$$F_i(x) = 0, \quad i = 1, \ldots, n.$$

Зададим $n+1$ точку

$$x_0 = \{x_0^1, \ldots, x_0^n\}^T, \ldots, \{x_n^1, \ldots, x_n^n\}^T \qquad (2.24)$$

и вычислим значения функций $F_i(x)$ в каждой из них:

$$F_i(x_0), F_i(x_1), \ldots, F_i(x_n), \quad i = 1, \ldots, n.$$

По аналогии с одномерным случаем проведем в $2n$-мерном пространстве «секущие», проходящие через точки

$$\{x_0, F(x_0)\}, \{x_1, F(x_1)\}, \ldots, \{x_n, F(x_n)\}.$$

Это – гиперплоскости, параметры которых a_j^i, b^i, i, $j = 1, \ldots, n$, удовлетворяют уравнениям

$$\begin{aligned} a_1^i x_0^1 + \ldots + a_n^i x_0^n - b^i &= F_{0i} = F_i(x_0), \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ a_1^i x_n^1 + \ldots + a_n^i x_n^n - b^i &= F_{ni} = F_i(x_n) \end{aligned} \tag{2.25}$$

для всех $i = 1, \ldots, n$.

Эта система разрешима относительно величин a_j^i, b^i, если не равен нулю определитель

$$\begin{vmatrix} 1 & x_0^1 & \ldots & x_0^n \\ 1 & x_1^1 & \ldots & x_1^n \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_n^1 & \ldots & x_n^n \end{vmatrix}.$$

Будем считать, что начальные точки (2.24) выбраны так, что это условие выполнено.

Далее, по аналогии с одномерным случаем, в качестве следующего приближения $\tilde{x}$ возьмем «точку пересечения» «секущих»

$$y^i = a_1^i x^1 + \ldots + a_n^i x^n - b^i, \qquad i = 1, \ldots, n,$$

с осями координат, т. е. будем искать точку $\tilde{x}$, удовлетворяющую системе уравнений

$$a_1^i \tilde{x}^1 + \ldots + a_n^i \tilde{x}^n - b^i = 0, \quad i = 1, 2, \ldots, n. \tag{2.26}$$

Чтобы избежать вычисления величин $a_1^i, \ldots, a_n^i$, b^i, $i = 1, \ldots, n$, для определения $\tilde{x}$, используем следующий искусственный прием. Вычтя равенства (2.26) из (2.25), получим

$$\begin{aligned} a_1^i (x_0^1 - \tilde{x}^1) + \ldots + a_n^i (x_0^n - \tilde{x}^n) &= F_{0i}, \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ a_1^i (x_n^1 - \tilde{x}^1) + \ldots + a_n^i (x_n^n - \tilde{x}^n) &= F_{ni}, \\ i = 1, 2, \ldots, n. & \end{aligned} \tag{2.27}$$

Составим теперь матрицу

$$\begin{bmatrix} x_0^1 - \tilde{x}^1 & \ldots & x_0^n - \tilde{x}^n & F_{01} & \ldots & F_{0n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ x_n^1 - \tilde{x}^1 & \ldots & x_n^n - \tilde{x}^n & F_{n1} & \ldots & F_{nn} \end{bmatrix}.$$

Ее ранг, как следует из (2.27), не больше, чем n, так

как каждый из последних столбцов этой матрицы является линейной комбинацией n первых. Поэтому все определители

$$\begin{vmatrix} x_0^i - \tilde{x}^i & F_{01} & \dots & F_{0n} \\ \dots & \dots & \dots & \dots \\ x_n^i - \tilde{x}^i & F_{n1} & \dots & F_{nn} \end{vmatrix}, \quad i = 1, 2, \dots, n,$$

равны нулю, откуда

$$\tilde{x}^i = \frac{\Delta_i}{\Delta}, \tag{2.28}$$

где

$$\Delta = \begin{vmatrix} 1 & F_{01} & \dots & F_{0n} \\ \dots & \dots & \dots & \dots \\ 1 & F_{n1} & \dots & F_{nn} \end{vmatrix}, \quad \Delta_i = \begin{vmatrix} x_0^i & F_{01} & \dots & F_{0n} \\ \dots & \dots & \dots & \dots \\ x_n^i & F_{n1} & \dots & F_{nn} \end{vmatrix}.$$

Чтобы (2.28) имело смысл, необходимо потребовать неравенство нулю определителя Δ.

Теперь нужно заменить одну из точек $x_0, \dots, x_n$ точкой $\tilde{x}$ и повторить итерацию. Например, можно заменить ту точку, для которой величина

$$\sum_{i=1}^{n} (F_i)^2$$

наибольшая, и т. д. Если выбрать исходные точки следующим образом:

$$\begin{aligned} x_0 &= \{x_0^1, x_0^2, \dots, x_0^n\}^T, \\ x_1 &= \{x_0^1 + \Delta x^1, x_0^2, \dots, x_0^n\}^T, \\ &\dots\dots\dots\dots\dots\dots \\ x_n &= \{x_0^1, x_0^2, \dots, x_0^n + \Delta x^n\}^T, \end{aligned}$$

то при $\Delta x^i \to 0$ метод секущих перейдет в метод Ньютона

Чтобы сравнить метод секущих и метод Ньютона по трудоемкости, рассмотрим схему (2.4) для решения методом Ньютона системы уравнений

$$F(x) = 0. \tag{2.29}$$

Соответствующий итерационный процесс имеет вид

$$x_{k+1} = x_k - (F'(x_k))^{-1} F(x_k). \tag{2.30}$$

Допустим, что матрица $F'(x)$ вычисляется по конечно-разностным формулам. Тогда для определения $(k+1)$-го при-

ближения к решению уравнения (2.29), исходя из k-го приближения, требуется $n+1$ раз вычислить вектор-функцию $F(x)$. Метод же секущих требует на каждом шаге вычисления лишь одного значения $F(x)$, а именно — значения $F(\tilde{x})$. Следовательно, количество вычислений на одной итерации в методе секущих меньше, чем в методе Ньютона.

§ 3. Метод сопряженных градиентов

1. Предварительные замечания. В этом параграфе будет изложен метод сопряженных градиентов, относящийся к группе методов сопряженных направлений. Этот метод, как и методы градиентного спуска, является методом первого порядка, т. е. использует информацию только о первых производных минимизируемой функции. Однако метод сопряженных градиентов выгодно отличается от градиентных методов более высокой скоростью сходимости, которая, при определенных предположениях относительно минимизируемой функции, приближается к скорости сходимости метода Ньютона. Положительно определенная квадратичная форма n переменных минимизируется методом сопряженных градиентов за n или менее шагов. Так как любая гладкая функция в окрестности точки своего минимума хорошо аппроксимируется квадратичной, метод сопряженных градиентов с успехом применяется для минимизации и неквадратичных функций. Правда, при этом метод перестает быть конечным, а становится итеративным.

Первоначально метод сопряженных градиентов был разработан Хестенсом и Штифелем (1952) для решения систем линейных алгебраических уравнений $Ax=b$ с симметричной, положительно определенной матрицей A. Очевидно, что решение этой системы эквивалентно минимизации квадратичной функции $\varphi(x)=(x, Ax)-(b, x)$. Поэтому развитие и использование метода сопряженных градиентов как метода минимизации представляется совершенно естественным.

Прежде, чем переходить к описанию конкретных алгоритмов, определим и проиллюстрируем на примерах свойство сопряженности векторов.

2. Сопряженность и сопряженные направления.

Определение 3.1. Два вектора x и y в пространстве E_n называют *H-сопряженными* (или сопряженными по

отношению к матрице H) или H-ортогональными, если

$$(x,\ Hy) = 0. \tag{3.1}$$

Сопряженность можно считать обобщением понятия ортогональности. В самом деле, когда $H = E$, векторы x и y в соответствии с уравнением (3.1) ортогональны.

Как было указано в п. 1 настоящего параграфа, квадратичная функция n переменных может быть минимизирована за n (или менее) шагов, если эти шаги предпринимать в сопряженных направлениях. Мы сначала проиллюстрируем это замечательное свойство сопряженных направлений на простом примере, а затем докажем его для квадратичной функции n переменных.

Пример 3.1. Рассмотрим задачу минимизации функции двух переменных

$$f(x) = (x^1)^2 + 4(x^2)^2 - 1.$$

Пусть x_0 — произвольная начальная точка, а s_0 — произвольное начальное направление поиска. Тогда следующую точку x_1 определим по формуле

$$x_1 = x_0 + \lambda^0 s_0, \tag{3.2}$$

в которой длину шага λ^0 вычислим из условия минимума функции $f(x)$ по λ в направлении движения, т. е. из условия

$$\frac{df(x_0 + \lambda^0 s_0)}{d\lambda^0} = 0. \tag{3.3}$$

Возьмем в качестве x_0 точку с координатами $\{1,1\}^T$, а в качестве s_0 — вектор $\{1,\ 2\}^T$. (Здесь вектор s_0 в иллюстративных целях взят произвольным, хотя можно было бы начать движение из точки x_0 по антиградиенту функции $f(x)$.) Координаты точки x_1 в соответствии с формулой (3.2) равны

$$x_1^1 = 1 + \lambda^0 = x_0^1 + \lambda^0 s_0^1,$$
$$x_1^2 = 1 + 2\lambda^0 = x_0^2 + \lambda^0 s_0^2.$$

Для вычисления длины шага, согласно (3.3), получим уравнение

$$\frac{df}{d\lambda^0} = 2(1 + \lambda^0) + 16(1 + 2\lambda^0) = 0,$$

откуда

$$\lambda^0 = -\frac{9}{17},\quad x_1^1 = \frac{8}{17},\quad x_1^2 = -\frac{1}{17}.$$

Выберем теперь направление s_1 движения из точки x_1 сопряженным к s_0 относительно матрицы вторых производных целевой функции, т. е. из условия

$$(s_1,\ f''_{(x)}\ s_0) = 0. \tag{3.4}$$

В данном примере матрица вторых производных постоянна и имеет вид

$$f'' = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix},$$

а направление минимизации s_1, согласно (3.4), таково:

$$s_1^1 = 1, \quad s_1^2 = -\frac{1}{8}.$$

Заметим, что из уравнения (3.4) компоненты вектора s_1 определяются неоднозначно с точностью до произвольного множителя. Следующую точку, x_2, вычисляем по формуле

$$x_2 = x_1 + \lambda^1 s_1$$

или, в координатной форме,

$$x_2^1 = x_1^1 + \lambda^1 s_1^1 = \frac{8}{17} + \lambda^1,$$

$$x_2^2 = x_1^2 + \lambda^1 s_1^2 = -\frac{1}{17} - \frac{\lambda^1}{8}.$$

Величину λ^1 снова определим, минимизируя функцию $f(x)$, по выбранному направлению

$$\frac{df(x_1 + \lambda^1 s_1)}{d\lambda^1} = 2\left(\frac{8}{17} + \lambda^1\right) + \left(\frac{1}{17} + \frac{1}{8}\lambda^1\right) = 0.$$

Отсюда получаем

$$\lambda^1 = -\frac{8}{17}$$

и

$$x_2^1 = \frac{8}{17} - \frac{8}{17} = 0, \quad x_2^2 = -\frac{1}{17} - \frac{1}{8} \cdot \left(-\frac{8}{17}\right) = 0.$$

На рис. 3.1 изображена траектория поиска. Как видно из примера, квадратичная функция двух переменных минимизируется за два шага, по одному в каждом из сопряженных направлений.

Рассмотрим теперь квадратичную функцию n переменных

$$f(x) = a + (x,\ b) + \frac{1}{2}(x,\ Hx) \tag{3.5}$$

с положительно определенной $(n \times n)$-матрицей H. Покажем, что функция (3.5) может быть минимизирована методом сопряженных направлений не более чем за n шагов.

Пусть $s_0, s_1, \ldots, s_{n-1}$ — заданная система H-сопряженных векторов. Для минимизации функции $f(x)$ возьмем следующий итерационный процесс:

$$x_{k+1} = x_k + \lambda^k s_k, \tag{3.6}$$

где s_k — направление спуска на k-м шаге, а величина шага λ^k выбирается из условия минимума функции $f(x)$ по λ в направлении движения. Нетрудно видеть, что λ^k вычисляются по формулам

$$\lambda^k = -\frac{(f'(x_k), s_k)}{(s_k, Hs_k)}. \tag{3.7}$$

Зададимся начальным приближением x_0. Применяя последовательно формулы (3.6), (3.7), на n-м шаге итерационного процесса получим

$$x_n = x_0 + \sum_{k=0}^{n-1} \lambda^k s_k = x_0 - \sum_{k=0}^{n-1} \frac{(f'(x_k), s_k)}{(s_k, Hs_k)} s_k. \tag{3.8}$$

Покажем теперь, что точка x_n, определяемая формулой (3.8), совпадает с точным минимумом $\hat{x}$ функции $f(x)$. Это и будет означать, что не более чем за n шагов итерационного процесса (3.6) мы придем в точку минимума.

Заметим, что минимум функции (3.5) достигается в точке $\hat{x}$, где

$$f'(\hat{x}) = b + H\hat{x} = 0.$$

Отсюда

$$\hat{x} = -H^{-1}b. \tag{3.9}$$

Нам потребуется для дальнейшего вспомогательная

Лемма 3.1. *H-сопряженные векторы $s_0, \ldots, s_{n-1}$ линейно независимы.*

Доказательство. Допустим противное. Тогда найдутся числа β_i, не все равные нулю, такие, что

$$s_k = \sum_{i=0}^{k-1} \beta_i s_i.$$

Умножая это равенство на вектор Hs_k, получим

$$(s_k, Hs_k) = 0,$$

что возможно лишь при $s_k = 0$, так как матрица H положительно определена. Лемма доказана.

Поскольку векторы $s_0, \ldots, s_{n-1}$ линейно независимы, мы можем представить вектор $\hat{x} - x_0$ в виде их линейной комбинации

$$\hat{x} = x_0 + \sum_{i=0}^{n-1} \alpha_i s_i. \tag{3.10}$$

Учитывая, что $\hat{x} = -H^{-1}b$ и

$$(f'(x_k),\ s_k) = (b + Hx_k,\ s_k) =$$

$$= \left(b + H\left(x_0 + \sum_{i=0}^{k-1} \lambda^i s_i\right),\ s_k\right) = (Hx_0 + b,\ s_k),$$

получим для α_i формулы

$$\alpha_i = -\frac{(f'(x_i),\ s_i)}{(s_i,\ Hs_i)}. \tag{3.11}$$

Сравнивая теперь (3.10) и (3.8), заключаем, что

$$x_n = \hat{x} = -H^{-1}b.$$

Таким образом, процедура (3.6) с выбором λ^k по формуле (3.7) действительно позволяет найти минимум квадратичной функции за n шагов. «Настоящих» шагов может быть меньше n, если некоторые λ^k окажутся нулями.

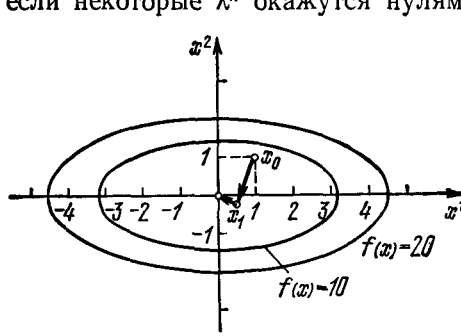


Рис. 3.1.

Чтобы воспользоваться изложенной выше схемой минимизации квадратичной функции (3.5), нужно знать n взаимно сопряженных направлений $s_0, s_1, \ldots, s_{n-1}$. Эффективное построение таких направлений — самостоятельная проблема. В принципе последовательность H-сопряженных

векторов $s_0, s_1, \ldots, s_{n-1}$ можно построить по следующему правилу. Зададимся произвольным ненулевым вектором s_0. Возьмем в качестве s_1 какое-нибудь нетривиальное решение уравнения

$$(s_1, Hs_0) = 0. \tag{3.12}$$

Далее найдем вектор s_2, H-сопряженный векторам s_0 и s_1, т. е. удовлетворяющий уравнениям

$$(s_2, Hs_1) = 0, \quad (s_2, Hs_0) = 0 \tag{3.13}$$

и т. д. Вектор s_r будет определяться как ненулевое решение уравнений

$$(s_r, Hs_i) = (s_i, Hs_r) = 0, \quad i = 0, \ldots, r-1. \tag{3.14}$$

Понятно, что в рамках описанной процедуры могут получаться различные наборы векторов $\{s_k\}$. В частности, для определения n компонент вектора s_1 у нас есть только одно уравнение (3.12), решение которого будет содержать $(n-1)$ произвольных постоянных. Для нахождения вектора s_2 имеем два уравнения (3.13). Соответственно, его n компонент определятся с точностью до $(n-2)$ произвольных постоянных и т. д.

Неоднозначностью в определении векторов s_i можно воспользоваться так, чтобы упростить процедуру их построения и, в частности, не решать вспомогательных уравнений (3.14). В методе сопряженных градиентов Флетчера — Ривса, который излагается ниже, выбор H-сопряженных направлений осуществляется совместно с одномерной минимизацией $f(x)$ по λ.

3. Метод Флетчера — Ривса. Этот метод был предложен в 1964 г., он использует последовательность направлений поиска, каждое из которых является линейной комбинацией антиградиента в текущей точке и предыдущего направления спуска.

Вернемся к квадратичной функции

$$f(x) = a + (x, b) + \frac{1}{2}(x, Hx).$$

При минимизации ее методом Флетчера — Ривса векторы s_k вычисляются по формулам

$$\begin{aligned} s_k &= -f'(x_k) + \beta_{k-1} s_{k-1}, \quad k \geqslant 1, \\ s_0 &= -f'(x_0). \end{aligned} \tag{3.15}$$

Величины β_{k-1} выбираются так, чтобы направления s_k, s_{k-1} были H-сопряженными:

$$(s_k,\ Hs_{k-1}) = 0 = -(f'(x_k),\ Hs_{k-1}) + \beta_{k-1}(s_{k-1},\ Hs_{k-1}).$$

Отсюда

$$\beta_{k-1} = \frac{(f'(x_k),\ Hs_{k-1})}{(s_{k-1},\ Hs_{k-1})}. \tag{3.16}$$

Точка x_{k+1} определяется в результате минимизации функции $f(x)$ в направлении s_k, исходящем из точки x_k, т. е.

$$x_{k+1} = x_k + \lambda^k s_k, \tag{3.17}$$

где λ^k доставляет минимум по λ функции $f(x_k + \lambda s_k)$ и может быть вычислена по формуле

$$\lambda^k = -\frac{(f'(x_k),\ s_k)}{(s_k,\ Hs_k)}.$$

Итак, предлагаемая процедура минимизации функции $f(x)$ выглядит следующим образом. В заданной начальной точке x_0 вычисляется антиградиент $s_0 = -f'(x_0)$. Осуществляется одномерная минимизация в этом направлении и определяется точка x_1. В точке x_1 снова вычисляется антиградиент $-f'(x_k)$. Так как эта точка доставляет минимум функции $f(x)$ вдоль направления $s_0 = -f'(x_0)$, вектор $f'(x_1)$ ортогонален $f'(x_0)$. Затем по известному значению $f'(x_1)$ по формуле (3.15) вычисляется вектор s_1, который, согласно (3.16), будет H-сопряженным к s_0. Далее отыскивается минимум функции $f(x)$ вдоль направления s_1 и т. д. (см. рис. 3.2).

Покажем теперь, что направления $s_0, \ldots, s_k$, получаемые по формулам (3.15), являются H-сопряженными. Нам потребуются некоторые вспомогательные утверждения, выясняющие характер связи между тремя последовательностями векторов: $\{s_k\}$, $\{f'(x_k)\}$ и $\{x_k\}$.

Лемма 3.2. *Справедливо соотношение*

$$f'(x_{k+1}) = f'(x_k) + \lambda^k H s_k. \tag{3.18}$$

Доказательство получается подстановкой градиента функции $f(x)$ в точке x_k:

$$f'(x_k) = b + Hx_k,$$

в уравнение (3.17), помноженное слева на H.

Лемма 3.3. *Векторы* $f'(x_k)$ *и* $f'(x_{k+1})$ *ортогональны между собой.*

Доказательство. Подставляя в уравнение (3.18) λ^k из формулы (3.7), получим

$$f'(x_{k+1}) = f'(x_k) - \frac{(f'(x_k),\ s_k)}{(s_k,\ Hs_k)} Hs_k,$$

откуда

$$(f'(x_{k+1}),\ f'(x_k)) = (f'(x_k), f'(x_k)) - \frac{(f'(x_k),\ s_k)}{(s_k,\ Hs_k)} (f'(x_k), Hs_k). \tag{3.19}$$

Так как точка x_k получена в результате минимизации

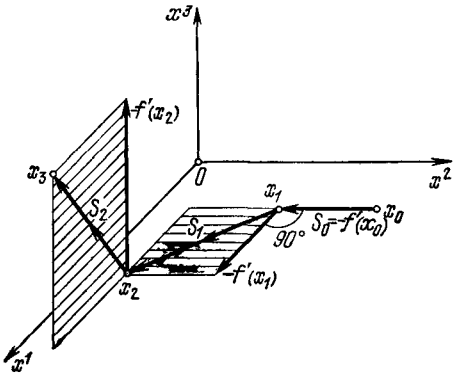


Рис. 3.2.

функции $f(x)$ в направлении s_{k-1}, векторы $f'(x_k)$ и s_{k-1} взаимно ортогональны. Поэтому

$$(f'(x_k),\ s_k) = -(f'(x_k),\ f'(x_k)) + \beta_{k-1}(f'(x_k),\ s_{k-1}) = -(f'(x_k),\ f'(x_k)).$$

Далее,

$$(s_k,\ Hs_k) = -(f'(x_k),\ Hs_k) + \beta_{k-1}(s_{k-1},\ Hs_k) = -(f'(x_k),\ Hs_k)$$

в силу H-сопряженности векторов s_{k-1}, s_k. Учитывая два последних равенства, из (3.19) получим

$$(f'(x_{k+1}),\ f'(x_k)) = 0,$$

что и требовалось доказать.

Теорема 3.1. *Векторы* $s_0, s_1, \ldots, s_k$, $k \leqslant n$, *в методе Флетчера — Ривса являются H-сопряженными, а градиенты* $f'(x_0), \ldots, f'(x_k)$ *взаимно ортогональными.*

Доказательство. Воспользуемся методом математической индукции. При $k=1$ утверждение теоремы справедливо, так как векторы s_0 и s_1 H-сопряжены в силу выбора β_0, а градиенты $f'(x_0)$ и $f'(x_1)$ ортогональны по лемме 3.3. Допустим теперь, что при $k \geqslant 2$ векторы $s_0, \ldots, s_{k-1}$ взаимно H-сопряжены, а $f'(x_0), \ldots, f'(x_{k-1})$ — взаимно ортогональны. Покажем, что при этом вектор s_k H-сопряжен всем $s_0, \ldots, s_{k-1}$, а $f'(x_k)$ ортогонален градиентам $f'(x_0), \ldots, f'(x_{k-1})$. По лемме 3.3

$$(f'(x_k),\ f'(x_{k-1}))=0.$$

При $0 \leqslant i \leqslant k-2$ имеем

$$(f'(x_k),\ f'(x_i))=(f'(x_{k-1})+\lambda^{k-1}Hs_{k-1},\ f'(x_i))=$$
$$=(f'(x_{k-1}),\ f'(x_i))+\lambda^{k-1}(Hs_{k-1},\ f'(x_i))=\lambda^{k-1}(Hs_{k-1},\ f'(x_i)),$$

так как векторы $f'(x_{k-1})$ и $f'(x_i)$, по предположению, ортогональны при $i \leqslant k-2$.

Подставляя сюда $f'(x_i)$ из (3.15), получим

$$(f'(x_k),\ f'(x_i))=\lambda^{k-1}(Hs_{k-1},\ -s_i+\beta_{i-1}s_{i-1})=0,$$

так как, по предположению, вектор s_{k-1} сопряжен всем s_i при $i \leqslant k-2$. Итак, ортогональность векторов $\{f'(x_k)\}$ доказана. Рассмотрим теперь вектор s_k. Согласно (3.15), (3.16) он H-сопряжен с s_{k-1}. При $0 \leqslant i \leqslant k-2$ получим

$$(s_k,\ Hs_i)=(-f'(x_k)+\beta_{k-1}s_{k-1},\ Hs_i)=-(f'(x_k),\ Hs_i), \quad (3.20)$$

так как, по предположению, $(s_{k-1},\ Hs_i)=0$, $i \leqslant k-2$. Из леммы 3.2 следует, что

$$Hs_i=\frac{f'(x_{i+1})-f'(x_i)}{\lambda^i}. \quad (3.21)$$

(Величина λ^i может быть равна нулю только при $f'(x_i)=0$, т. е. когда процесс вычислений окончен.) Подставляя (3.21) в (3.20), получим

$$(s_k,\ Hs_i)=-\left(f'(x_k),\ \frac{f'(x_{i+1})-f'(x_i)}{\lambda^i}\right)=0,$$

так как все градиенты взаимно ортогональны. Следовательно, вектор s_k H-сопряжен всем s_i, $i=0, 1, \ldots, k-1$. Теорема доказана.

Для того чтобы изложить в окончательном виде алгоритм Флетчера — Ривса, докажем еще одну вспомогательную лемму.

Лемма 3.4. *Справедлива формула*

$$\beta_{k-1} = \frac{(f'(x_k),\ f'(x_k))}{(f'(x_{k-1}),\ f'(x_{k-1}))}.$$

Доказательство. Из формул (3.16), (3.18) имеем

$$\beta_{k-1} = \frac{(f'(x_k),\ Hs_{k-1})}{(s_{k-1},\ Hs_{k-1})} = \frac{(f'(x_k),\ f'(x_k) - f'(x_{k-1}))}{(s_{k-1},\ f'(x_k) - f'(x_{k-1}))} =$$

$$= -\frac{(f'(x_k),\ f'(x_k))}{(s_{k-1},\ f'(x_{k-1}))} = \frac{(f'(x_k),\ f'(x_k))}{(f'(x_{k-1}),\ f'(x_{k-1}))}.$$

Итак, мы пришли к следующей процедуре минимизации квадратичной функции:

1) вычисляем в точке x_0

$$s_0 = -f'(x_0);$$

2) на k-м шаге решаем задачу минимизации по $\lambda \geqslant 0$ функции $f(x_k + \lambda s_k)$, в результате чего определяем шаг λ^k и точку

$$x_{k+1} = x_k + \lambda^k s_k;$$

3) вычисляем величины $f(x_{k+1})$ и $f'(x_{k+1})$;

4) если $f'(x_{k+1}) = 0$, то точка x_{k+1} — решение задачи; если нет — определяем s_{k+1} из соотношения

$$s_{k+1} = -f'(x_{k+1}) + \frac{(f'(x_{k+1}),\ f'(x_{k+1}))}{(f'(x_k),\ f'(x_k))} s_k$$

и переходим к следующей итерации.

Это и есть окончательный вид алгоритма Флетчера — Ривса. Как уже было сказано ранее, он найдет минимум квадратичной функции не более чем за n шагов.

4. Минимизация неквадратичных функций. Метод Флетчера — Ривса в форме 1) — 4) может применяться для минимизации и неквадратичных функций. Он является методом первого порядка и в то же время, как мы увидим ниже, скорость его сходимости квадратична. Этим метод сопряженных градиентов выгодно отличается от обычных градиентных методов (§ 1 главы II). Разумеется, если функция не квадратична, метод уже не будет конечным. Поэтому после $(n+1)$-й итерации процедура 1) — 4) циклически повторяется с заменой x_0 на x_{n+1}, а счет заканчивается при $\|f'(x_k)\| < \varepsilon$, где ε — заданное число. При минимизации неквадратичных функций обычно применяют следующую модификацию метода Флетчера —

Ривса:

$$x_{k+1} = x_k + \lambda^k s_k,$$
$$s_k = -f'(x_k) + \beta_{k-1} s_{k-1}, \quad k \geqslant 1,$$
$$s_0 = -f'(x_0), \tag{3.22}$$
$$f(x_k + \lambda^k s_k) = \min_{\lambda \geqslant 0} f(x_k + \lambda s_k),$$
$$\beta_{k-1} = \begin{cases} \dfrac{(f'(x_k),\ f'(x_k) - f'(x_{k-1}))}{(f'(x_{k-1}),\ f'(x_{k-1}))}, & k \notin I, \\ 0, & k \in I. \end{cases}$$

Здесь I множество индексов $I = \{0, n, 2n, 3n, \ldots\}$, т. е. обновление метода происходит через каждые n шагов. Модификация (3.22) отличается от оригинала формулой расчета коэффициентов β_{k-1}. Это различие существенно только в неквадратичном случае, когда градиенты $f'(x_k)$ и $f'(x_{k-1})$ уже не являются взаимно ортогональными.

5. О сходимости метода сопряженных градиентов. Отметим, прежде всего, что, поскольку каждый первый шаг процесса (3.22) после восстановления осуществляется так же, как и в методе наискорейшего спуска, при достаточно общих предположениях относительно свойств $f(x)$ метод (3.22) должен сходиться к некоторой стационарной точке x_* функции $f(x)$. Справедлива (см., например, [9]).

Теорема 3.2. *Если функция $f(x)$ ограничена снизу, ее градиент удовлетворяет условию Липшица с константой R, то в методе* (3.22)

$$\lim_{k \to \infty} \|f'(x_k)\| = 0.$$

При более жестких предположениях относительно свойств $f(x)$ можно доказать более сильную теорему.

Теорема 3.3. *Пусть $f(x)$ — трижды дифференцируемая и сильно выпуклая функция. Тогда последовательность $\{x_k\}$, построенная по методу* (3.22), *сходится к минимуму x_* функции $f(x)$, причем имеет место оценка*

$$\|x_{k+n} - x_*\| \leqslant q \|x_k - x_*\|^2$$

для всех $k \in I$, $k \geqslant N$, q, N — некоторые постоянные.

Доказательство этой теоремы приводится в [8]. Сравнивая эту теорему с теоремой 2.2 из § 2 настоящей главы (для сходимости метода Ньютона), заключаем, что n шагов

метода сопряженных градиентов примерно эквивалентны одному шагу метода Ньютона. Подчеркнем еще раз, что метод сопряженных градиентов при этом является методом первого порядка.

§ 4. Одномерный оптимальный поиск

Как мы уже убедились выше, в ряде методов отыскания экстремума функций многих переменных таких, как метод наискорейшего спуска, метод Ньютона, сопряженных градиентов и т. д., в качестве важного элемента присутствует поиск минимума функции одной переменной. От того, насколько хорошо организован такой одномерный поиск, существенно зависит успех решения всей задачи. Поэтому мы рассмотрим более подробно этот вопрос, приняв за основу

1. Унимодальность. Прежде всего, мы опишем тот класс функций, о котором будет идти речь. А именно, мы будем для определенности говорить о поиске минимума и рассматривать так называемые унимодальные функции, т. е. функции, имеющие на заданном интервале $[a, b]$ единственный минимум. Не нарушая общности, будем полагать, что $f(x)$ минимизируется на интервале $[0, 1]$. Обозначим через x_* — искомое значение, доставляющее минимум функции $f(x)$. Тогда точное определение унимодальной функции таково. Пусть $x_1 \in [0, 1]$, $x_2 \in [0, 1]$ — любые две точки такие, что $x_1 < x_2$. Тогда функция $f(x)$ унимодальна, если из условия $x_1 > x_*$ следует, что

$$f(x_1) < f(x_2),$$

а из условия $x_2 < x_*$ следует, что

$$f(x_1) > f(x_2).$$

Подчеркнем, что мы не требуем дифференцируемости и даже непрерывности $f(x)$. Наши рассмотрения будут справедливы и для функции с разрывами и изломами. Из определения также следует, что унимодальная функция не может содержать участков, где она постоянна. Простейшим примером унимодальной функции является функция, сильно выпуклая на $[0, 1]$.

Итак, предполагается, что кроме свойства унимодальности функции $f(x)$ на интервале $[0, 1]$ нам больше ничего о ней не известно. Мы можем только измерять (вычислять) значение функции для любого $x \in [0, 1]$. Измерение будем называть экспериментом. Оказывается, что свойство унимодальности позволяет по результатам любой пары экспериментов указать интервал, в котором заключено значение x_*, более узкий, чем начальный. В самом деле, возьмем два эксперимента x_1, x_2, причем $x_1 < x_2$. Возможны три различных исхода:

а) $f(x_1) > f(x_2)$;

б) $f(x_1) < f(x_2)$;

в) $f(x_1) = f(x_2)$ (см. рис. 4.1).

Легко видеть, что точка минимума x_* не может находиться в заштрихованных интервалах, и поэтому после данных экспериментов эти интервалы (в зависимости от исхода) могут быть отброшены.

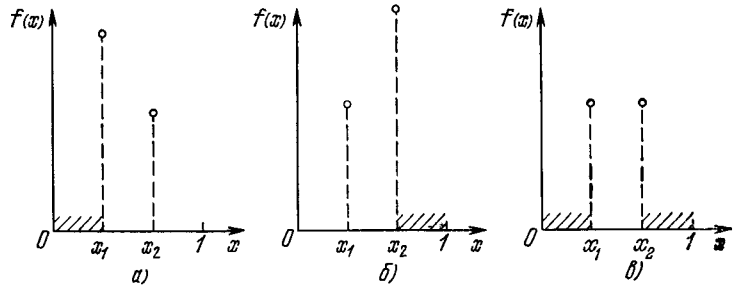


Рис. 4.1.

2. Эффективность поиска и интервал неопределенности. Интуитивно ясно, что от стратегии поиска, т. е. от правила выбора экспериментов, существенно зависит быстрота отыскания x_*. Однако так же ясно, что последняя величина зависит и от самой функции $f(x)$. Поэтому естественный вопрос об отыскании наиболее эффективной (оптимальной) стратегии поиска требует ряда уточнений.

Рассмотрим одну частную стратегию с тремя экспериментами:

$$x_1 = 0{,}1, \quad x_2 = 0{,}4, \quad x_3 = 0{,}8$$

(рис. 4.2). Мы видим, что в зависимости от исхода экспе-

рименtов длина интервала, в котором находится x_*, есть, соответственно, 0,4, 0,7 и 0,6. Этот интервал будем называть интервалом неопределенности. Дадим его выражение для серии из n экспериментов. Пусть k — индекс эксперимента x_k, при котором получено наименьшее значение $f(x)$, т. е.

$$f(x_k) = \min_{1 \leqslant j \leqslant n} f(x_j). \tag{4.1}$$

Тогда длина интервала неопределенности после n экспериментов есть

$$l_n(x_1, \ldots, x_n, k) = x_{k-1} - x_{k\ 1}. \tag{4.2}$$

Мы здесь указали зависимость интервала неопределенности от экспериментов $x_1, \ldots, x_n$ и от конкретного номера k, определяемого (4.1). Отметим, что в соответствии с записью (4.2) можно положить

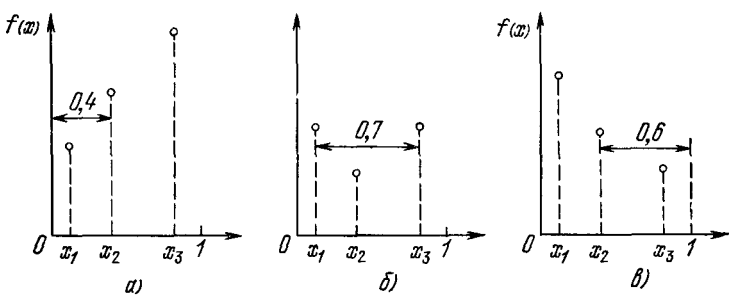


Рис. 4.2.

$$x_0 = 0, \quad x_{n+1} = 1. \tag{4.3}$$

Очевидно, предсказать величину l_n заранее нельзя, она станет известна только после завершения всех n экспериментов. Поэтому некорректно брать ее за меру эффективности поиска. Однако если мы будем интересоваться наихудшим случаем, т. е. наибольшей длиной интервала неопределенности при данной стратегии, то и реализовавшийся результат будет заведомо не хуже рассчитанного (гарантированного). Запишем этот гарантированный результат

$$L_n(x_1, \ldots, x_n) = \max_{1 \leqslant k \leqslant n} \{l_n(x_1, \ldots, x_n, k)\}. \tag{4.4}$$

В приведенном выше примере (рис. 4.2) с тремя экспериментами максимальная (гарантированная) величина интервала неопределенности равна

$$L_3(0{,}1;\ 0{,}4;\ 0{,}8) = \max\{(x_2 - x_0),\ (x_3 - x_1),\ (x_4 - x_2)\} = \\ = \max\{0{,}4;\ 0{,}7;\ 0{,}6\} = 0{,}7.$$

Очевидно, хуже этого результата при данной стратегии мы не получим (но можно получить и лучше, например, 0,6 или даже 0,4).

3. **Принцип минимакса.** Естественно теперь поставить задачу о выборе такой стратегии поиска $x_1, \ldots, x_n$, которая приводила бы к минимальному значению введенного критерия эффективности $L_n(x_1, \ldots, x_n)$ (4.4), т. е. минимизировала максимальную длину интервала неопределенности. Назовем такую минимизирующую стратегию оптимальной и обозначим ее $\bar{x}_1, \ldots, \bar{x}_n$. Тогда

$$\bar{L}_n = L_n(\bar{x}_1, \ldots, \bar{x}_n) = \inf\{L_n(x_1, \ldots, x_n)\}, \qquad (4.5)$$
$$0 \leqslant x_1 < x_2 < \ldots < x_n \leqslant 1.$$

Подчеркнем, что мы пишем inf, а не min, поскольку область изменения $x_1, \ldots, x_n$ открыта.

Комбинируя (4.4) и (4.5), имеем

$$L_n = \inf_{0 \leqslant x_1 < x_2 < \ldots < x_n \leqslant 1} \max_{1 \leqslant k \leqslant n} \{l_n(x_1, \ldots, x_n, k)\}. \qquad (4.6)$$

Принцип выбора стратегии из условия (4.6) в соответствии с терминологией теории исследования операций носит название принципа минимакса. Подчеркнем еще раз, что результат, получаемый применением минимаксной стратегии, гарантирован. Это означает, что реализация не может дать худший результат, а возможно, даст и лучший.

Примечание. Описанный подход к поиску минимума можно трактовать как «игру с природой». Стратегией экспериментатора является выбор точек эксперимента $x_1, \ldots, x_n$, а стратегией природы — выбор функции $f(x)$ (точнее, ее значений в указанных точках). Принцип минимакса соответствует расчету на наихудший случай, т. е. гипотезе, что природа является активным противником и пытается «подсунуть» экспериментатору наиболее «плохую» функцию.

4. **Пассивный поиск.** Описанный выше способ, соответствующий записи (4.6), носит название пассивного поиска,

так как все эксперименты производятся одновременно. Построим оптимальные пассивные стратегии. Начнем сразу с двух экспериментов ($n=2$), поскольку один эксперимент не позволяет уменьшить исходный интервал неопределенности $[0, 1]$.

Итак, пусть $0 \leqslant x_1 < x_2 \leqslant 1$. Тогда, согласно (4.4),

$$L_2 = \max\{(x_2 - x_0),\ (x_3 - x_1)\} = \max\{x_2,\ (1 - x_1)\}.$$

Легко показать, что

$$\bar{L}_2 = \inf_{0 \leqslant x_1 < x_2 \leqslant 1} \max\{x_2,\ (1 - x_1)\} = \frac{1}{2} + \frac{\varepsilon}{2}, \qquad (4.7)$$

причем

$$\bar{x}_1 = \frac{1}{2} - \frac{\varepsilon}{2},\ \bar{x}_2 = \frac{1}{2} - \frac{\varepsilon}{2}, \qquad (4.8)$$

где ε — любое малое положительное число. Это число можно трактовать как чувствительность экспериментатора в различении двух близких точек x_1 и x_2.

Рассмотрим теперь три эксперимента. Легко показать, что

$$\bar{L}_3 = \inf_{0 \leqslant x_1 < x_2 < x_3 \leqslant 1} \max\{x_2,\ (x_3 - x_1),\ (1 - x_2)\} = \frac{1}{2},$$

причем $x_2 = \frac{1}{2}$, а $\bar{x}_1$ и $\bar{x}_3$ — любые, удовлетворяющие условию $\bar{x}_3 - \bar{x}_1 \leqslant \frac{1}{2}$ и исходным ограничениям.

Мы видим, что прибавление третьего эксперимента улучшает результат всего на $\bar{L}_3 - \bar{L}_2 = \frac{\varepsilon}{2}$. Можно показать, что для любого четного n

$$\bar{L}_{n+1} - \bar{L}_n = \frac{\varepsilon}{\frac{n}{2} + 1},$$

т. е. использование нечетного числа экспериментов в пассивной стратегии нецелесообразно.

Для четного числа экспериментов наилучшее размещение получается разбиением их на равноотстоящие ε-пары (т. е. эксперименты в каждой из таких пар разнесены на ε). Оптимальная стратегия имеет вид

$$x_k = \frac{2\left(\left[\frac{k-1}{2}\right] + 1\right)}{n+2} - \left\{\left[\frac{k+1}{2}\right] - \left[\frac{k}{2}\right] - \frac{1}{2}\right\}\varepsilon,\ k = 1, \ldots, n, \qquad (4.9)$$

где $[z]$ обозначает наибольшее целое число, меньшее или равное z. Наилучший гарантированный результат для серии из n экспериментов (n — четное) есть

$$L_n = \frac{1+\varepsilon}{n/2+1}. \tag{4.10}$$

5. Последовательный поиск. Метод Фибоначчи. Откажемся теперь от предположения об одновременности проведения экспериментов, т. е. будем производить каждый следующий эксперимент с учетом информации, полученной в предыдущих опытах. Интуитивно ясно, что такая постановка должна привести за то же число экспериментов к гарантированной меньшей длине интервала неопределенности. Для примера начнем рассмотрение с *метода дихотомии* (половинного деления). Как указывалось выше, любой процесс поиска начинается с двух экспериментов. Выберем их, согласно (4.8). После этого у нас останется один из двух интервалов $\left[0, \frac{1}{2}+\frac{\varepsilon}{2}\right]$ или $\left[\frac{1}{2}-\frac{\varepsilon}{2}, 1\right]$. Выберем теперь третий и четвертый эксперименты как ε-пару в середине оставшегося интервала. После этого интервал неопределенности станет равным $\left(\frac{1}{4}+\frac{3}{4}\varepsilon\right)$. После пятого и шестого экспериментов, произведенных по тому же правилу, интервал сократится до $\left(\frac{1}{8}+\frac{7}{8}\varepsilon\right)$. Легко видеть, что после n экспериментов (n — четно) минимум функции лежит в интервале

$$L_n = 2^{-n/2} + (1-2^{-n/2})\,\varepsilon. \tag{4.11}$$

Сравнение (4.11) и (4.10) показывает, что метод дихотомии существенно эффективнее метода поиска однородными парами. Так, для уменьшения интервала неопределенности до 0,01, если пренебречь величиной ε, требуется 198 пассивных экспериментов и всего 14 — по методу дихотомии. Интересно поставить вопрос об отыскании оптимальной стратегии последовательного поиска. Такая задача была поставлена и решена в 1953 г. Кифером, причем неожиданно она оказалась связана с работами математика XII века Фибоначчи и его знаменитыми числами, и даже с геометрическими построениями Евклида. Итак, пусть мы намерены провести n экспериментов для отыскания минимума унимодальной функции на интервале $[0, 1]$.

Рассмотрим ситуацию, которая возникла после того, как все эксперименты, кроме последнего, уже проведены. Мы имеем некоторый интервал неопределенности длины $\tilde{L}_{n-1}$. Внутри него находится эксперимент с наименьшим из $(n-1)$ испытаний значением функции f и также внутри него следует произвести последний эксперимент. Поскольку расположение обоих экспериментов целиком зависит от нашей стратегии и мы хотим, чтобы длина последнего интервала, $\tilde{L}_n$, была наименьшей, то ситуация полностью эквивалентна поиску с единственной парой экспериментов. Тогда, как следует из вышеизложенного, оптимальным будет расположение экспериментов в точках, симметричных относительно середины интервала и удаленных от нее на $\frac{\varepsilon}{2}$ (см. рис. 4.3). Таким образом,

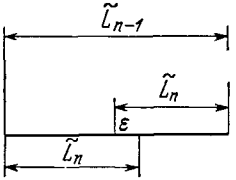


Рис. 4.3.

$$\tilde{L}_{n-1} = 2\tilde{L}_n - \varepsilon. \tag{4.12}$$

Далее, рассмотрим ситуацию, когда проведены все эксперименты, кроме двух последних, и длину имеющегося интервала неопределенности обозначим $\tilde{L}_{n-2}$. Внутри этого интервала находится точка с наименьшим из $(n-2)$ испытаний значением $f(x)$ и также внутри него следует провести следующий, $(n-1)$-й эксперимент. По результатам этого эксперимента часть интервала будет отброшена, а оставшаяся есть $\tilde{L}_{n-1}$. Поскольку заранее не ясно, какая часть будет отброшена, — это станет ясно только после эксперимента, указанные точки должны располагаться на

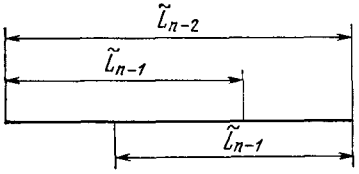


Рис. 4.4.

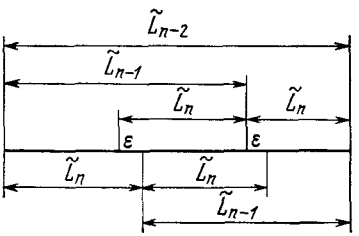


Рис 4.5.

равных расстояниях от концов интервала $L_{n\ 2}$ (см. рис. 4.4). Но одна из точек внутри интервала $\tilde{L}_{n-2}$ (рис. 4.4) останется после $(n-1)$-го эксперимента и станет точкой внутри интервала $\tilde{L}_{n-1}$ (см. рис. 4.3). Сочетание возможных комбинаций рис. 4.3 и 4.4 приводит к схеме разбиения интервала $\tilde{L}_{n-2}$, изображенной на рис. 4.5.

Таким образом, имеем

$$\tilde{L}_{n-2} = \tilde{L}_{n-1} + \tilde{L}_n. \tag{4.13}$$

Проведя аналогичные рассуждения для любого интервала $\tilde{L}_j$, получающегося после завершения j экспериментов, имеем рекуррентную формулу

$$\tilde{L}_{j-1} = \tilde{L}_j + \tilde{L}_{j+1}, \quad j = 2,\ 3,\ \ldots,\ n-1. \tag{4.14}$$

Пользуясь формулами (4.14) и (4.12), можно вычислить

$$\begin{aligned}
\tilde{L}_{n-2} &= 3\tilde{L}_n - \varepsilon,\\
\tilde{L}_{n-3} &= 5\tilde{L}_n - 2\varepsilon,\\
\tilde{L}_{n-4} &= 8\tilde{L}_n - 3\varepsilon,\\
\tilde{L}_{n-5} &= 13\tilde{L}_n - 5\varepsilon.
\end{aligned}$$

Для получения общей формулы длины $\tilde{L}_{n-k}$ введем последовательность чисел Фибоначчи F_k, определяемую следующим образом:

$$\begin{aligned}
&F_0 = F_1 = 1,\\
&F_k = F_{k-1} + F_{k-2}, \quad k = 2,\ 3,\ \ldots
\end{aligned} \tag{4.15}$$

Тогда имеем

$$\tilde{L}_{n-k} = F_{k+1}\tilde{L}_n - F_{k-1}\varepsilon. \tag{4.16}$$

Учитывая, что длина исходного интервала есть 1, получаем

$$\tilde{L}_1 = 1 = F_n\tilde{L}_n - F_{n-2}\varepsilon,$$

откуда находим выражение для длины интервала, оставшейся после проведения n последовательных экспериментов

$$\tilde{L}_n = \frac{1}{F_n} + \frac{F_{n-2}}{F_n}\varepsilon. \tag{4.17}$$

Напомним, что какова бы ни была исходная унимодальная функция $f(x)$, построенная нами оптимальная страте-

гия последовательного поиска гарантирует, что длина интервала неопределенности $\tilde{L}_n$ после n экспериментов будет не больше $\tilde{L}_n$ (4.17) Эту стратегию называют часто методом Фибоначчи.

Как мы видели, если поиск методом Фибоначчи начат, то действия на любом шаге определяются весьма просто. В каждый оставшийся интервал попадает предыдущий эксперимент и для продолжения поиска нужно провести следующий эксперимент симметрично уже имеющемуся. Для того же, чтобы начать процесс, следует вычислить интервал $\tilde{L}_2$. Мы опускаем вывод соответствующей формулы, который использует ряд свойств последовательности Фибоначчи (подробнее см. [3]). Приведем только окончательный результат

$$\tilde{L}_n = \frac{F_{n-2}}{F_n} + \frac{(-1)^n}{F_n}\varepsilon. \tag{4.18}$$

Отметим, что для получения интервала неопределенности длины 0,01 требуется всего 11 экспериментов по методу Фибоначчи.

6. Метод «золотого сечения». Согласно (4.18), расположение первого эксперимента (точнее, двух первых) зависит от общего числа экспериментов n, которые мы намерены проводить. Однако, начиная поиск минимума, мы можем и не иметь четкого представления о желаемом числе экспериментов. В то же время для метода Фибоначчи такое число требуется задать. Заметим, что метод дихотомии, например, не нуждается в этом. Процесс идет независимо от числа экспериментов n. Мы сейчас опишем еще один метод, не зависящий от числа готовящихся экспериментов и почти столь же эффективный, что и метод Фибоначчи.

Мы воспользуемся тем же соотношением (4.14):

$$\hat{L}_{j-1} = \hat{L}_j + \hat{L}_{j+1}, \quad j = 2, 3, \ldots, n-1, \tag{4.19}$$

но не будем прибегать к «начальному условию» (4.12), поскольку оно зависит от n. Вместо этого будем выдерживать постоянным (равным τ) отношение длин последовательных интервалов, т. е.

$$\frac{\hat{L}_{j-1}}{\hat{L}_j} = \frac{\hat{L}_j}{\hat{L}_{j+1}} = \tau. \tag{4.20}$$

Это условие известно, как «правило золотого сечения» (деление отрезка в данном соотношении выполнял еще Евклид с помощью циркуля и линейки). Разделив (4.19) на $\hat{L}_{j+2}$ и приняв во внимание, что, согласно (4.20), $\frac{\hat{L}_{j-1}}{\hat{L}_{j+1}} = \tau^2$, имеем

$$\tau^2 = \tau + 1. \tag{4.21}$$

Это уравнение имеет единственный положительный корень

$$\tau = \frac{1+\sqrt{5}}{2} = 1{,}618. \tag{4.22}$$

Таким образом, в методе золотого сечения начальный единичный отрезок делится по «правилу золотого сечения» (4.20), (4.22) и первые два эксперимента располагаются симметрично на расстоянии 0,618 от соответствующих концов интервала. По результатам этих экспериментов сохраняется один из интервалов, в котором расположен оставшийся эксперимент, и симметрично ему располагается следующий эксперимент и т. д. После n экспериментов имеем интервал неопределенности

$$\tilde{L}_n = \frac{1}{\tau^{n-1}}. \tag{4.23}$$

Сравним теперь метод золотого сечения с методом Фибоначчи. Для этого воспользуемся следующей формулой (см. [3]):

$$F_n = \frac{\tau^{n+1} - (-\tau)^{-(n+1)}}{\sqrt{5}}. \tag{4.24}$$

При больших n можно приближенно положить

$$F = \frac{\tau^{n+1}}{\sqrt{5}}. \tag{4.25}$$

Таким образом, для больших n имеем

$$\frac{\hat{L}_n}{\tilde{L}_n} = \frac{\tau^{n+1}}{\sqrt{5}\,\tau^{n-1}} = \frac{\tau^2}{\sqrt{5}} = 1{,}1708\ldots,$$

т. е. окончательный интервал в методе золотого сечения всего лишь на 17% больше, чем в методе Фибоначчи. Можно также показать, что при больших n оба метода начинаются практически из одной и той же точки.

7. Заключение. Методами, изложенными в этой главе, не исчерпываются все известные в настоящее время алгоритмы безусловной оптимизации Мы рассмотрели лишь наиболее распространенные из них. В частности, мы не касались методов двойственных направлений, основная идея которых состоит в построении процессов спуска, обладающих сверхлинейной скоростью сходимости и не требующих при этом вычисления вторых производных минимизируемой функции. С ними можно ознакомиться, например, в [9]. Мы оставили также в стороне многие известные методы нулевого порядка, кроме рассмотренных в §§ 2 и 4 настоящей главы, которые тоже можно найти в уже цитировавшихся книгах.

Следующая, третья, глава посвящена изучению наиболее простого класса оптимизационных задач с ограничениями — задачам линейного программирования. Методы решения этих задач являются хорошо изученными, а потому и широко распространенными. В то же время задачи линейного программирования имеют некоторые специфические черты, не свойственные другим задачам оптимизации, что побуждает выделить этот класс задач особо.

Глава III

ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

Введение

В течение последних двух десятилетий возник целый ряд разделов математики, в название которых входит слово программирование: *линейное программирование, нелинейное программирование, динамическое программирование, стохастическое программирование* и т. д. Совокупность этих дисциплин часто объединяется единым термином *математическое программирование*. Вряд ли этот термин удачен, поскольку одновременно под программированием понимают кодирование математических алгоритмов на машинном языке. Поэтому впредь он употребляться не будет. Однако термины динамическое программирование, линейное программирование и т. д. стали общепринятыми и поэтому будут сохранены.

Задачи линейного программирования были первыми подробно изученными задачами оптимизации при наличии ограничений типа неравенств. Термин «линейное программирование» возник в результате неточного перевода с английского «linear programming». Одно из значений слова «programming» — составление планов, планирование. Следовательно, правильным переводом этого термина было бы не «линейное программирование», а «линейное планирование», «планирование на основе линейных соотношений», что более точно отражает содержание дисциплины.

§ 1. О постановках задачи линейного программирования и ее приложениях

1. Основные определения. В этой главе мы будем рассматривать задачу максимизации линейной функции

$$f(x) = c^1x^1 + c^2x^2 + \ldots + c^nx^n = (c,\ x)$$

Эта функция неограничена, и поэтому искать ее максимум, не налагая никаких ограничений на область изменения

вектора x, бессмысленно. Интерес представляет задача максимизации $f(x)$ при условии, что x принадлежит некоторому множеству. Те из соответствующих задач, в которых область изменения x — многогранник, т. е. выбор значений вектора x стеснен условиями типа линейных равенств и неравенств, составляют предмет линейного программирования.

В общем случае задача линейного программирования формулируется следующим образом: найти величины $x^1, \ldots, x^n$, доставляющие максимум (минимум) линейной функции

$$f(x) = (c, x) = c^1x^1 + c^2x^2 + \ldots + c^nx^n$$

на множестве значений $x^1, \ldots, x^n$, удовлетворяющих ограничениям, в числе которых могут присутствовать только равенства и неравенства вида

$$a_{i1}x^1 + \ldots + a_{in}x^n \leqslant b^i,$$
$$a_{k1}x^1 + \ldots + a_{kn}x^n = b^k,$$
$$a_{l1}x^1 + \ldots + a_{ln}x^n \geqslant b^l.$$

Среди ограничений задач линейного программирования часто встречаются условия неотрицательности всех или части переменных:

$$x^j \geqslant 0, \quad j = 1, \ldots, p.$$

Хотя формально эти условия являются частным случаем представленных выше условий общего вида, на практике, при построении алгоритмов, их обычно выделяют в особую группу. *Целевую функцию* $f(x) = (c, x)$ принято называть *критерием* задачи линейного программирования (иногда ее называют также функционалом задачи линейного программирования). Эти термины отражают природу тех экономических задач, которые послужили источником излагаемой в этой главе математической теории. С некоторыми примерами подобных задач мы сейчас познакомимся.

2. Примеры прикладных задач линейного программирования. Задача о выборе оптимального плана. Допустим, что для создания некоторого продукта требуется m видов ресурсов и можно использовать n способов (технологий) производства. Обозначим через a_{ij} расход ресурса номера i при единичной интенсивности

использования технологии номера j *), а через c^j — количество производимой при этом продукции, и будем считать, что зависимость расходов и выпусков от интенсивностей линейна Предположим, далее, что производству выделено b^i единиц i-го ресурса, и обозначим через x^j интенсивность использования j-й технологии. Тогда под оптимизацией плана можно понимать поиск максимума обьема выпуска продукции, равного

$$\sum_{j=1}^{n} c^j x^j,$$

при заданных расходах ресурсов:

$$a_{11}x^1 + \ldots + a_{1n}x^n = b^1,$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$a_{m1}x^1 + \ldots + a_{mn}x^n = b^m.$$

Интенсивности использования технологий по смыслу неотрицательны, т. е. переменные x^j должны подчиняться еще и ограничениям

$$x^1 \geqslant 0, \ \ldots\ , \ x^n \geqslant 0.$$

Таким образом, задача оптимального распределения ресурса по заданным способам производства может быть сформулирована как задача линейного программирования.

Задача о рационе. Предположим, что нужно выбрать суточный рацион питания, имея в распоряжении n продуктов, содержащих m питательных веществ.

Пусть a_{ij} — количество j-го питательного вещества в единице i-го продукта, b^j — суточная потребность организма в j-м питательном веществе, c^i — цена единицы i-го продукта Если обозначить суточное потребление i-го продукта через x^i, стоимость рациона будет равна $c^1x^1 + \ldots + c^nx^1$. Естественно поставить задачу минимизации при условии полного удовлетворения потребности организма в питательных веществах:

$$a_{11}x^1 + \ldots + a_{1n}x^n \geqslant b^1,$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$a_{m1}x^1 + \ldots + a_{mn}x^n \geqslant b^m$$

*) Эту интенсивность можно измерять, например, долей суток, в течение которой эксплуатируется данная технология. Одним из ресурсов в таком случае будет время.

Потребление i-го продукта должно быть, разумеется, неотрицательно:

$$x^i \geqslant 0, \quad i = 1, \ldots, n.$$

Конечно, постановка такой задачи не учитывает многих важных факторов, и ее решение часто приводит к парадоксам. Например, рацион коровы, если не учитывать емкость ее желудка, должен состоять из одной соломы. Поэтому формулировка задачи о рационе для того, чтобы ее решение представляло практический интерес, должна быть уточнена.

Транспортная задача. Транспортная задача является одной из самых распространенных задач линейного программирования специального вида. Предположим, что имеется m складов, где хранится некоторый продукт в количествах $a_1, \ldots, a_m$, и n пунктов реализации этого продукта, потребности которых равны $b^1 \ldots, b^n$. Требуется найти наиболее экономичный способ перевозки продукта со складов к потребителям, если затраты на перевозку единицы продукта с i-го склада в j-й пункт потребления равны c_{ij}. Обозначим через x_{ij} количество продукта, перевозимое с i-го склада в j-й пункт. Тогда задача минимизации транспортных расходов формулируется так: найти

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

при ограничениях

$$\sum_{j=1}^{n} x_{ij} \leqslant a_i, \quad i = 1, \ldots, m,$$

$$\sum_{i=1}^{m} x_{ij} \geqslant b^j, \quad j = 1, \ldots, n,$$

$$x_{ij} \geqslant 0, \quad i = 1, \ldots, m, \; j = 1, \ldots, n.$$

Первое неравенство означает, что количество продукта, вывезенного со склада, не должно превышать запаса, второе — что потребности каждого пункта должны быть удовлетворены. Третье условие обеспечивает неотрицательность планируемых объемов перевозок.

Мы получили задачу линейного программирования, называемую транспортной и отличающуюся от рассмот-

ренных ранее двойной индексацией переменных. Специфика ограничений этой задачи позволила разработать весьма эффективные алгоритмы ее решения. Этим оправданы значительные усилия, которые нередко тратят математики на преобразование с помощью всевозможных искусственных приемов той или иной задачи в транспортную.

3. Стандартные формы представления задач линейного программирования. Существует немало различных вариантов формулировок задач линейного программирования, но наиболее употребительны две формы их записи. И к той, и к другой любую задачу можно привести с помощью простых приемов. Некоторые из таких приемов перечислены ниже.

Переход от задачи минимизации к задаче максимизации. Для этого, как мы уже видели в главе I, нужно изменить знак критериальной функции, т. е. задача минимизации функции

$$f(x) = (c,\ x)$$

эквивалентна задаче на поиск максимума функции

$$\tilde{f}(x) = -(c,\ x).$$

Переход к эквивалентной системе неравенств. Меняя знаки свободного члена и коэффициентов в ограничении — неравенстве, можно поменять знак этого неравенства на обратный. Например, ограничение

$$d^1x^1 + d^2x^2 + \ldots + d^nx^n \geqslant d$$

можно заменить следующим:

$$(-d^1)\,x^1 + \ldots (-d^n)\,x^n \leqslant -d.$$

Переход от ограничения — неравенства к равенству. Любое ограничение в форме неравенства введением дополнительной неотрицательной переменной может превратиться в ограничение — равенство. Так, к примеру, условие

$$a^1x^1 + \ldots + a^nx^n \leqslant a$$

эквивалентно двум ограничениям

$$a^1x^1 + \ldots + a^nx^n + x^{n+1} = a, \quad x^{n+1} \geqslant 0.$$

Переменные типа x^{n+1} называют *фиктивными* или *дополнительными*.

Представление ограничения-равенства парой неравенств. Ограничение

$$b^1x^1 + \ldots + b^nx^n = b$$

можно представить парой ограничений

$$b^1x^1 + \ldots + b^nx^n \leqslant b,$$
$$(-b^1)x^1 + \ldots + (-b^n)x^n \leqslant -b.$$

Переход к неотрицательным переменным Если на знак переменной x^j не наложено ограничений, можно заменить ее разностью двух неотрицательных переменных

$$x^j = v^j - w^j, \quad v^j \geqslant 0, \quad w^j \geqslant 0.$$

Переход от переменных, ограниченных снизу, к неотрицательным переменным. Пусть переменная ограничена снизу:

$$x^j \geqslant b^j.$$

Заменив ее по формуле

$$x^j = y^j + b^j,$$

перейдем к задаче, в которой фигурирует неотрицательная переменная

$$y^j \geqslant 0.$$

Используя перечисленные и подобные им несложные приемы, можно свести произвольную задачу линейного программирования к любой из двух общеупотребительных форм: канонической и с однотипными условиями.

Каноническая форма задачи линейного программирования Будем говорить, что задача линейного программирования записана в *канонической форме*, если она формулируется следующим образом. найти

$$\max (c^1x^1 + \ldots + c^nx^n) \qquad (1\ 1)$$

при ограничениях

$$a_{11}x^1+\ldots+a_{1n}x^n=b^1,$$
$$\ldots\ldots\ldots\ldots \tag{1.2}$$
$$a_{m1}x^1+\ldots+a_{mn}x^n=b^m,$$
$$x^i\geqslant 0, \qquad i=1,\ 2,\ \ldots,\ n. \tag{1.3}$$

В матричной форме эта задача записывается так:

$$\max(c,\ x), \tag{1.1'}$$
$$Ax=b, \tag{1.2'}$$
$$x\geqslant 0. \tag{1.3'}$$

Здесь $A=\|a_{ij}\|$ есть $(m\times n)$-матрица условий, ее столбцы $\{a_{1j},\ \ldots,\ a_{mj}\}^T$, $j=1,\ 2,\ \ldots,\ n$, называются векторами условий, вектор $b=\{b^1,\ \ldots,\ b^m\}^T$ — вектором правых частей, а $\{c^1,\ \ldots,\ c^n\}$ — вектором коэффициентов линейной формы. Далее всегда будем считать, что ранг матрицы A равен числу ее строк. Если это не так и задача, тем не менее, разрешима, некоторые из уравнений (1.2) должны быть линейными комбинациями остальных. Они «лишние» и их следовало бы отбросить. Точку x, удовлетворяющую ограничениям (1.2), (1.3), называют допустимым решением. Допустимое решение, на котором линейная форма (1.1) принимает максимальное значение, называется оптимальным решением или просто решением задачи (1.1)—(1.3).

Задача линейного программирования с однотипными условиями. Задача линейного программирования с однотипными условиями формулируется так: найти

$$\max(c^1x^1+\ldots+c^nx^n) \tag{1.4}$$

при ограничениях

$$a_{11}x^1+\ldots+a_{1n}x^n\leqslant b^1,$$
$$\ldots\ldots\ldots\ldots \tag{1.5}$$
$$a_{m1}x^1+\ldots+a_{mn}x^n\leqslant b^m,$$

или, в матричной форме,

$$\max(c,\ x), \quad Ax\leqslant b.$$

Эту запись иногда называют *сопряженной канонической формой* (смысл названия станет ясным из дальнейшего).

§ 2. Геометрическая интерпретация задач линейного программирования

1. Интерпретация в пространстве переменных и некоторые свойства задач линейного программирования. Двум основным формам задач линейного программирования соответствуют два вида геометрических представлений этих задач. Одно из них реализуется в пространстве переменных, другое — в пространстве условий.

В этом пункте мы рассмотрим геометрическую интерпретацию задачи с однотипными условиями в пространстве переменных.

Область допустимых решений задачи (1.4)—(1.5) образуется пересечением m множеств. Каждое из них определяется соответствующим неравенством

$$a_{i1}x^1 + \ldots + a_{in}x^n \leqslant b^i$$

и представляет собой полупространство, лежащее по одну сторону от гиперплоскости

$$a_{i1}x^1 + \ldots + a_{in}x^n = b^i.$$

Пересечение указанных полупространств является многогранником, который и будет областью допустимых решений задачи. Последний впредь будем обозначать через X.

Линии уровня минимизируемой функции

$$c^1x^1 + \ldots + c^nx^n = \text{const} \tag{2.1}$$

образуют семейство параллельных плоскостей. Вектор нормали к этим плоскостям

$$c = \{c^1, \ldots, c^n\}^T$$

определяет направление возрастания линейной формы (рис. 2.1).

Выберем из семейства (2.1) любую плоскость, пересекающую многогранник допустимых векторов X, и будем смещать ее в направлении c до такого предельного положения, когда многогранник X окажется в одном из полупространств, порождаемых нашей плоскостью, и хотя бы одна точка из X все еще будет ей принадлежать. Полученная предельная плоскость называется опорной для многогранника X, а его точки, лежащие в этой плоскости, будут решениями задачи.

Заметим, что множество точек, удовлетворяющих неравенствам (1.5), может быть пустым, ограниченным и неограниченным. В первом случае (рис. 2.2) задача не имеет

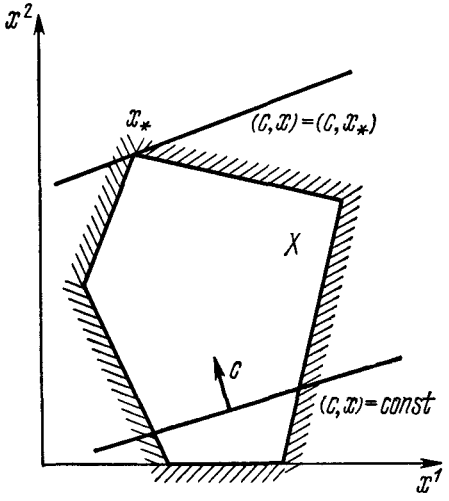


Рис. 2.1.

решения. Во втором случае она заведомо разрешима и имеет либо единственное решение (рис. 2.1), совпадающее с одной из вершин допустимого многогранника, либо бесконечное множество решений (рис. 2.3) — ребро или грань многогранника, параллельные плоскостям семейства (2.1).

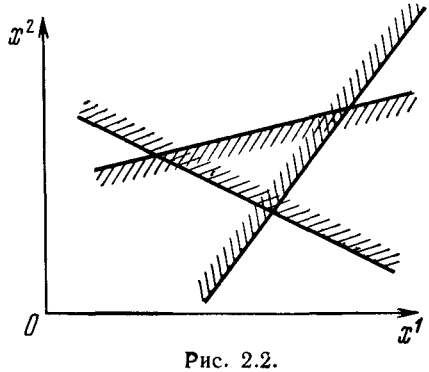


Рис. 2.2.

Если допустимое множество задачи линейного программирования неограничено, ответ на вопрос о существовании ее решения зависит от того, ограничена сверху на этом множестве целевая функция или нет. Если ограничена — задача разрешима, причем возможны те же ситуа-

ции, что и во втором из рассмотренных выше случаев. Если нет — решение в обычном понимании отсутствует (рис. 2.4).

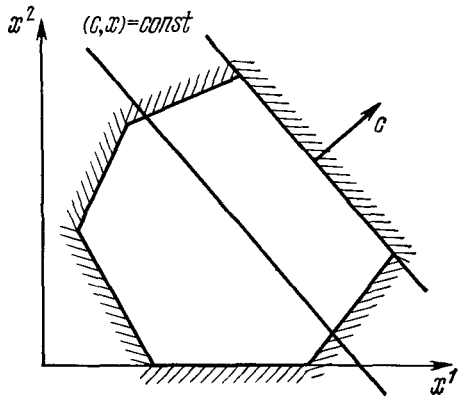


Рис. 2.3.

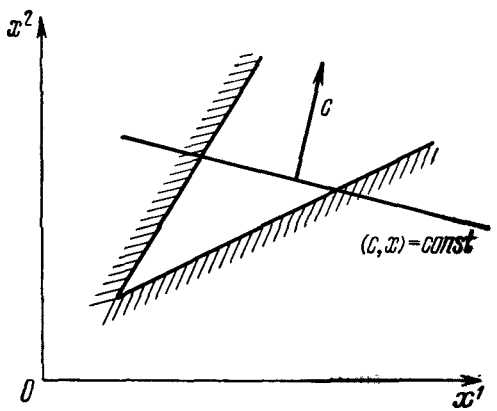


Рис. 2 4.

2. Геометрическая интерпретация задачи линейного программирования в канонической форме. Введем для задачи (1.1)—(1.3) дополнительную переменную

$$x^0 = c^1x^1 + \ldots + c^nx^n$$

и запишем ее в следующем виде:

$$\max x^0, \tag{2.2}$$

$$\begin{aligned} c^1x^1 + \ldots + c^nx^n &= x^0, \\ a_{11}x^1 + \ldots + a_{1n}x^n &= b^1, \\ \ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{m1}x^1 + \ldots + a_{mn}x^n &= b^m, \end{aligned} \tag{2.3}$$

$$x^1 \geqslant 0, \quad x^2 \geqslant 0, \quad \ldots, \quad x^n \geqslant 0. \tag{2.4}$$

Правая часть уравнений (2.3) представляет собой вектор

$$\tilde{b} = \{x^0, b^1, \ldots, b^m\}^T$$

с m фиксированными компонентами $b^1, \ldots, b^m$ и одной переменной компонентой x^0. Левая часть является линейной комбинацией расширенных векторов условий

$$\tilde{a}_j = \{c^j, a_{1j}, \ldots, a_{mj}\}^T, \qquad j = 1, \ldots, n,$$

с неотрицательными коэффициентами x^j. Мы будем рассматривать далее множество всевозможных комбинаций такого сорта, т. е. множество $(m+1)$-мерных векторов

$$u = \{u^0, u^1, \ldots, u^m\}^T,$$

компоненты которых определены соотношениями

$$\begin{aligned} u^0 &= c^1x^1 + \ldots + c^nx^n, \\ u^1 &= a_{11}x^1 + \ldots + a_{1n}x^n, \\ \ldots\ldots\ldots\ldots\ldots\ldots & \\ u^m &= a_{m1}x^1 + \ldots + a_{mn}x^n, \\ x^1 \geqslant 0, \quad x^2 \geqslant 0, \quad \ldots, \quad x^n &\geqslant 0. \end{aligned} \tag{2.5}$$

На рис. 2.5 это множество показано для случая, когда $m = 2$, т. е. задача имеет два ограничения типа равенства.

Неотрицательные линейные комбинации расширенных векторов-условий образуют многогранный конус K, ребрами которого будут векторы $\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_5$. Проекции их на плоскость $u^0 = 0$ есть, соответственно, $a_1, \ldots, a_5$. Длина перпендикуляра, соединяющего конец вектора $\tilde{a}_j$ с концом его проекции a_j, равна коэффициенту линейной формы c^j.

При произвольном значении x^0 вектор $\tilde{b}$ указывает в некоторую точку, лежащую на вертикальной прямой Q

(см. рис. 2.5), проходящей через конец вектора

$$b = \{b^1, \ldots, b^m\}^T$$

на плоскости $u^0 = 0$ (точку с координатами $\{0, b^1, \ldots, b^m\}^T$). Если же x^0 — компонента какого-нибудь допустимого решения задачи (2.2)—(2.4), вектор $\tilde{b}$ будет принадлежать также конусу K. Ясно и то, что если вектор $\tilde{b}$ указывает

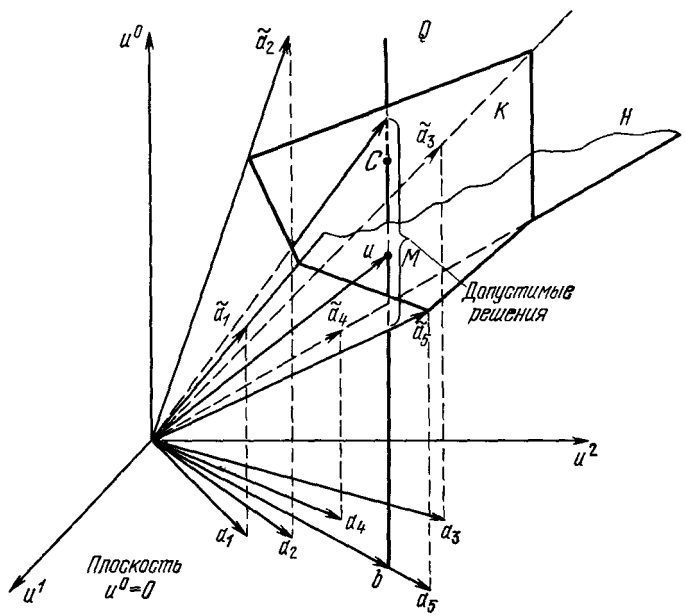


Рис. 2.5.

в точку пересечения конуса K и прямой Q, соответствующее значение x^0 будет компонентой некоторого допустимого решения. Таким образом, допустимое множество задачи (2.2)—(2.4) отражается на отрезок прямой Q, принадлежащей конусу K. Если прямая Q проходит вне этого конуса, задача не имеет допустимых решений. Когда конус K содержит ось u^0, значение критерия на допустимом множестве не ограничено сверху (рис. 2.6).

Значения целевой функции задачи (2.2)—(2.4) на любом из ее допустимых решений, соответствующих определен-

ному вектору $\tilde{b}$, совпадают и равны его первой координате. Оптимальные решения (их может быть целое множество) являются прообразами верхней точки пересечения прямой Q с конусом K (точка C на рис. 2.5). Коэффициенты произвольного разложения вектора $\tilde{b}$, указывающего в эту точку, в неотрицательную линейную комбинацию расширенных векторов условий $\tilde{a}_j$ будут компонентами одного из оптимальных решений. Ясно (в частности, из рис. 2.5), что среди них ненулевыми могут быть лишь компоненты, которым отвечают векторы $\tilde{a}_j$, принадлежащие той же грани конуса K, что и точка C (на рис. 2.5 это ребра $\tilde{a}_2$, $\tilde{a}_3$). Если число таких векторов не превосходит m и соответствующие им a_j линейно независимы, задача имеет единственное решение. В противном случае будет существовать континуум решений.

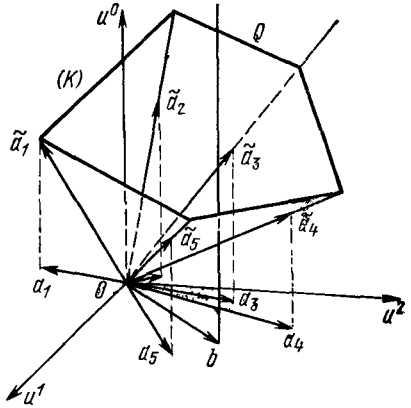


Рис. 2.6.

В дальнейшем нам понадобится специфическое понятие *невырожденности*, используемое в линейном программировании. Мы будем говорить, что задача (2.2)—(2.4) *не вырождена*, если вектор b нельзя разложить в неотрицательную линейную комбинацию менее чем m векторов a_j.

На основе представленной выше интерпретации можно наметить путь решения задачи линейного программирования (2.2)—(2.4). Построим некоторый допустимый вектор такой задачи, все компоненты которого, за исключением $x^{s_1}, \ldots, x^{s_m}$, равны нулю, причем выберем индексы $s_1, \ldots, s_m$ так, чтобы векторы условий a_{s_i}, $i = 1, \ldots, m$, были линейно независимы. Всевозможные линейные комбинации соответствующих им расширенных векторов условий $\tilde{a}_{s_1}, \ldots, \tilde{a}_{s_m}$ образуют в $(m+1)$-мерном пространстве гиперплоскость H. Первая координата ее точки пересечения с прямой Q равна значению целевой функции на рассмат-

риваемом допустимом решении (на рис. 2.5 это решение имеет ненулевые компоненты x^1, x^4, гиперплоскость H натянута на векторы $\tilde{a}_1$, $\tilde{a}_4$ и пересекается с прямой Q в точке M).

По предположению, векторы a_{s_i} линейно независимы, т. е. образуют в m-мерном пространстве базис. Разложим остальные векторы условий по этому базису:

$$a_k = a_{s_1} z_{1k} + \ldots + a_{s_m} z_{mk} = \sum_{i=1}^{m} a_{s_i} z_{ik}. \qquad (2.6)$$

Этому разложению в $(m+1)$-мерном пространстве соответствуют точки

$$\tilde{a}_k^0 = \tilde{a}_{s_1} z_{1k} + \ldots + \tilde{a}_{s_m} z_{mk} = \sum_{i=1}^{m} \tilde{a}_{s_i} z_{ik}, \qquad (2.7)$$

принадлежащие плоскости H. Расстояние от точки $\tilde{a}_k^0$ до плоскости $u^0 = 0$ определяется ее первой координатой и, согласно формуле (2.7), равно

$$c_k^0 = c^{s_1} z_{1k} + \ldots + c^{s_m} z_{mk}.$$

Если все разности $c_k^0 - c^k$ неотрицательны, конус K «лежит под» гиперплоскостью H и наше допустимое решение оптимально. Если же для какого-нибудь индекса k разность $c_k^0 - c^k$, носящая название *оценки замещения*, меньше нуля, точка $\tilde{a}_k$ окажется «выше» плоскости H. Отсюда следует, что если точка пересечения прямой Q и плоскости, натянутой на вектор $\tilde{a}_k$ и какие-нибудь $(m-1)$ векторов a_{s_i}, не выйдет за пределы конуса K, то она будет лежать не ниже прежней точки пересечения Таким образом, построив допустимое решение с m, вообще говоря, отличными от нуля компонентами, среди которых будет присутствовать x^k и отсутствовать одна из старых компонент, мы тем самым, возможно, увеличим значение целевой функции. Так, для задачи, изображенной на рис. 2.5, имея допустимое решение с ненулевыми компонентами x^1, x^4, можно перейти к другому, в котором отличны от нуля x^2 или x^3. Точка пересечения прямой Q с плоскостью, натянутой, например, на векторы $\tilde{a}_2$, $\tilde{a}_4$, будет лежать между точками M и C, т. е значение критерия при переходе к допустимому решению с ненулевыми x^2, x^4 увеличится.

В описанной схеме возрастание критерия на каждой итерации гарантируется только, когда задача не вырождена. В противном случае, при формальном переходе от одного индекса s_i к другому допустимая точка и, соответственно, критерий могут остаться неизменными. Это произойдет, например, в задаче, изображенной на рис. 2.5, если перейти от $s_1=2$, $s_2=5$ к $s_1=4$, $s_2=5$. Вопросы, связанные с вырожденностью, подробнее будут рассмотрены ниже, когда мы перейдем к обсуждению конкретной реализации указанной схемы — так называемого симплекс-метода решения задач линейного программирования. Однако, прежде чем заняться им, необходимо рассмотреть некоторые общие свойства этих задач.

§ 3. Некоторые свойства задач линейного программирования

Все сказанное в этом параграфе будет относиться к задаче линейного программирования в канонической форме, т. е. к задаче вида

$$\begin{aligned} &\max (c^1x^1+\ldots+c^nx^n),\\ &a_{11}x^1+\ldots+a_{1n}x^n=b^1,\\ &\ldots\ldots\ldots\ldots\\ &a_{m1}x^1+\ldots+a_{mn}x^n=b^n,\\ &x^1\geqslant 0,\ \ldots,\ x^n\geqslant 0. \end{aligned} \tag{3.1}$$

1. Выпуклость множества допустимых решений.

Определение 3.1. Некоторое множество X называют *выпуклым*, если наряду с любыми двумя своими точками оно содержит соединяющий их отрезок, т. е. при любых x, $y\in X$, $0\leqslant\alpha\leqslant 1$, точка $y+\alpha(x-y)$ принадлежит X. Вводя число $\beta=1-\alpha$, это определение можно перефразировать так: множество X выпукло, если для любых x, $y\in X$ и любых α, β таких, что $\alpha\geqslant 0$, $\beta\geqslant 0$, $\alpha+\beta=1$, точка $\alpha x+\beta y$ принадлежит X. Пример выпуклого множества показан на рис. 3.1. Справедлива **следующая**

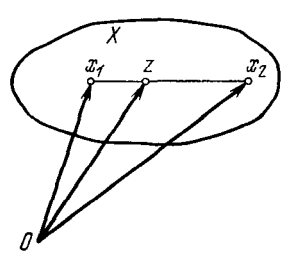


Рис. 3.1.

Теорема 3.1. *Множество X допустимых решений задачи линейного программирования выпукло.*

Доказательство. Пусть x, y — произвольные допустимые решения задачи (3.1). Тогда из

$$Ax=b, \quad x\geqslant 0, \qquad Ay=b, \quad y\geqslant 0,$$
$$\alpha\geqslant 0, \quad \beta\geqslant 0, \quad \alpha+\beta=1$$

следует, что

$$A(\alpha x+\beta y)=\alpha Ax+\beta Ay=(\alpha+\beta)\,b=b,$$
$$\alpha x+\beta y\geqslant 0.$$

Таким образом, точка $z=\alpha x+\beta y$ удовлетворяет тем же ограничениям, что и точки x, y, т. е. является допустимой. Теорема доказана.

Итак, мы показали, что допустимое множество задачи (3.1) — выпуклый многогранник. Среди всех его точек для нас особый интерес будут представлять так называемые крайние точки.

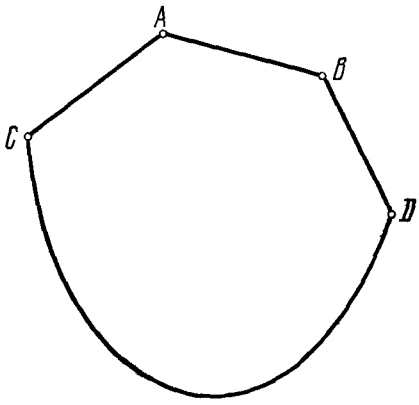


Рис. 3.2.

Определение 3.2. Точка z называется *крайней точкой* выпуклого множества X, если не существует векторов x, $y\in X$, $x\neq y$, таких, что

$$z=\alpha x+\beta y,$$

где α, β — некоторые числа, удовлетворяющие условиям

$$\alpha+\beta=1, \quad \alpha>0, \quad \beta>0.$$

Иными словами, точка z называется крайней точкой выпуклого множества X, если она не принадлежит внутренности ни одного из отрезков, целиком лежащих в этом множестве. На рис. 3.2 крайними будут точки A, B и все точки дуги C, D. Крайние точки выпуклого многогранника принято называть вершинами. Мы тоже будем использовать этот термин.

Интерес к вершинам допустимого многогранника задачи линейного программирования объясняется тем, что, как будет установлено ниже, среди ее оптимальных решений

обязательно найдется хотя бы одна вершина Последнее, в силу ограниченности числа вершин, позволяет строить конечные алгоритмы линейной оптимизации.

2. Существование базисных оптимальных решений. В линейном программировании принят термин *базисное допустимое решение задачи* (3.1). Так называют точку x, удовлетворяющую ограничениям этой задачи и обладающую тем свойством, что векторы условий, соответствующие ее положительным координатам, линейно независимы. (Само собой, таких координат может быть не больше m.) Покажем, что базисные допустимые решения и вершины допустимого многогранника суть одно и то же. Этот факт устанавливает следующая

Теорема 3.2. *Для того чтобы точка x была вершиной множества допустимых решений задачи* (3.1), *необходимо и достаточно, чтобы векторы условий, отвечающие ее положительным координатам, были линейно независимы.*

Доказательство. И необходимость и достаточность будем доказывать от противного. Начнем с необходимости. Пусть x — вершина многогранника (3.1), не являющаяся базисным допустимым решением. Не умаляя общности, будем считать, что ненулевыми являются ее первые s координат. Тогда

$$a_1x^1+\ldots+a_sx^s=b,$$

и векторы $a_1, \ldots, a_s$ линейно зависимы Последнее означает, что можно подобрать s чисел $z^1, \ldots, z^s$, среди которых есть ненулевые, так, что

$$a_1z^1+\ldots+a_sz^s=0.$$

Ясно, что при этом для любого ε выполнено равенство

$$a_1(x^1\pm\varepsilon z^1)+\ldots+a_s(x^s\pm\varepsilon z^s)=b.$$

Если же взять ε достаточно малым по модулю, кроме того, будет

$$x^1\pm\varepsilon z^1\geqslant 0, \ \ldots, \ x^s\pm\varepsilon z^s\geqslant 0.$$

Таким образом, при малых ε точки $x+\varepsilon z$, $x-\varepsilon z$, где

$$z=\{z^1, z^2, \ldots, z^s, 0, 0, \ldots, 0\}^T,$$

допустимы, причем

$$x=\frac{1}{2}(x+\varepsilon z)+\frac{1}{2}(x-\varepsilon z).$$

Последнее противоречит определению точки x как вершины допустимого многогранника. Необходимость доказана.

Докажем достаточность. Пусть точка x — допустимое базисное решение, но при этом найдутся отличные от x допустимые точки y и z такие, что при некоторых $\alpha > 0$, $\beta > 0$, $\alpha + \beta = 1$ имеем

$$x = \alpha y + \beta z.$$

Из этого равенства следует, что у точек y и z ненулевыми будут те же самые координаты, что и у точки x. Считая, как и прежде, что это — первые s координат, получим

$$a_1 y^1 + \ldots + a_s y^s = b,$$
$$a_1 z^1 + \ldots + a_s z^s = b,$$

откуда

$$a_1 (y^1 - z^1) + \ldots + a_s (y^s - z^s) = 0.$$

Так как среди чисел $y^1 - z^1, \ldots, y^s - z^s$ есть ненулевые, это равенство означает, что векторы $a_1, \ldots, a_s$ линейно зависимы, что противоречит определению точки x. Полученное противоречие завершает доказательство теоремы.

Теперь мы можем доказать основное утверждение данного параграфа.

Теорема 3.3. *Среди оптимальных решений любой задачи* (3.1) *всегда найдется хотя бы одна вершина допустимого многогранника.*

Доказательство. Допустим, что теорема не верна и мы построили такую задачу (3.1), среди решений которой нет вершин. Возьмем какое-нибудь из этих решений $\hat{x}$ В силу предыдущей теоремы векторы условий, соответствующие его ненулевым компонентам, линейно зависимы. Стало быть (см. доказательство теоремы 3.2), можно построить ненулевой вектор z такой, что его компоненты, индексы которых совпадают с индексами нулевых компонент вектора $\hat{x}$, будут нулевыми, причем

$$Az = 0$$

Отсюда следует, что при малых по модулю $\rho > 0$ векторы $\hat{x} + \rho z, \hat{x} - \rho z$ будут допустимы. При этом, так как $\hat{x}$ — решение задачи, имеем

$$(c, \hat{x} + \rho z) \leqslant (c, \hat{x}),$$
$$(c, \hat{x} - \rho z) \leqslant (c, \hat{x}),$$

что возможно только при соблюдении равенства

$$(c,\ z) = 0.$$

Последнее означает, что при тех ρ, для которых точки $\hat{x} + \rho z$ или $\hat{x} - \rho z$ допустимы, они будут и оптимальными

У вектора z есть хотя бы одна ненулевая компонента z^q. Без ограничения общности можно считать, что она отрицательна. Это значит, что, двигаясь из точки $\hat{x}$ в направлении z и не желая покидать допустимое множество, мы сможем сделать шаг, не превышающий $x^q/|z^q|$. (При таком шаге q-я координата точки $x^q + \rho z^q$ станет нулевой, а при бо́льшем — отрицательной.) Следовательно, при движении вдоль z одна из координат точки $\hat{x} + \rho z$, соответствующая положительной координате точки $\hat{x}$, в какой-то момент станет нулевой. Точка, в которой это произойдет, в силу сказанного ранее будет решением задачи. При этом у нее будет на одну положительную координату меньше, чем у $\hat{x}$. Применяя к ней все прежние рассуждения, мы придем к решению, число положительных координат которого меньше, чем у $\hat{x}$, на две и т. д. Если предположение о существовании задач (3.1), среди решений которых нет вершин, справедливо, процесс будет бесконечным. Однако это невозможно, так как у $\hat{x}$ есть только конечное число положительных координат. Следовательно, упомянутых задач не существует. Теорема доказана.

Итак, мы показали, что хотя бы одно из решений задачи (3.1) будет базисным, т. е. — вершиной допустимого многогранника, а еще точнее — точкой, число положительных координат которой не превосходит m и соответствующие этим координатам векторы условий линейно независимы. Коль скоро рассматриваемая задача не вырождена, у каждой вершины будет ровно m положительных координат. В противном случае их число может оказаться меньше m. Однако, если ранг матрицы условий равен m (а мы всегда предполагаем это), и здесь с каждой вершиной можно связать набор (причем, вообще говоря, неоднозначно) из m индексов таких, что столбцы матрицы A с этими индексами линейно независимы и координаты вершины с номерами, не вошедшими в этот набор, равны нулю. Для этого, возможно, придется пополнить список

номеров положительных координат вершины одним или несколькими номерами ее нулевых координат.

Сказанное приводит к новому, эквивалентному предыдущему, определению допустимого базисного решения: допустимое решение задачи (3.1) называется базисным, если все его координаты удается разбить на две группы так, чтобы в первую попало ровно m координат и соответствующие им векторы условий были линейно независимы, а вторая состояла бы только из нулей. Список координат первой группы принято называть допустимым базисом, а сами эти координаты и отвечающие им векторы условий — базисными. Координаты, относящиеся ко второй группе, и их векторы условий называют небазисными. Если все координаты, составляющие допустимый базис, положительны, говорят, что он не вырожден. Такими будут все допустимые базисы невырожденной задачи. При этом между ними и вершинами допустимого многогранника существует взаимно однозначное соответствие. Если же задача вырождена, найдутся вершины, которым отвечают по несколько допустимых базисов.

Нетрудно понять, что у задачи (3.1) может быть не более чем C_n^m допустимых базисных решений. Среди них есть оптимальное решение этой задачи. Чтобы найти его, достаточно:

а) отыскать решения всевозможных систем из m уравнений с m неизвестными, каждая из которых получается, если в ограничениях — равенствах задачи (3.1) зафиксировать на нуле $n-m$ некоторых координат вектора x;

б) выделить из них те, компоненты которых неотрицательны;

в) выбрать из выделенных решений те, которым соответствует максимум критерия.

Все это требует конечного числа арифметических операций и может быть названо конечным алгоритмом решения задач линейного программирования.

Однако всерьез говорить о таком алгоритме не приходится, поскольку величина C_n^m становится астрономически большой даже при весьма скромных значениях n и m. Здесь нужен упорядоченный перебор допустимых базисных решений Как организовать такой перебор — показано ниже.

§ 4. Симплекс-метод

В этом параграфе рассмотрены некоторые способы реализации общей схемы решения задач вида

$$\begin{aligned} &\max (c^1x^1 + \ldots + c^nx^n), \\ &a_{11}x^1 + \ldots + a_{1n}x^n = b^1, \\ &\ldots\ldots\ldots\ldots \\ &a_{m1}x^1 + \ldots + a_{mn}x^n = b^m, \\ &x^1 \geqslant 0, \ \ldots, \ x^n \geqslant 0. \end{aligned} \tag{4.1}$$

В основе данной схемы, именуемой симплекс-методом, лежит идея, упорядоченного перебора вершин допустимого многогранника. Мы начнем с простейшей версии симплекс-метода.

1. Алгоритм с использованием симплекс-таблиц. Пусть поставлена *невырожденная* задача (4.1) и известно одно из ее допустимых базисных решений $\tilde{x}$. Среди координат $\tilde{x}$ будет ровно m положительных (мы договорились называть их базисными). Не ограничивая общности, можно считать, что это — первые m координат. Тогда столбцы $a_1, a_2, \ldots, a_m$ линейно независимы (по определению базиса) и, соответственно, уравнения задачи (4.1) разрешимы относительно базисных переменных $x^1, \ldots, x^m$. Выражения последних через остальные переменные должны быть линейны, т. е. выглядят так:

$$x^i = \mu^i - \sum_{k=m+1}^{n} z_{ik}x^k, \quad i = 1, \ \ldots, \ m,$$

где μ^i, z_{ik} — некоторые параметры. Поскольку эти равенства должны выполняться и для $x^i = \tilde{x}^i$, $i = 1, \ \ldots, \ m$, а небазисные координаты $\tilde{x}^i$, $i = m+1, \ \ldots, \ n$, равны нулю, окончательно получим

$$x^i = \tilde{x}^i - \sum_{k=m+1}^{n} z_{ik}x^k, \quad i = 1, \ \ldots, \ m. \tag{4.2}$$

Параметры z_{ik} в этом выражении можно вычислить, например, методом исключения Гаусса. Они являются коэффициентами разложения небазисных столбцов по базисным.

В силу (4.2) значение критерия задачи (4.1) в любой точке x, удовлетворяющей ее ограничениям —

равенствам, равно

$$(c,\ x) = \sum_{i=1}^{m} c^i \tilde{x}^i - \sum_{i=1}^{m} c^i \sum_{k=m+1}^{n} z_{ik} x^k + \sum_{k=m+1}^{n} c^k x^k =$$

$$= \sum_{i=1}^{m} c^i \tilde{x}^i - \sum_{k=m+1}^{n} \left(\sum_{i=1}^{m} c^i z_{ik} - c^k \right) x^k. \qquad (4.3)$$

Величины

$$\sigma^k = \sum_{i=1}^{m} c^i z_{ik} - c^k, \quad k = m+1, \ \ldots, \ n,$$

называют оценками замещения, а z_{ik} — коэффициентами замещения (смысл этих терминов станет ясен чуть ниже). И те, и другие, а также значения базисных координат точки $\tilde{x}$ и критерия в ней сводят в так называемую симплекс-таблицу:

				столбец s		столбец k		столбец l	
	$\tilde{x}^0$	0	0 ...	0	...	σ^k	...	σ^l	... σ^n
	$\tilde{x}^1$	1	0 ...	0	...	z_{1k}	...	z_{1l}	... z_{1n}
	$\tilde{x}^2$	0	1 ...	0	...	z_{2k}	...	z_{2l}	... z_{2n}
	⋮	⋮	⋮	⋮		⋮		⋮	⋮
строка i	$\tilde{x}^i$	0	0 ...	0	...	z_{ik}	...	z_{il}	... z_{in}
	⋮	⋮	⋮	⋮		⋮		⋮	⋮
строка s	$\tilde{x}^s$	0	0 ...	1	...	z_{sk}	...	z_{sl}	... z_{sn}
	⋮	⋮	⋮	⋮		⋮		⋮	⋮
	$\tilde{x}^m$	0	0 ...	0	...	z_{mk}	...	z_{ml}	... z_{mn}

(4.4)

Она образована расширением матрицы системы уравнений (4.2). Слева к ней приписан столбец с компонентами $\tilde{x}^i$, $i = 1, \ \ldots, \ m$; сверху — оценки замещения и, наконец, в верхнем левом углу стоит значение критерия в точке $\tilde{x}$. Столбцы, соответствующие базисным координатам, в рассматриваемом случае составляют единичную подматрицу симплекс-таблицы. Вообще же, они бывают расположены в ней произвольным образом.

При анализе симплекс-таблицы могут встретиться три случая:

Случай 1. Все оценки замещения σ^k неотрицательны. Тогда $\tilde{x}$ — решение задачи (4.1). Действительно, при $\sigma^k \geqslant 0$, $k = m+1, \ \ldots, \ n$, из равенства (4.3) следует, что

$$(c,\ x) \leqslant (c,\ \tilde{x}) \qquad (4.5)$$

для любой точки x, удовлетворяющей ограничениям — равенствам задачи (4.1) и имеющей неотрицательные координаты $x^{m+1}, \ldots, x^n$. Множество таких точек включает допустимый многогранник задачи (4.1) и поэтому неравенство (4.5) заведомо будет выполняться для любого допустимого решения. Это и означает, что точка $\tilde{x}$ оптимальна.

Случай 2. Для некоторого k оценка замещения σ^k отрицательна и среди коэффициентов замещения z_{ik}, $i = 1, \ldots, m$, нет ни одного положительного. Тогда у задачи (4.1) нет конечного решения. Чтобы убедиться в этом, рассмотрим точки $x(\rho)$ с координатами

$$\begin{aligned} &x^i(\rho) = \tilde{x}^i - z_{ik}\rho, \quad i = 1, \ldots, m, \\ &x^k(\rho) = \rho, \\ &x^i(\rho) = 0, \quad i = m+1, \ldots, k-1, k+1, \ldots, n. \end{aligned} \tag{4.6}$$

При любом ρ точка $x(\rho)$ является решением уравнений (4.2) и, следовательно, удовлетворяет ограничениям — равенствам задачи (4.1). Если $\rho \geqslant 0$, в силу неположительности z_{ik} имеем $x(\rho) \geqslant \tilde{x} \geqslant 0$. Таким образом, при любом $\rho \geqslant 0$ точка $x(\rho)$ допустима, а из (4.3) следует, что

$$(c,\ x(\rho)) = \sum_{i=1}^{m} c^i \tilde{x}^i - \rho\sigma^k. \tag{4.7}$$

Поскольку оценка σ^k, по предположению, отрицательна, отсюда ясно, что, выбирая соответствующие значения $\rho > 0$, можно, не выходя из допустимого многогранника, получить сколь угодно большие значения критерия.

Случай 3. Среди оценок замещения есть отрицательные и для каждого $\sigma^k < 0$ существует $z_{ik} > 0$. Тогда нетрудно найти допустимое базисное решение со значением критерия большим, чем $(c, \tilde{x})$. В качестве такового можно взять точку $x(\rho)$, вычисленную по формулам (4.6) при любом k, для которого $\sigma^k > 0$, полагая

$$\bar{\rho} = \min_{\{i \cdot z_{ik} > 0\}} \frac{\tilde{x}^i}{z_{ik}} > 0. \tag{4.8}$$

Здесь $\bar{\rho}$ — максимальное для выбранного k положительное значение величины ρ в (4.6), при котором координаты

$x^i(\rho)$, $i=1, \ldots, m$, неотрицательны. Для построенной данным способом точки $x(\bar{\rho})$ справедливо неравенство

$$(c,\ x(\bar{\rho})) = \sum_{i=1}^{m} c^i \tilde{x}^i - \bar{\rho}\sigma^k > (c,\ \tilde{x}).$$

То, что $x(\bar{\rho})$ будет допустимым решением, мы уже установили, разбирая случай 2. То, что это допустимое решение будет базисным, т. е. столбцы матрицы условий, соответствующие его неположительным координатам, будут линейно независимы, следует из неравенства нулю величины z_{sk}, где

$$\tilde{x}^s > 0, \quad \bar{\rho} = \frac{\tilde{x}^s}{z_{sk}}, \quad x^s(\bar{\rho}) = 0.$$

Эта величина, как уже было сказано ранее, есть коэффициент при a_s в разложении по a_i, $i=1,\ \ , m$, столбца a_k и должна была бы быть нулем, окажись векторы условий

$$a_1,\ a_2,\ \ldots,\ a_{s\ 1},\ a_{s+1},\ \ldots,\ a_m,\ a_k$$

линейно зависимыми.

Итак, имея симплекс-таблицу для некоторого допустимого базисного решения задачи (4.1), легко установить, является ли оно оптимальным, и если нет, то вычислить координаты нового базисного допустимого решения с большим значением критерия. Осталось указать способ определения компонент симплекс-таблицы в новой точке — и алгоритм решения задачи (4.1) построен. Основу данной таблицы составляет матрица системы уравнений, получающейся, если разрешить условия-равенства в (4.1) относительно новых базисных переменных. Это можно сделать методом исключения Гаусса, исходя как из самих равенств (4.1), так и из имеющихся, эквивалентных им, равенств (4.2), а попросту говоря — из старой симплекс-таблицы. Второй путь значительно более экономен. Он сводится к последовательному умножению на определенные коэффициенты строки старой таблицы, связывающей выводимую из базиса переменную с небазисными, и вычитанию результатов этих умножений из остальных строк. Цель указанных операций — добиться, чтобы в столбце, соответствующем вводимой в базис переменной, отличалась от нуля и была равной единице только одна компонента — та, которая принадлежит упомянутой выше строке. Тем

самым и будет получена новая симплекс-таблица (включая новые значения базисных координат, оценок замещения и критерия)

Суммируя сказанное выше, можно предложить алгоритм перебора допустимых базисных решений невырожденной задачи (4.1), на каждом шаге которого выполняются следующие операции:

1) Определение ведущих столбца и строки. Просматриваются оценки замещения σ^k. Если все они неотрицательны, текущая вершина оптимальна. В противном случае по какому-либо признаку (обычно по признаку минимальности σ^k) выбирается *ведущий столбец* k с оценкой замещения $\sigma^k < 0$. Далее просматриваются значения коэффициентов замещения z_{ik}, $i = 1, \ldots, m$. Если среди них нет положительных, задача не имеет конечного решения В противном случае перебором тех значений индекса i, для которых $z_{ik} > 0$, определяется, причем единственным образом, номер s такой, что

$$\frac{\tilde{x}^{\tilde{l}_s}}{z_{sk}} = \min_{\{i \mid z_{ik}>0\}} \frac{\tilde{x}^{\tilde{l}_i}}{z_{ik}}, \tag{4.9}$$

где $\tilde{l}_s$, $\tilde{l}_i$ — номера базисных координат. Теперь нужно перейти в вершину, соответствующую новому допустимому базису, в котором есть k-я переменная и нет переменной с номером $\tilde{l}_s$. Это осуществляется при выполнении второй группы операций.

2) Пересчет симплекс-таблицы. Определяется новая s-я строка, равная частному от деления старой на *ведущий элемент* z_{sk}. Для каждого $i \neq s$ вычисляется новая i-я строка — разность старой и результата умножения z_{ik} на новую s-ю строку. Короче говоря, формулы расчета элементов $\bar{z}_{ik}$ новой симплекс-таблицы выглядят так:

$$\bar{z}_{ij} = \begin{cases} z_{ij} - \dfrac{z_{sj} z_{ik}}{z_{sk}}, & i \neq s, \\ \dfrac{z_{ij}}{z_{ik}}, & i = s, \end{cases}$$

$$i = 0, \ldots, m, \quad j = 0, \ldots, n.$$

В этой записи $\bar{z}_{0i} = \bar{x}^{l_i}$, $i = 1, \ldots, m$, причем $l_s = k$, $l_i = \tilde{l}_i$ для $i \neq s$ и $\bar{z}_{00} = (c, \bar{x})$, где $\bar{x}$ — новая вершина.

Сходимость описанного алгоритма решения невырожденной задачи гарантируется тем, что при сменах допустимых базисов значение критерия возрастает. Поэтому повторяться они не могут, и так как их число не больше, чем C_n^m, за конечное число шагов либо будет найдено решение задачи, либо установлено, что ограниченного решения не существует. Алгоритм применим и в вырожденных случаях, но тогда формальная замена одного допустимого базиса другим не обязательно приведет к изменению точки. Соответственно, возможно так называемое *зацикливание* — бесконечное повторение набора допустимых базисов, отвечающих одной вершине. Реализуется оно или нет — зависит от того, по каким правилам выбираются номера k и $\tilde{l}_s$ вводимой и выводимой из базиса координат (формула (4.9) в вырожденном случае, вообще говоря, не определяет s единственным образом).

Проиллюстрируем работу метода с использованием симплекс-таблиц на задаче

$$\max(-2x^1 - x^2 - x^3),$$
$$\begin{cases} x^1 - x^4 - 2x^6 = 5, \\ x^2 + 2x^4 - 3x^5 + x^6 = 3, \\ x^3 + 2x^4 - 5x^5 + 6x^6 = 5, \\ x^i \geqslant 0, \quad i = 1, \quad , 6 \end{cases}$$

В качестве исходного базисного допустимого решения здесь можно взять $\tilde{x}^1 = 5$, $\tilde{x}^2 = 3$, $\tilde{x}^3 = 5$, $\tilde{x}^i = 0$, $i = 4, 5, 6$. Элементы z_{ij}, $i = 1, 2, 3$, $j = 1, 2, \ldots, 6$, связанной с ним симплекс-таблицы совпадают с соответствующими элементами матрицы условий. В целом таблица, отвечающая выбранному допустимому базису, выглядит так:

−13	0	0	0	−3	8	−5
5	1	0	0	−1	0	−2
3	0	1	0	2	−3	1
5	0	0	1	2	−5	6

Поскольку среди оценок замещения есть отрицательные, его можно улучшить. Для этого введем в базис переменную x^4 (т. е. возьмем в качестве ведущего столбец с оценкой замещения, равной —3). Тогда *ведущей строкой* будет вторая, т. е. из базиса будет выведена переменная x^2. Чтобы вычислить элементы новой симплекс-таблицы, ведущую строку нужно разделить на ведущий элемент $z_{24}=2$ и, умножив результат на 3, 1, —2, сложить получающиеся строки с нулевой, первой и третьей строкой, соответственно. Это приведет к таблице:

—8,5	0	1,5	0	0	3,5	—3,5
6,5	1	0,5	0	0	—1,5	—1,5
1,5	0	0,5	0	1	—1,5	0,5
2	0	—1	1	0	—2	5

Таким образом, базисные компоненты новой допустимой вершины $\tilde{x}^{l_1}=6{,}5$, $\tilde{x}^{l_2}=1{,}5$, $\tilde{x}^{l_3}=2$, где $l_1=1$, $l_2=4$, $l_3=3$. Она опять-таки не оптимальна — шестая оценка замещения отрицательна. Введем шестую переменную в базис. Для нее ведущей строкой будет третья, т. е. из базиса надо вывести переменную с номером l_3 (это — переменная x^3). Симплекс-таблица, связанная с новым допустимым базисом, выглядит так:

—7,1	0	0,8	0,7	0	2,1	0
7,1	1	0,2	0,3	0	—2,1	0
1,3	0	0,6	—0,1	1	—1,3	0
0,4	0	—0,2	0,2	0	—0,4	1

Все оценки замещения положительны и, следовательно, базисное допустимое решение с ненулевыми компонентами $\tilde{x}^{l_1}=7{,}1$, $\tilde{x}^{l_2}=1{,}3$, $\tilde{x}^{l_3}=0{,}4$, где $l_1=1$, $l_2=4$, $l_3=6$, оптимально.

2. Выбор начального допустимого базисного решения. В примере, рассмотренном в конце предыдущего пункта, матрица условий имела специальную структуру, что позволило сразу указать допустимое базисное решение, с которого мы и начали искать оптимум с помощью симплекс-метода. В общем же случае выделить для задачи вида

$$\begin{gathered}\max\,(c^1x^1+\ldots+c^nx^n),\\ a_{11}x^1+\ldots+a_{1n}x^n=b^1,\\ \ldots\ldots\ldots\ldots\\ a_{m1}x^1+\ldots+a_{mn}x^n=b^m,\\ x^1\geqslant 0,\quad\ldots,\quad x^n\geqslant 0\end{gathered}\tag{4.10}$$

набор из m координат точки x, которые были бы базисными для некоторой вершины допустимого многогранника, немногим проще, чем отыскать ее решение. На практике проблему выбора начального допустимого базиса обходят ценой увеличения размерности задачи. Как именно — показано ниже.

Преобразуем условия — равенства в (4.10) так, чтобы вектор правых частей стал неотрицательным. Для этого, возможно, придется умножить некоторые из них на -1. Далее, обозначив через $\bar{a}_{ij}$, $\bar{b}^j$ параметры преобразованных условий, введем вспомогательные переменные x^{n+i}, $i=1,\ldots,m$, и рассмотрим многогранник, заданный системой ограничений:

$$\begin{gathered}\bar{a}_{11}x^1+\ldots+\bar{a}_{1n}x^n+x^{n+1}=\bar{b}^1,\\ \ldots\ldots\ldots\ldots\\ \bar{a}_{1m}x^1+\ldots+\bar{a}_{mn}x^n+x^{n+m}=\bar{b}^m,\\ x^i\geqslant 0,\quad i=1,\ \ldots,\ m+n.\end{gathered}\tag{4.11}$$

Легко понять, что его вершиной будет, в частности, точка $\tilde{x}$ с координатами

$$\begin{aligned}\tilde{x}^i&=0,\quad i=1,\ \ldots,\ m,\\ \tilde{x}^{n+j}&=\bar{b}^j,\quad j=1,\ \ldots,\ m.\end{aligned}\tag{4.12}$$

Начиная с нее, мы можем отыскать симплекс-методом минимум суммы вспомогательных переменных при огра-

ничениях (4.11). Если допустимое множество исходной задачи (4.10) непусто, в полученном решении $\bar{x}$ эта сумма, а стало быть, и каждая из вспомогательных переменных будет равна нулю. Последнее означает, что n-мерная точка с координатами $\bar{x}^1, \ldots, \bar{x}^n$ является вершиной допустимого многогранника задачи (4.10).

Таким образом, можно предложить следующий способ решения задачи (4.10): начиная с точки (4.12) симплекс-методом решается вспомогательная задача на поиск минимума суммы $x^{n+1}+\ldots+x^{n+m}$ при ограничениях (4.11); найденная точка определяет вершину допустимого многогранника задачи (4.10), исходя из которой эта задача и решается симплекс-методом. Это — так называемый двухфазный симплекс-метод.

Более распространена схема применения симплекс-метода, в которой фаза поиска допустимого базиса не отделяется от фазы движения к оптимуму. Эта схема состоит в том, что вместо (4.10), начиная с точки (4.12), решается задача (ее принято называть M-задачей) максимизации линейной формы

$$c^1x^1+\ldots+c^nx^n-M\,(x^{n+1}+\ldots+x^{n+m})$$

при ограничениях (4.11), где M — положительное число. Нетрудно показать, что при любом, достаточно большом M первые n координат полученного решения определяют оптимальную вершину задачи (4.10). Строго доказывать этот факт мы не будем, но поясним его на простом примере. Рассмотрим такую задачу:

$$\begin{aligned} &\max c^1x^1,\\ &x^1+x^2=\bar{x},\\ &x^1-x^3=\underline{x},\\ &x^i\geqslant 0,\quad i=1,\ 2,\ 3, \end{aligned} \tag{4.13}$$

где $\bar{x}>\underline{x}>0$ и $c^1<0$. Множество допустимых для нее значений переменной x^1 представляет собой отрезок $[\underline{x},\ \bar{x}]$. Максимальное значение критерия реализуется при $x^1=\underline{x}$, $x^2=\bar{x}-\underline{x}$, $x^3=0$. Связанная с (4.13) M-задача выглядит

так:

$$\begin{gathered}\max\,(c^1x^1 - M\,(x^4 + x^5)),\\ x^1 + x^2 + x^4 = \bar{x},\\ x^1 - x^3 + x^5 = \underline{x},\\ x^i \geqslant 0, \quad i = 1, \ldots, 5.\end{gathered} \tag{4.14}$$

Значение первой координаты ее оптимальной точки есть решение задачи

$$\begin{gathered}\max F\,(x^1),\\ x^1 \geqslant 0,\end{gathered}$$

где $F\,(x^1)$ — максимальное значение критерия в задаче

$$\begin{gathered}\max_{x^2,\,x^3,\,x^4,\,x^5}\,(c^1x^1 - Mx^4 - Mx^5),\\ x^2 + x^4 = \bar{x} - x^1,\\ -x^3 + x^5 = \underline{x} - x^1,\\ x^i \geqslant 0, \quad i = 2, \ldots, 5.\end{gathered}$$

(Если допустимое множество последней пусто, а это будет при $x^1 > \bar{x}$, полагаем $F\,(x^1) = -\infty$.) График функции $F\,(x^1)$

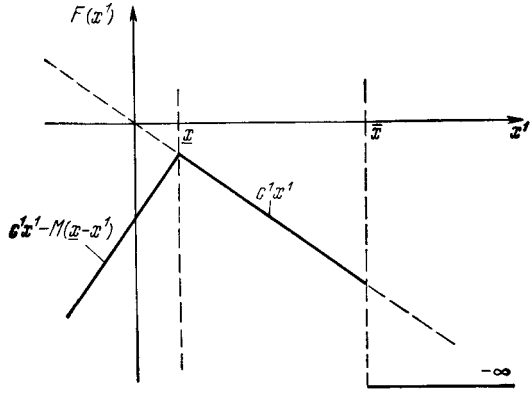


Рис. 4.1.

при $M > |c^1|$ показан на рис. 4.1. Точкой ее максимума является $x^1 = \underline{x}$. Соответственно, решением задачи (4.14) будет $x^1 = \underline{x}$, $\bar{x}^2 = \bar{x} - \underline{x}$, $x^3 = x^4 = x^5 = 0$. Его первые три координаты составляют решение задачи (4.13).

3. Вырожденность. Описание функционирования простейшей версии симплекс-метода, приведенное в начале параграфа, дано применительно к невырожденной задаче линейного программирования. Невырожденность обеспечивает смену вершин на каждой итерации, на основе чего доказывается сходимость алгоритма. Последний, будучи дополнен правилом выбора ведущей строки среди строк с номерами s, удовлетворяющих (4.9), реализуем и в вырожденных случаях. Однако здесь нет гарантии того, что на каждой итерации вершина допустимого многогранника сменится, — базис, и симплекс-таблица, разумеется, будут меняться, но если список положительных базисных координат при этом сохраняется, точка останется неизменной. В данном случае возникает опасность вырождения процесса поиска решения задачи в бесконечный циклический перебор нескольких допустимых базисов, отвечающих одной, вообще говоря, не оптимальной вершине. Например, если решение задачи

$$\max\left(-\frac{3}{4}x^1+150x^2-\frac{1}{50}x^3+6x^4\right),$$

$$\frac{1}{4}x^1-60x^2-\frac{1}{25}x^3+9x^4+x^5=0,$$

$$\frac{1}{2}x^1-90x^2-\frac{1}{50}x^3+3x^4+x^6=0,$$

$$x^3+x^7=0,$$

$$x^i\geqslant 0,\quad i=1,\ \ldots,\ 7,$$

начать с допустимого базиса, содержащего пятую, шестую и седьмую координаты, причем ведущий столбец k выбирать по принципу минимума оценки замещения, а среди строк с номерами s, для которых

$$\frac{\tilde{x}^{l_s}}{z_{sk}}=\min_{\{i\ \ z_{ik}>0\}}\frac{\tilde{x}^{l_i}}{z_{ik}}=0,$$

ведущей всегда считать строку с наименьшим номером, симплекс-алгоритм зациклится в исходной точке (которая, как легко видеть, не оптимальна; см. табл. 4.1.).

Нужно сказать, что зацикливание на практике встречается крайне редко, хотя среди решаемых задач вырожденные составляют большинство (в частности, если вектор правых частей ограничений — равенств задачи имеет

нулевые компоненты, процедуру поиска ее решения по схеме, описанной в предыдущем пункте, предлагается начинать с вырожденного допустимого базиса). Однако раз оно возможно, нужно иметь средства борьбы с ним, а точнее — правила вывода из базиса, исключающие вероятность его повторения. С одним из таких правил мы сейчас познакомимся.

Таблица 4.1

№ итерации	Коэффициенты нулевой строки (оценка замещения)							Базисные переменные	Значение критериальной функции
1	—3/4	150	—1/50	6	0	0	0	$x^5x^6x^7$	0
2	0	—30	—7/50	33	3	0	0	$x^1x^6x^7$	0
3	0	0	—2/25	18	1	1	0	$x^1x^2x^7$	0
4	1/4	0	0	—3	—2	3	0	$x^3x^2x^7$	0
5	—1/2	120	0	0	—1	1	0	$x^3x^4x^7$	0
6	—7/4	330	1/50	0	0	—2	0	$x^5x^4x^7$	0
7	—3/4	150	—1/50	6	0	0	0	$x^5x^6x^7$	0

Пусть дана вырожденная задача, матричная запись которой выглядит так:

$$\begin{gathered} \max(c,\ x), \\ Ax = b, \\ x \geqslant 0. \end{gathered} \tag{4.15}$$

Рассмотрим наряду с ней задачу вида

$$\begin{gathered} \max(c,\ x), \\ Ax = b + \sum_{i=1}^{n} \varepsilon^i a_i, \\ x \geqslant 0, \end{gathered} \tag{4.16}$$

где ε — положительное число, а ε^i — его i-я степень. При достаточно малых ε задача (4.16) не вырождена и каждый из ее допустимых базисов будет допустимым базисом исходной задачи, причем оптимальный допустимый базис для задачи (4.16) будет оптимальным и для задачи (4.15). Следовательно, найти решение последней можно, переби-

рая ее допустимые базисы в том порядке, в котором симплекс-алгоритм перебирал бы их, решая задачу (4.16). Зацикливание при этом исключено, поскольку задача (4.16) не вырождена.

Посмотрим теперь, как сменяются допустимые базисы при решении симплекс-алгоритмом задачи (4.16). Обозначим через $\tilde{x}$, $\tilde{x}(\varepsilon)$ допустимые базисные решения задач (4.15), (4.16), соответствующие очередному из них. Тогда, как легко понять,

$$\tilde{x}^{\tilde{l}_i}(\varepsilon) = \tilde{x}^{\tilde{l}_i} + \varepsilon z_{i1} + \ldots + \varepsilon^n z_{in}, \quad i = 1, \ldots, m,$$

где $\tilde{l}_i$ — номера базисных координат, а z_{ij} — элементы симплекс-таблицы для задачи (4.15). Сразу отметим, что эта таблица отличается от симплекс-таблицы для задачи (4.16) только крайним левым столбцом. Поэтому в качестве номера координаты, которую следовало бы ввести в базис на текущей итерации симплекс-алгоритма решения задачи (4.16), мы можем взять любой номер k, для которого оценка замещения σ^k в симплекс-таблице задачи (4.15) отрицательна. При этом номер $\tilde{l}_s$ выводимой из базиса координаты определится из равенства

$$\frac{\tilde{x}^{\tilde{l}_s}(\varepsilon)}{z_{sk}} = \min_{\{i \;\; z_{ik} > 0\}} \frac{\tilde{x}^{\tilde{l}_i}(\varepsilon)}{z_{ik}} =$$

$$= \min_{\{i \;\; z_{ik} > 0\}} \frac{1}{z_{ik}} \left(\tilde{x}^{\tilde{l}_i} + \varepsilon z_{i1} + \ldots + \varepsilon^n z_{in}\right).$$

Когда величина $\varepsilon > 0$ близка к нулю, процедура определения ведущей строки s состоит в следующем: сравниваются значения $\tilde{x}^{\tilde{l}_i}/z_{ik}$ для i таких, что $z_{ik} > 0$; если одно из них меньше, чем все остальные, соответствующая строка и будет ведущей; в противном случае для тех i, при которых $z_{ik} > 0$ и $\tilde{x}^{\tilde{l}_i}/z_{ik}$ минимально, сравниваются отношения z_{i1}/z_{ik}; опять-таки, если минимальным будет только одно из них, отвечающая ему строка — ведущая; если же минимальных z_{i1}/z_{ik} несколько, для них сравниваются отношения z_{i2}/z_{ik} и так далее.

Мы пришли к так называемому *лексикографическому* правилу выбора ведущей строки, исключающему возмож-

ность зацикливания симплекс-алгоритма при решении вырожденной задачи: в качестве ведущей строки надо брать ту, для которой вектор

$$\left\{\frac{\tilde{x}_i}{z_{ik}}, \frac{z_{i1}}{z_{ik}}, \ldots, \frac{z_{in}}{z_{ik}}\right\}^T$$

лексикографически минимален. Напомним, что вектор a *лексикографически* меньше вектора b, если при сравнении их координат слева направо первая несовпадающая координата вектора a меньше, чем соответствующая координата вектора b.

4. Алгоритм с обратной матрицей. В изложенной выше простейшей версии симплекс-метода мы оперировали симплекс-таблицами, основу которых составляют матрицы систем уравнений, получающихся из исходных ограничений — равенств задачи

$$\begin{gathered}\max (c, x),\\ Ax = b,\\ x \geqslant 0,\end{gathered} \tag{4.17}$$

если разрешать их относительно базисных переменных, т. е. умножать A слева на матрицу A'^{-1}, где A' составлена из базисных столбцов. Допустим, что на очередной итерации это — первые m столбцов, и обозначим через x', x'' векторы, образованные из первых m и последних $n-m$ координат вектора x, через c', c'' — соответствующие им части вектора c, а через A'' — матрицу, составленную из последних, небазисных, $n-m$ столбцов матрицы A. Тогда ограничения-равенства задачи (4.17) можно переписать так:

$$A'x' + A''x'' = b.$$

Эквивалентная им система уравнений, разрешенных относительно базисных переменных (т. е. вектора x'), имеет вид

$$x' + A'^{-1}A''x'' = A'^{-1}b = \tilde{x}', \tag{4.18}$$

где $\tilde{x}'$ — вектор, образованный координатами рассматриваемой вершины. При этом значение критерия в любой

точке $x=\{x', x''\}^T$, удовлетворяющей (4.18), равно

$$(c', x')+(c'', x'')=(c', \tilde{x}')-(c', A'^{-1}A''x'')+(c'', x'')=$$
$$=(c', \tilde{x}')-((A'^{-1}A'')^T c'-c'', x'').$$

Сопоставляя это равенство и (4.18) с (4.3), (4.2), видим, что симплекс-таблицу в новых обозначениях можно записать так:

$$\left[\begin{array}{c|cc} (c', \tilde{x}') & 0 & c'^T A'^{-1}A''-c''^T \\ \tilde{x}' & E & A'^{-1}A'' \end{array}\right], \qquad (4.19)$$

где F — единичная матрица.

Из полученного представления симплекс-таблицы ясно, что для перехода к новому допустимому базису по прежней схеме достаточно знать матрицу A'^{-1} и вектор-строку $\lambda=c'^T A'^{-1}$: последовательно перемножая λ на небазисные столбцы и вычитая каждый раз из результата соответствующий элемент вектора c'', вычислим оценки замещения и выделим ведущий столбец; умножив на него матрицу A'^{-1}, найдем те коэффициенты замещения, которые используются в формуле для определения ведущей строки, и выделим эту строку; затем по старым формулам вычислим значения координат нового базиса. Таким образом, можно предложить версию симплекс-метода, при реализации которой на ЭВМ в памяти машины вместо симплекс-таблиц придется хранить только $(m\times m)$-матрицу A'^{-1}, m-мерный вектор λ, текущие значения базисных координат и критерия. Как правило, эта версия требует значительно меньшего объема памяти, чем первоначальная. Чтобы она была, кроме того, эффективна в смысле быстродействия, надо заложить в нее экономный алгоритм пересчета (при сменах допустимых базисов) матрицы, обратной к базисной. Вывести такой алгоритм совсем несложно.

Вернемся к системе уравнений (4.18). Ее матрица,

$$[E \mid A'^{-1}A''] = A'^{-1}A$$

составляет основной блок симплекс-таблицы (4.19). Представленные в п. 1 формулы преобразования последней при переходе к новому базису есть не что иное, как

формулы умножения ее на матрицу

$$M = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & -\frac{z_{0k}}{z_{sk}} & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & -\frac{z_{1k}}{z_{sk}} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & 1 & \dots & -\frac{z_{s-1\,k}}{z_{sk}} & 0 & \dots & 0 \\ \dots & \dots & \dots & 0 & \dots & \frac{1}{z_{sk}} & 0 & \dots & 0 \\ \dots & \dots & \dots & 0 & \dots & -\frac{z_{s+1\,k}}{z_{sk}} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & 0 & -\frac{z_{mk}}{z_{sk}} & 0 & \dots & 1 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} = \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix}$$

+ 1-я строка

$$= \left[\begin{array}{c|ccc} 1 & 0 & \dots \\ \hline 0 & & \\ 0 & & M' \end{array}\right].$$

При этом $A'^{-1}A$ умножается слева на M', и в результате получается матрица эквивалентной (4.18) системы уравнений, разрешенных относительно новых базисных переменных, т. е.

$$M'A'^{-1}A = \bar{A}'^{-1}A,$$

где $\bar{A}'$ — новая базисная матрица. Отсюда следует, что

$$M'A'^{-1} = \bar{A}'^{-1},$$

и, стало быть, матрица, обратная к базисной, пересчитывается по тем же формулам, что и симплекс-таблица, т. е.

$$\bar{\beta}_{ij} = \begin{cases} \dfrac{\beta_{ij} - \beta_{sj}z_{ik}}{z_{sk}}, & i \neq s, \\ \dfrac{\beta_{sj}}{z_{sk}}, & i = s, \end{cases} \qquad i = 1, \dots, m, \ j = 1, \dots, m,$$

где через β_{ij}, $\bar{\beta}_{ij}$, $i = 1, \dots, m$, $j = 1, \dots, m$, обозначены (i, j)-е элементы матриц A'^{-1}, $\bar{A}'^{-1}$. Полагая $j = 0$, получим формулы пересчета вектора λ, которые выглядят так:

$$\bar{\lambda}^i = \bar{\beta}_{i0} = \begin{cases} \dfrac{\lambda^i - \lambda^s z_{ik}}{z_{sk}}, & i \neq s, \\ \dfrac{\lambda^s}{z_{sk}}, & i = s, \end{cases} \qquad i = 1, \dots, m.$$

Описанная версия симплекс-метода используется на практике значительно чаще, чем алгоритм с симплекс-таблицами. Наиболее эффективные из существующих реализаций симплекс-метода представляют собой различные ее воплощения, отличающиеся друг от друга методами хранения обратной матрицы в компактной форме.

§ 5. Двойственные задачи и методы

1. Теоремы двойственности в линейном программировании. Как уже было сказано в предыдущем параграфе, оценки замещения $\hat{\sigma}^k$, соответствующие некоторому базисному допустимому решению $\hat{x}$ задачи

$$\begin{aligned} &\max (c,\ x), \\ &Ax = b, \\ &x \geqslant 0, \end{aligned} \tag{5.1}$$

можно вычислить, в матричной записи, по формуле

$$\hat{\sigma} = A''^T \hat{\lambda} - c'',$$
$$\hat{\lambda} = (A'^{-1})^T c',$$

т. е.

$$\hat{\sigma}^k = \sum_{r=1}^{m} \hat{\lambda}^r a_{rk} - c^k,$$

где величины $\hat{\lambda}^r$, $r = 1, \ldots, m$, определяются из решения системы уравнений

$$\sum_{r=1}^{m} \hat{\lambda}^r a_{rl_i} - c^{l_i} = 0, \quad i = 1, \ldots, m. \tag{5.2}$$

Здесь l_i — номера базисных переменных, $\hat{\sigma}$ — вектор, составленный из оценок замещения для небазисных переменных, c', c'' — векторы, координатами которых являются компоненты вектора c, отвечающие базисным и небазисным переменным соответственно, A' — матрица, образованная из базисных, а A'' — из небазисных столбцов матрицы A. При этом вектор $\hat{x}'$, состоящий из базисных координат

вектора $\hat{x}$, удовлетворяет равенствам

$$A'\hat{x}' = A\hat{x} = b,$$
$$(c',\ \hat{x}') = (c,\ \hat{x})$$

и, следовательно,

$$(b,\ \hat{\lambda}) = (\hat{\lambda},\ A'\hat{x}') = (A'^{T}\hat{\lambda},\ \hat{x}') = (c',\ \hat{x}') = (c,\ \hat{x}) \qquad (5.3)$$

Пусть теперь $\hat{x}$ — не просто допустимое, а оптимальное базисное решение задачи (5.1), найденное симплекс-методом. Тогда соответствующие ему оценки замещения неотрицательны, т. е. для вектора $\hat{\lambda}$, связанного с $\hat{x}$, помимо равенств (5.2) справедливы неравенства:

$$\sum_{r=1}^{m} \hat{\lambda}^r a_{rj} \geqslant c^j,\quad 1 \leqslant j \leqslant n;\quad j \neq l_i,\quad i = 1,\ \ldots,\ m.$$

Таким образом, вектор $\hat{\lambda}$ является допустимым для системы ограничений

$$\sum_{r=1}^{m} \lambda^r a_{rj} \geqslant c^j,\quad j = 1,\ \ldots,\ n, \qquad (5.4)$$

и мы покажем сейчас, что $\hat{\lambda}$ — решение задачи минимизации линейной формы $(b,\ \lambda)$ на множестве точек, удовлетворяющих данным ограничениям. Эту задачу, матричная запись которой имеет вид

$$\begin{aligned} &\min\,(b,\ \lambda), \\ &A^{T}\lambda \geqslant c, \end{aligned} \qquad (5.5)$$

называют двойственной по отношению к задаче (5.1).

Чтобы установить оптимальность рассматриваемого вектора $\hat{\lambda}$ в задаче (5.5), сравним значения критериев прямой (5.1) и двойственной (5.5) задач на их произвольных допустимых решениях x, λ. Для этого умножим векторное неравенство

$$A^{T}\lambda \geqslant c,$$

которому, по определению, подчиняется вектор λ, скалярно на вектор x Все компоненты последнего неотрицательны, и поэтому после умножения знак неравенства сохранится, т е.

$$(A^{T}\lambda,\ x) = (\lambda,\ Ax) \geqslant (c,\ x),$$

а так как

$$Ax = b,$$

отсюда следует, что

$$(b,\ \lambda) \geqslant (c,\ x). \tag{5.6}$$

Это — фундаментальное неравенство теории двойственности в линейном программировании. В частности, полагая в нем $x = \hat{x}$ и учитывая (5.3), для любого допустимого для задачи (5.5) вектора λ получим

$$(b,\ \lambda) \geqslant (c,\ \hat{x}) = (\hat{\lambda},\ b),$$

что и требовалось доказать.

Итак, коль скоро прямая задача (5.1) разрешима, можно утверждать, что разрешима и двойственная по отношению к ней задача (5.5). Справедливо и обратное утверждение: из разрешимости задачи (5.5) следует разрешимость задачи (5.1). Для доказательства этого факта снова обратимся к симплекс-методу. Чтобы можно было говорить о его применении в том виде, как он был описан ранее, для поиска решения задачи (5.5), преобразуем ее в следующую, эквивалентную ей, задачу:

$$\max\left(-\sum_{i=1}^{m} b^i\lambda^i\right),$$

$$\sum_{i=1}^{m} a_{ik}\lambda^i - \lambda^{m+k} = c^k, \quad k = 1, \ldots, n, \tag{5.7}$$

$$\lambda^{m+k} \geqslant 0, \quad k = 1, \ldots, n.$$

Здесь λ^{m+k} — вспомогательные переменные. Эту задачу от канонической, на которую были рассчитаны представленные выше версии симплекс-метода, отличает одно — отсутствие условий неотрицательности для первых m переменных. Однако это не должно смущать читателя. Легко понять, что́ достаточно изменить в алгоритмах, о которых идет речь, чтобы приспособить их к задаче (5.7), — нужно при выборе ведущей строки искать минимум отношения базисной координаты к коэффициенту замещения только по координатам с номером большим, чем m, а прочие координаты всегда считать базисными. Вся остальная техника поиска решения сохраняется.

Если задача (5.5) разрешима, таковой будет и задача (5.7). Пусть $\hat{\lambda}^i$, $i = 1, \ldots, m+n$, есть координаты ее базис-

ного оптимального решения, найденного симплекс-методом. Тогда оценки замещения для небазисных переменных λ^{μ_i}, $i=1, \ldots, m$, $m<\mu_i\leqslant n+m$, неотрицательны. Они, по определению, равны умноженным на минус единицу коэффициентам при λ^{μ_i} в выражениях для критерия задачи (5.7) через небазисные координаты произвольного вектора λ, удовлетворяющего ее ограничениям-равенствам. Чтобы получить это выражение, добавим к целевой функции линейную комбинацию разностей

$$\sum_{i=1}^{n} a_{ik}\lambda^i-\lambda^{m+k}-c^k$$

с некоторыми весами $\hat{x}^k$. Это не изменит ее значения при рассматриваемых λ, т. е. будет

$$-\sum_{i=1}^{m} b^i\lambda^i=-\sum_{i=1}^{m} b^i\lambda^i+\sum_{k=1}^{m}\hat{x}^k\left(\sum_{i=1}^{m} a_{ik}\lambda^i-\lambda^{m+k}-c^k\right)=$$

$$=\sum_{i=1}^{m}\lambda^i\left(\sum_{k=1}^{n} a_{ik}\hat{x}^k-b^i\right)-$$

$$-\sum_{\mu_i}\lambda^{\mu_i}\hat{x}^{\mu_i-m}-\sum_{\substack{m<j\leqslant n\\ j\neq\mu_i}}\lambda^j\hat{x}^{j-m}-\sum_{k=1}^{n} c^k\hat{x}^k.$$

Подберем теперь $\hat{x}^k$ так, чтобы коэффициенты перед базисными переменными в правой части равенства обратились в нуль:

$$\begin{gathered}\sum_{k=1}^{n} a_{ik}\hat{x}^k=b^i,\\ \hat{x}^{j-m}=0,\quad m<j\leqslant n,\quad j\neq\mu_i.\end{gathered}\tag{5.8}$$

(Эта система уравнений разрешима относительно $\hat{x}^k$, так как ее матрица есть транспонированная к базисной матрице решения $\hat{\lambda}^i$, $i=1, \ldots, m+n$, и, соответственно, не вырождена.) Тогда получим

$$-\sum_{i=1}^{m} b^i\lambda^i=-\sum_{\mu_i}\lambda^{\mu_i}\hat{x}^{\mu_i-m}-\sum_{k=1}^{n} c^k\hat{x}^k.\tag{5.9}$$

Следовательно, оценки замещения есть просто-напросто $\hat{x}^{\mu_i-m}$. Так как они неотрицательны, вектор $\hat{x}$, с учетом

(5.8), будет допустимым для задачи (5.1), а поскольку $\hat{\lambda}^{\mu_i} = 0$, из (5.9) имеем

$$\sum_{i=1}^{m} b^i \hat{\lambda}^i = \sum_{k=1}^{n} c^k \hat{x}^k.$$

Вектор, составленный из первых m компонент решения задачи (5.7), является решением задачи (5.5). Поэтому из последнего равенства и неравенства (5.6) следует, что $\hat{x}$ — решение задачи (5.1).

Итак, мы установили, что из разрешимости одной из задач (5.1), (5.5) следует разрешимость другой, причем оптимальные значения критериев совпадают. Этот результат называют *первой теоремой двойственности*. В ее доказательстве центральную роль играло неравенство (5.6). Из него вытекает и *вторая теорема двойственности*: для того чтобы задачи (5.1), (5.5) были разрешимы, достаточно, чтобы их допустимые множества содержали хотя бы по одной точке.

Действительно, в данном случае в силу (5.6) значение критерия прямой задачи ограничено сверху на ее допустимом многограннике величиной (b, λ), где λ — некоторое допустимое решение двойственной задачи (5.5). Но линейная форма, ограниченная сверху на многограннике, достигает на нем максимума и, следовательно, прямая задачи разрешима. Аналогично устанавливается и разрешимость двойственной задачи.

Как утверждает первая теорема двойственности, если $\hat{x}$, $\hat{\lambda}$ — решения задачи (5.1), (5.5), выполнено равенство

$$(b, \hat{\lambda}) = (c, \hat{x}),$$

т. е. совпадение значений критериев пары двойственных задач в их допустимых точках есть необходимое условие оптимальности этих точек. Это же условие в силу (5.6) является и достаточным. Таким образом, мы получили простой *критерий оптимальности* двух допустимых решений задачи (5.1), (5.6): для того чтобы допустимые решения $\hat{x}$, $\hat{\lambda}$ прямой и двойственной задач были оптимальными, необходимо и достаточно, чтобы значения прямого и двойственного критериев на них совпадали.

Поскольку для любых допустимых x, λ имеем

$$(b, \lambda) - (c, x) = (Ax, \lambda) - (c, x) = \\ = (A^T\lambda, x) - (c, x) = (A^T\lambda - c, x),$$

полученный критерий оптимальности можно сформулировать и так: для того чтобы допустимые $\hat{x}$, $\hat{\lambda}$ были решениями прямой и двойственной задач, необходимо и достаточно, чтобы выполнялось равенство

$$(A^T\hat{\lambda} - c, \hat{x}) = 0, \tag{5.10}$$

или, что то же самое, равенства

$$\left(\sum_{r=1}^{m} \hat{\lambda}^r a_{rj} - c^j\right)\hat{x}^j = 0, \quad j = 1, \ldots, n. \tag{5.11}$$

Последние принято называть *условиями дополняющей нежесткости*, а эквивалентность (5.10), (5.11) следует из неотрицательности сомножителей скалярного произведения в (5.10).

Наконец, отметим, что если строки матрицы условий A в задаче (5.1) линейно независимы, допустимые множества этой и двойственной к ней задач одновременно пустыми быть не могут. (Установить этот факт несложно, но для этого нужна теорема об отделимости, которая будет доказана лишь в следующей главе. Поэтому мы приводим его без доказательства.) Таким образом, в силу теорем двойственности при анализе пары задач (5.1), (5.5) с матрицей A полного ранга могут реализоваться только два случая:

а) обе задачи имеют допустимые решения, и тогда обе они разрешимы;

б) у одной из задач нет допустимых решений, и тогда значения критерия второй задачи на ее допустимом множестве не ограничены сверху, если это — задача максимизации, и снизу — в противоположном случае.

2. Геометрическая интерпретация теорем двойственности. Вспомним геометрическую интерпретацию задачи (5.1), изложенную в третьем параграфе данной главы. Мы рассматривали преобразование n-мерного пространства векторов x в $m+1$-мерное пространство векторов $u = \{u^0,$

$u^1, \ldots, u^m\}^T$ по формулам

$$u^0 = \sum_{j=1}^{n} c^j x^j,$$
$$u^i = \sum_{j=1}^{n} a_{ij} x^j, \quad i = 1, \ldots, m. \tag{5.12}$$

Это преобразование переводит положительный ортант пространства E_n (т. е. множество векторов x таких, что $x^j \geqslant 0$, $j = 1, \ldots, n$) в многогранный конус K в пространстве E_{m+1}. Направляющими ребер этого конуса будут векторы $\{c^j, a_{1j}, \ldots, a_{mj}\}^T$, $j = 1, \ldots, n$. Образом в E_{m+1} множества допустимых решений задачи (5.1) является заключенный в конусе K отрезок прямой Q, проходящей через точку $\{0, b^1, \ldots, b^m\}^T$ параллельно оси Ou^0. Координаты точек этой прямой определяются так:

$$-\infty < u^0 < +\infty, \quad u^1 = b^1, \ldots, u^m = b^m.$$

Рассмотрим теперь в пространстве E_{m+1} семейство проходящих через начало координат гиперплоскостей с уравнениями

$$\sum_{i=1}^{m} \lambda^i u^i - u^0 = 0, \tag{5.13}$$

где λ^i — некоторые числа. Нетрудно убедиться, что если λ^i — координаты какого-нибудь допустимого решения задачи (5.5), двойственной к (5.1), конус K будет лежать с одной стороны от гиперплоскости (5.13). Действительно, возьмем произвольный вектор u из K. Его координаты, по определению конуса K, есть линейные комбинации координат некоторого неотрицательного вектора x и связаны с последними преобразованиями (5.12). Таким образом, для $u \in K$ имеем

$$\sum_{i=1}^{m} \lambda^i u^i - u^0 = \sum_{i=1}^{m} \lambda^i \sum_{j=1}^{n} a_{ij} x^j - \sum_{j=1}^{n} c^j x^j =$$
$$= \sum_{j=1}^{n} (\lambda^i a_{ij} - c^j) x^j, \tag{5.14}$$
$$x^j \geqslant 0, \quad j = 1, \ldots, n.$$

Если же λ^i — координаты допустимого решения двойственной задачи, т. е.

$$\sum_{i=1}^{n} \lambda^i a_{ij} - c^j \geqslant 0, \quad i = 1, \ldots, n,$$

из (5.14) следует, что

$$\sum_{i=1}^{m} \lambda^i u^i - u^0 \geqslant 0. \tag{5.15}$$

Данное неравенство, полученное для произвольного вектора из конуса K, и означает, что этот конус лежит с одной стороны от гиперплоскости (5.13), причем с той же стороны, что и вектор $\{-1, 0, \ldots, 0\}^T$, т. е. «под гиперплоскостью».

Мы показали, что каждому допустимому решению λ двойственной к (5.1) задачи (5.5) отвечает гиперплоскость (5.13), расположенная «над конусом K» Есть и обратная связь — параметры λ^i любой гиперплоскости (5.13), лежащей «над конусом K», образуют допустимое решение задачи (5.5). Действительно, соблюдение при некоторых λ^i неравенства (5.15) для каждого u из конуса K или, что то же самое, соблюдение неравенства

$$\sum_{j=1}^{n} (\lambda^i a_{ij} - c^j) x^j \geqslant 0$$

для любых $x^j \geqslant 0$ возможно лишь в том случае, если

$$\lambda^i a_{ij} - c^j \geqslant 0, \quad i = 1, \ldots, m.$$

Итак, допустимое множество двойственной задачи отождествляется с множеством направляющих векторов семейства тех гиперплоскостей вида (5.15), которые лежат над конусом K. При этом координаты точек пересечения каждой из них с прямой Q определяются уравнениями

$$\sum_{i=1}^{m} \lambda^i u^i - u^0 = 0,$$
$$u^1 = b^1, \ \ldots, \ u^m = b^m,$$

т. е. первая координата u^0 равна $\sum_{i=1}^{m} \lambda^i b^i$ и, соответственно, двойственная задача интерпретируется как задача поиска

гиперплоскости из указанного семейства с «наинизшей» точкой пересечения прямой Q. Полученный в предыдущем пункте критерий оптимальности позволяет утверждать, что эта точка будет принадлежать конусу K.

Чтобы пояснить сказанное рисунками, рассмотрим задачу

$$\begin{aligned} &\max\,(c^1x^1 + c^2x^2 + c^3x^3 + c^4x^4),\\ &a_{11}x^1 + a_{12}x^2 + a_{13}x^3 + a_{14}x^4 = b^1,\\ &a_{21}x^1 + a_{22}x^2 + a_{23}x^3 + a_{24}x^4 = b^2,\\ &x^1 \geqslant 0, \quad x^2 \geqslant 0, \quad x^3 \geqslant 0, \quad x^4 \geqslant 0. \end{aligned} \tag{5.16}$$

Соответствующий ей конус K с направляющими векторами ребер $\tilde{a}_t = \{c^t, a_{1t}, a_{2t}\}^T$ изображен на рис. 5.1. Рассечем

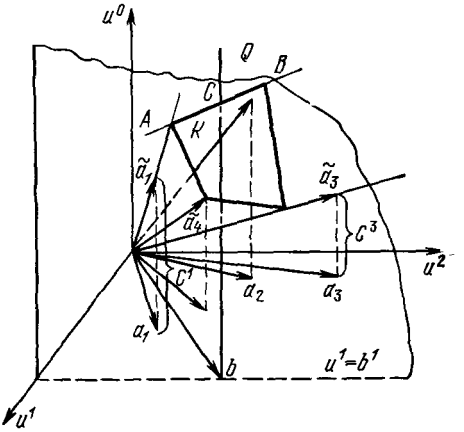


Рис. 5.1.

его плоскостью с уравнением $u^1 = b^1$ (см. рис. 5.1), содержащей прямую Q. В сечении получим четырехугольник, вершинами которого являются точки пересечения ребер конуса K (т. е. прямых, состоящих из точек $u = \tilde{a}_t t$, $t \geqslant 0$) с рассматриваемой плоскостью. Этот четырехугольник представлен на рис. 5.2 Допустимым решениям задачи (5.16) соответствуют принадлежащие ему точки прямой Q. Оптимальному решению — точка C.

На рис. 5.2 показаны также прямые AB, $A'B'$, по которым с плоскостью $u^1 = b^1$ пересекаются две плоскости

с уравнениями

$$u^0 = \lambda^1 u^1 + \lambda^2 u^2, \quad u^0 = \lambda'^1 u^1 + \lambda'^2 u^2,$$

расположенные над конусом K. Им отвечают допустимые решения $\lambda = \{\lambda^1, \lambda^2\}^T$, $\lambda' = \{\lambda'^1, \lambda'^2\}^T$ двойственной к (5.16) задачи. Значения критерия последней на этих решениях есть ординаты точек пересечения прямых AB, $A'B'$ с прямой Q. Из рис. 5.2 видно, что любая плоскость, лежащая над конусом K, пересечет эту прямую выше, чем та, которой соответствует прямая $A'B'$. Следовательно, λ' — оптимальное решение двойственной задачи. При этом оптимальное значение двойственного критерия есть ордината точки C, равная по построению оптимальному значению прямого критерия.

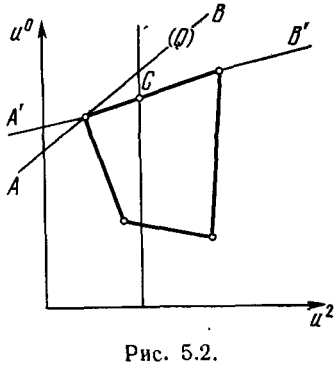


Рис. 5.2.

3. **Двойственный симплекс-метод.** В предыдущем параграфе были описаны две версии симплекс-метода, применяя которые к задаче

$$\max (c, x), \quad Ax = b, \quad x \geqslant 0, \tag{5.17}$$

мы за конечное число шагов получим ее решение. Последнее можно отыскать и по-другому, применив, к примеру, алгоритм с использованием симплекс-таблиц к задаче

$$\max (-b^1\lambda^1 - b^2\lambda^2 - \ldots - b^m\lambda^m),$$
$$\sum_{r=1}^{m} a_{ri}\lambda^r - \lambda^{m+i} = c^i, \quad i = 1, \ldots, n, \tag{5.18}$$
$$\lambda^{m+i} \geqslant 0, \quad i = 1, \ldots, n.$$

Тогда базисные координаты $\hat{x}^{s_i}$, $i = 1, \ldots, m$, оптимальной вершины задачи (5.17) будут получены на последней итерации как оценки замещения для небазисных переменных λ^{s_i+m}, $i = 1, \ldots, m$, $1 \leqslant s_i \leqslant n$, задачи (5.18) (спра-

ведливость данного утверждения была установлена в п. 1 настоящего параграфа). При этом нетрудно построить алгоритм решения задачи (5.18), оперирующий симплекс-таблицами задачи (5.17). К описанию этого алгоритма мы сейчас и перейдем.

Пусть дан некоторый допустимый базис задачи (5.18), т. е. известен список номеров n координат $(n+m)$-мерного вектора λ такой, что

а) в него входят первые m координат;

б) столбцы матрицы условий задачи (5.18), отвечающие попавшим в него координатам, линейно независимы;

в) значения координат из этого списка с номерами больше m, получающиеся из решения уравнений задачи (5.18) при нулевых значениях не включенных в него координат, неотрицательны.

Обозначим номера небазисных координат через μ_i, $i = 1, \ldots, m$, $m < \mu_i \leqslant n+m$ Тогда, как было показано в п. 1, соответствующие им оценки замещения $\hat{x}^{\mu_i - m}$ будут решением уравнений

$$\sum_{\substack{j=\mu_i-m \\ i=1,\ldots,m}} a_{rj}\hat{x}^j = b^r, \quad r = 1, \ldots, m.$$

Если все они неотрицательны, рассматриваемый базис оптимален для задачи (5.18) и вектор $\hat{x}$ ($\hat{x}^j = 0$, $j \neq \mu_i - m$) — решение задачи (5.17). В противном случае можно, используя технику симплекс-метода, найти лучший базис. Включить в него нужно переменную λ^{μ_i} такую, что $\hat{x}^{\mu_i - m} < 0$, а какую из старых базисных переменных с номерами, большими m, исключить — определяется, во-первых, их величинами и, во-вторых, связанными с ними коэффициентами замещения. Последние, по определению, представляют собой параметры системы уравнений, получающихся, если разрешить относительно базисных переменных уравнения (5.18), которые можно переписать так:

$$\begin{aligned} &\sum_{r=1}^{m} a_{r\,(\mu_i - m)}\lambda^r = c^{\mu_i - m} + \lambda^{\mu_i}, \quad i = 1, \ldots, m, \\ &\lambda^{j+m} = \sum_{r=1}^{m} a_{rj}\lambda^r - c^j, \quad 1 \leqslant j \leqslant n, \quad j+m \neq \mu_i, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 1, \ldots, m. \end{aligned} \tag{5.19}$$

Первая группа данных уравнений позволяет выразить λ^r, $r=1, \ldots, m$, через небазисные переменные λ^{μ_l}. Подстановка получающихся при этом выражений во вторую группу уравнений дает зависимость базисных переменных λ^j, $j>m$, от λ^{μ_l}. Параметры этой зависимости и есть с точностью до знака интересующие нас коэффициенты замещения.

Обозначим через A', c', λ', $\hat{\lambda}$ матрицу и векторы, составленные из столбцов a_{μ_l-m} и координат c^{μ_l-m}, λ^{μ_l}, λ^r, соответственно Тогда из первой строки в (5.19) получим

$$\hat{\lambda}=(A'^T)^{-1}(c'+\lambda'),$$

а из второй следует, что

$$\lambda''=A''^T\hat{\lambda}-c''=A''^T(A'^T)^{-1}(c'+\lambda')-c''=$$
$$=(A'^{-1}A'')^T(c'+\lambda')-c'', \qquad (5.20)$$

где A'', c'', λ'' — матрица и векторы, образованные столбцами a_j и коэффициентами c^j, λ^{j+m} с номерами $j\neq\mu_l-m$, $i=1, \ldots, m$. Таким образом, коэффициенты замещения есть элементы матрицы $-[A'^{-1}A'']$ или, что то же самое, множители в разложениях столбцов $(-a_j)$, $j\neq\mu_l-m$, $i=1, \ldots, m$, по столбцам a_{μ_l-m}. Но эти множители есть с точностью до знака элементы симплекс-таблицы задачи (5.17), отвечающей ее *недопустимому базису*, в который входят координаты с номерами μ_l-m, $i=1, \ldots, m$, причем в верхней (нулевой) строке этой таблицы будут стоять значения координат задачи (5.18) с номерами, большими m, а в крайнем левом столбце — величины $\hat{x}^{\mu_l-m}$ — оценки замещения для небазисных переменных λ^{μ_l}. Следовательно, имея данную симплекс-таблицу, мы можем указать номера координат, которые следует ввести и вывести из базиса задачи (5.18) в соответствии с симплекс-методом. Первой из них отвечает строка таблицы с номером $s\geqslant 1$, элемент которой z_{s0} отрицателен Номер второй, $m+k$, где $1\leqslant k\leqslant n$, определяется из условия

$$-\frac{z_{0k}}{z_{sk}}=\min_{\{j\,:\,z_{sj}<0\}}\left(-\frac{z_{0j}}{z_{sj}}\right) \qquad (5.21)$$

или

$$\frac{z_{0k}}{z_{sk}} = \max_{\{l \mid z_{sl} < 0\}} \frac{z_{0l}}{z_{sl}}.$$

После того как новый допустимый базис задачи (5.18) (и отвечающий ему новый, вообще говоря, недопустимый базис задачи (5.17)) найден, симплекс-таблица пересчитывается по обычным формулам (см. § 3), и процедура повторяется

Если задача (5.18) не вырождена, ее оптимальный базис и решение задачи (5.17) будут получены описанным способом за конечное число шагов. В противном случае в принципе возможно зацикливание, но, как уже было сказано ранее, это явление крайне редкое.

Представленный алгоритм является простейшей версией *двойственного симплекс-метода.* (Название последнего отражает тот факт, что поиск с его помощью оптимума в задаче (5.17) состоит в решении задачи (5.18), эквивалентной двойственной к (5.17) задаче.) На практике чаще используют другую его версию: *двойственный симплекс-метод с обратной матрицей.* При реализации ее на ЭВМ в памяти машины хранятся только матрица A'^{-1}, обратная к составленной из столбцов $a_{\mu_l - m}$ матрице A' (где μ_l — номера небазисных переменных задачи (5.18)), а также текущие значения базисных переменных λ^r, $r = 1, \ldots, m$, и оценок замещения $x^{\mu_i - m}$, $i = 1, \ldots, m$. В данном случае используется формула (5.20), в соответствии с которой величина базисной переменной λ^{j+m}, $j \neq \mu_l - m$, $i = 1, \ldots, m$, есть

$$\lambda^{j+m} = z_{0j} = \sum_{r=1}^{m} a_{rj}\lambda^r - c^j, \qquad (5.22)$$

а связанные с ней коэффициенты замещения равны

$$z_{ij} = -\sum_{r=1}^{m} a_{rj}\beta_{ir}, \quad j \neq \mu_l - m, \quad i = 1, \ldots, m, \qquad (5.23)$$

где β_{ir} — элемент матрицы A'^{-1}. После того как номер s строки, отвечающей вводимой в базис переменной λ^{μ_s}, определен, при каждом $j \geqslant 1$, $j \neq \mu_l - m$, $i = 1, \ldots, m$, по формулам (5.22), (5.23) вычисляются величины z_{0j}, z_{sj}

и из равенства (5.21) определяется номер переменной, которую следует ввести в базис. При переходе к следующей итерации матрица A'^{-1} и величины λ^r, $r=1, \ldots, m$, x^{μ_i-m}, $i=1, \ldots, m$, пересчитываются по формулам, полученным в § 4 для прямого симплекс-метода с обратной матрицей.

Чтобы приступить к решению задачи (5.17) двойственным симплекс-методом, нужно знать какой-либо допустимый базис задачи (5.18). Если специфика последней не позволяет сразу указать такой базис, исходную задачу (5.17) модифицируют, вводя дополнительные переменную x^{n+1} и ограничения

$$\sum_{i=1}^{n-m} x^{s_i} + x^{n+1} = b^{m+1}, \quad x^{n+1} \geqslant 0,$$

где b^{m+1} — большое положительное число и индексы s_i подобраны так, чтобы столбцы a_j, $j \neq s_i$, $i=1, \ldots, n-m$ были линейно независимыми. Решения такая модификация не изменит. При этом задача типа (5.18), соответствующая «расширению» исходной задачи, выглядит так:

$$\max\left(-\sum_{r=1}^{n} \lambda^r b^r - \lambda^{m+1} b^{m+1}\right),$$

$$\sum_{r=1}^{m} \lambda^r a_{rj} - \lambda^{m+1+j} = c^j, \quad j \neq s_i, \quad i=1, \ldots, n-m,$$

$$\sum_{r=1}^{m} \lambda^r a_{rj} + \lambda^{m+1} - \lambda^{m+1+j} = c^j, \quad j = s_i, \quad i=1, \ldots, n-m.$$

Легко проверить, что вектор $\bar{\lambda}$, координаты которого определяются уравнениями

$$\sum_{r=1}^{m} \lambda^r a_{rj} = c^j, \quad \bar{\lambda}^{m+1+j} = 0, \quad j \neq s_i, \quad i=1, \ldots, n-m,$$

$$\lambda^{m+1} = \max_{\substack{j=s_i, \\ i=1, \ldots, n-m}} \left(c^j - \sum_{r=1}^{m} a_{rj}\bar{\lambda}^r\right) = c^k - \sum_{r=1}^{m} a_{rk}\lambda^r,$$

$$\bar{\lambda}^{m+1+k} = 0,$$

$$\bar{\lambda}^{m+1+j} = \bar{\lambda}^{m+1} - c^j + \sum_{r=1}^{m} a_{rj}\lambda^r, \quad j = s_i, \; i = 1, \ldots, n-m$$

является ее допустимым базисным решением, причем в базис войдут переменные с номерами от 1 до $m+1$ и переменные λ^{m+1+j}, $j=s_i$, $j\neq k$. Таким образом, расширение исходной задачи указанным способом позволяет сразу выявить допустимый начальный базис для двойственного симплекс-метода. Единственная трудность, которая может здесь возникнуть, связана с необходимостью выделения m линейно независимых столбцов матрицы A. Однако, как правило, это не требует больших усилий.

В заключение данного пункта следует сказать, что двойственный симплекс-метод обычно оказывается значительно эффективнее прямого в случаях, когда число переменных исходной задачи существенно больше числа ее уравнений.

4. Прямодвойственный метод (метод последовательного сокращения невязок). И в прямом и в двойственном симплекс-методах с каждой итерацией связана пара векторов $\hat{x}$, $\hat{\lambda}$ (в методах с обратными матрицами и тот и другой просто-напросто вычисляются) таких, что значения функционалов прямой и двойственной задач на них совпадают, или, что то же самое, — векторы $\hat{x}$, $\hat{\lambda}$ удовлетворяют условиям дополняющей нежесткости. В прямом симплекс-методе вектор $\hat{x}$ есть базисное допустимое решение прямой задачи, а ограничения двойственной задачи в точках $\hat{\lambda}$ на всех итерациях, кроме последней, не выполняются. В двойственном симплекс-методе все наоборот — вектор $\hat{\lambda}$, составленный из первых m координат очередного базисного решения задачи (5.18), указывает в допустимую вершину двойственной к (5.17) задачи, а $\hat{x}$ на всех итерациях, кроме последней, хотя и подчиняется уравнениям прямой задачи, имеет отрицательные компоненты, и поэтому недопустим.

В этом пункте мы рассмотрим алгоритм, в котором, как и в двойственном симплекс-методе, перебираются допустимые (правда, не базисные) решения $\hat{\lambda}$ двойственной задачи, а точки $\hat{x}$, связанные с $\hat{\lambda}$ условиями дополняющей нежесткости, ни на одной итерации, кроме последней, ограничениям прямой задачи не удовлетворяют. Однако теперь в $\hat{x}$ будут нарушаться не условия неотрицательности, а ограничения — равенства этой задачи

Итак, пусть дана задача

$$\max\left(\sum_{j=1}^{n} c^j x^j\right), \tag{5.24}$$

$$\sum_{j=1}^{n} a_{rj} x^j = b^r, \quad r = 1, \ldots, m,$$

$$x^j \geqslant 0, \quad j = 1, \ldots, n,$$

причем все b^r неотрицательны (этого всегда можно добиться, поменяв, если нужно, знаки коэффициентов и правых частей отдельных уравнений) Пусть, кроме того, известно допустимое решение $\hat{\lambda}$ двойственной к (5.24) задачи

$$\min\left(\sum_{r=1}^{m} b^r \lambda^r\right), \tag{5.25}$$

$$\sum_{r=1}^{m} a_{rj} \lambda^r \geqslant c^j, \quad j = 1, \ldots, n.$$

Обозначим через $\hat{J}$ набор тех индексов j, для которых выполнены равенства

$$\sum_{r=1}^{m} a_{rj} \hat{\lambda}^r = c^j,$$

и рассмотрим множество векторов x, связанных с $\hat{\lambda}$ условиями дополняющей нежесткости, т. е. таких x, что

$$x^k = 0, \quad k \notin \hat{J}.$$

Если среди них найдется неотрицательный и удовлетворяющий ограничениям задачи (5.24), он будет ее решением (тогда $\hat{\lambda}$ — решение задачи (5.25)). Чтобы отыскать такой вектор или убедиться в том, что его не существует, достаточно решить (заведомо разрешимую) задачу

$$\max\left(-\sum_{r=1}^{m} x^{n+r}\right),$$

$$\sum_{j \in \hat{J}} a_{rj} x^j + x^{n+r} = b^r, \quad r = 1, \ldots, m, \tag{5.26}$$

$$x^j \geqslant 0, \quad j \in \hat{J}, \quad x^{n+r} \geqslant 0, \quad r = 1, \ldots, m.$$

Коль скоро максимальное значение критерия этой задачи окажется нулем, вектор $\hat{x}$, j-е координаты которого для

$j \in \hat{J}$ совпадают с соответствующими координатами ее решения, а остальные координаты равны нулю, и есть то, что нам нужно В противном случае в рассматриваемом множестве векторов x оптимального для задачи (5.24) нет, и если мы все же хотим искать его, решая задачу типа (5.26), необходимо «уточнить набор $\hat{J}$». Посмотрим, как это можно сделать.

Получив решение задачи (5.26) любым из описанных выше алгоритмов, мы одновременно найдем и решение $\hat{\mu}^r$, $r=1, \ldots, m$, двойственной к ней задачи вида

$$\min\left(\sum_{r=1}^{m} b^r \mu^r\right),$$

$$\sum_{r=1}^{m} a_{rj}\mu^r \geqslant 0, \quad j \in \hat{J},$$

$$\mu^r \geqslant -1, \quad r=1, \ldots, m.$$

Минимальное значение критерия последней равно максимуму критерия задачи (5.26) и в исследуемой ситуации отрицательно:

$$\sum_{r=1}^{m} b^r \hat{\mu}^r < 0. \tag{5.27}$$

Это значит, что при движении вдоль $\hat{\mu}$ критерий задачи (5.25) будет убывать Максимальный шаг $\hat{t}$, который можно сделать из точки $\hat{\lambda}$ по направлению $\hat{\mu}$ не выходя из допустимого многогранника задачи (5.25), определяется ее ограничениями с номерами $j \notin \hat{J}$. Действительно,

$$\sum_{r=1}^{m} a_{rj}\hat{\mu}^r \geqslant 0, \quad j \in \hat{J},$$

и поэтому неравенства

$$\sum_{r=1}^{m} a_{rj}(\hat{\lambda}^r + t\hat{\mu}^r) = c^j + t\sum_{r=1}^{m} a_{rj}\hat{\mu}^r \geqslant c^j, \quad j \in J,$$

выполнены при любом $t \geqslant 0$. Если окажется, что справедливы неравенства

$$\sum_{r=1}^{m} a_{rj}\hat{\mu}^r \geqslant 0, \quad j \notin \hat{J},$$

шаг $\hat{t}$ будет равен бесконечности, т. е точки $\hat{\lambda}+t\hat{\mu}$ допустимы для всех $t\geqslant 0$ Тогда, учитывая (5 27), можно утверждать, что критерий задачи (5.25) не ограничен на ее допустимом множестве и, следовательно, задача (5 24) не имеет решения Если же есть индексы j такие, что

$$\sum_{r=1}^{m} a_{rj}\hat{\mu}^r < 0,$$

то шаг $\hat{t}$ определяется по формуле

$$\hat{t} = \min_{\left\{j \sum_{r=1}^{m} a_{rj}\hat{\mu}^r < 0\right\}} \left\{ \frac{\sum_{r=1}^{m} a_{rj}\hat{\lambda}^r - c^j}{-\sum_{r=1}^{m} a_{rj}\hat{\mu}^r} \right\} > 0.$$

При этом точка $\lambda = \hat{\lambda} + \hat{t}\,\hat{\mu}$ допустима для задачи (5.25) и

$$\sum_{r=1}^{m} b^r\lambda^r = \sum_{r=1}^{m} b^r(\hat{\lambda}^r + \hat{t}\,\hat{\mu}^r) =$$

$$= \sum_{r=1}^{m} b^r\hat{\lambda}^r + \hat{t}\sum_{r=1}^{m} b^r\hat{\mu}^r < \sum_{r=1}^{m} b^r\hat{\lambda}^r,$$

т. е. $\bar{\lambda}$ «ближе» к ее решению, чем $\hat{\lambda}$. Это позволяет надеяться на то, что набор индексов J, для которых

$$\sum_{r=1}^{m} a_{rj}\lambda^r = c^j,$$

«лучше» набора $\hat{J}$ с точки зрения возможности приблизиться к оптимуму задачи (5.24), решая аналог задачи (5.26) с $\bar{J}$ вместо $\hat{J}$ В определенном смысле так оно и есть, и мы сейчас покажем это.

Обозначим через $\hat{x}^{n+r}$, $\bar{x}^{n+r}$, $r=1, \ldots, m$, соответствующие координаты решений задачи (5.26) и задачи

$$\max\left(-\sum_{r=1}^{m} x^{n+r}\right),$$

$$\sum_{j\in\bar{J}} a_{rj}x^j + x^{n+r} = b^r, \quad r=1, \ldots, n, \qquad (5.28)$$

$$x^j \geqslant 0, \quad j\in J, \quad x^{n+r}\geqslant 0, \quad r=1, \ldots, m,$$

а через μ^r, $r=1, \ldots, m$, — координаты решения двойственной к (5.28) задачи вида

$$\min\left(\sum_{r=1}^{m} b^r\mu^r\right),$$

$$\sum_{r=1}^{m} a_{rj}\mu^r \geqslant 0, \quad j\in\bar{J}, \ \mu^r\geqslant -1, \quad r=1, \ldots, m. \tag{5.29}$$

Тогда выполнены равенства

$$\sum_{r=1}^{m} b^r\bar{\mu}^r = -\sum_{r=1}^{m} \bar{x}^{n+r},$$

$$\sum_{r=1}^{m} b^r\hat{\mu}^r = -\sum_{r=1}^{m} \hat{x}^{n+r}.$$

Ясно, что вектор $\hat{\mu}$ является решением задачи

$$\min\left(\sum_{r=1}^{m} b^r\mu^r\right),$$

$$\sum_{r=1}^{m} a_{rj}\mu^r \geqslant 0, \quad j\in\hat{J}' \tag{5.30}$$

$$\mu^r\geqslant -1, \quad r=1, \ldots, m,$$

где $\hat{J}'$ — множество индексов существенных ограничений из $\hat{J}$ таких, что

$$\sum_{r=1}^{m} a_{rj}\hat{\mu}^r = 0, \quad j\in\hat{J}',$$

а из определения вектора $\bar{\lambda}$ следует вложение

$$\hat{J}'\subset\bar{J}.$$

(т. е. в задаче (5.29) больше ограничений, чем в (5.30)). Поэтому

$$-\sum_{r=1}^{m} \bar{x}^{n+r} = \sum_{r=1}^{m} b^r\bar{\mu}^r \geqslant \sum_{r=1}^{m} b^r\hat{\mu}^r = -\sum_{r=1}^{m} \hat{x}^{n+r}$$

или, что то же самое,

$$\sum_{r=1}^{m} \bar{x}^{n+r} \leqslant \sum_{r=1}^{m} \hat{x}^{n+r}. \tag{5.31}$$

Таким образом, решив задачу (5.28), мы получим значение ее критерия, а попросту говоря, сумму невязок ограничений исходной задачи (5 24) не большую, чем для задачи (5.26) При этом оптимальные базисы задач (5.26), (5.28) будут допустимыми базисами задачи

$$\max\left(-\sum_{r=1}^{m} x^{n+r}\right),$$

$$\sum_{j=1}^{n} a_{rj}x^j + x^{n+r} = b^r, \quad r = 1, \ldots, m, \qquad (5.32)$$

$$x^j \geqslant 0, \quad j = 1, \ldots, n+m,$$

и если она не вырождена, неравенство в (5.31) будет строгим. Последнее, в свою очередь, гарантирует сходимость процесса поиска решения задачи (5.24), на каждом шаге которого решается задача вида (5.26) и множество $\hat{J}$ переопределяется описанным выше способом, за конечное число шагов. Этот процесс и называется *методом сокращения невязок* (в названии отражено неравенство (5.31)) Если задача (5.32) вырождена, метод может зациклиться, что, однако, случается крайне редко

На этом мы закончим изучение задач линейного программирования и перейдем к значительно более сложным задачам, в которых целевая функция и функции ограничений нелинейны. В заключение хотелось бы только добавить, что машинная реализация представленных методов решения линейных задач проста и надежна лишь в тех случаях, когда число ограничений невелико. Чем больше это число, тем большую роль будут играть ошибки округления, неизбежные при машинном счете, и тем менее надежными будут программы, построенные на основе описанных схем Здесь нужны специальные модификации, изложение которых выходит за рамки данной книги.

Глава IV

ТЕОРИЯ ЭКСТРЕМУМА В НЕЛИНЕЙНЫХ ЗАДАЧАХ С ОГРАНИЧЕНИЯМИ

Введение

В данной главе рассматриваются аналитические свойства оптимизационных задач, значительно более сложных, чем те, о которых шла речь в предшествующих главах. Это задачи поиска экстремума нелинейной функции на множестве точек, удовлетворяющих нелинейным неравенствам Мы получим необходимые условия такого экстремума, названные (в честь впервые установивших их авторов) условиями Куна — Таккера. Эти условия являются обобщением правила множителей Лагранжа (см. главу I), опираясь на которое их доказать совсем несложно. С другой стороны, их можно рассматривать как частный результат общей теории локальных экстремумов. Именно такая точка зрения принята в настоящей книге.

Теория локальных экстремумов применима для исследования оптимизационных задач как в конечномерных случаях, так и для задач поиска экстремума в функциональных пространствах. Разумеется, изложение ее в самом общем виде, требующее привлечения сложного аппарата современного функционального анализа, выходит далеко за рамки нашего курса Однако исходные идеи этой теории весьма прозрачны и их строгое изложение для конечномерных задач, которым посвящена данная книга, достаточно компактно и легко для восприятия. Оно приводится ниже *), и условия экстремума для задач с ограничениями будут получены как следствие основной теоремы Милютина — Дубовицкого. Доказательство последней опирается на элементарные сведения из теории выпуклых множеств. Краткое изложение этой теории дано в первом параграфе настоящей главы. Свойства выпуклых множеств и выпуклых функций, описанные в первых двух парагра-

*) При изложении теории Милютина — Дубовицкого мы, помимо работ этих авторов, широко использовали книгу [5].

фах, представляют для нас большой интерес также в связи с тем, что они позволяют строить эффективные алгоритмы решения задач минимизации, в которых целевая функция и допустимое множество выпуклы.

§ 1. Выпуклые множества и конусы

1. Простейшие свойства выпуклых множеств. Определение выпуклого множества уже было дано в § 3 предыдущей главы. Напомним, что множество X называется *выпуклым*, если для любых точек x_1, x_2 из X и любых $\lambda_1 \geqslant 0$, $\lambda_2 \geqslant 0$, $\lambda_1 + \lambda_2 = 1$ точка $z = \lambda_1 x_1 + \lambda_2 x_2$ принадлежит X.

Элементарные свойства выпуклых множеств, которые читатель без труда докажет сам, таковы: если X_i, $i = 1, \ldots, s$, — выпуклые множества, то их пересечение $\bigcap_{i=1}^{s} X_i$ и сумма $\sum_{i=1}^{s} X_i$ тоже выпуклы (напомним, что первое представляет собой множество точек x, принадлежащих одновременно всем X_i, $i = 1, \ldots, s$, а вторая — множество x таких, что $x = \sum_{i=1}^{s} x_i$, где $x_i \in X_i$, $i = 1, \ldots, s$). Кроме того, используя метод математической индукции, легко показать, что линейная комбинация $z = \sum_{i=1}^{k} \lambda_i x_i$ точек x_i из выпуклого множества X с коэффициентами $\lambda_i \geqslant 0$, $i = 1, \ldots, k$, $\sum_{i=1}^{k} \lambda_i = 1$ также принадлежит X.

Центральное место в теории выпуклых множеств занимают так называемые теоремы отделимости. Для их доказательства нам потребуется понятие проекции точки на замкнутое множество.

Определение 1.1. *Проекция* точки y на замкнутое множество X — это точка $x_y \in X$ такая, что расстояние от нее до y не больше расстояния до y от любой другой точки $x \in X$, т. е.

$$\|x_y - y\| \leqslant \|x - y\|, \tag{1.1}$$

если только $x \in X$ (рис. 1.1). (Здесь нормой $\|a\|$ произвольного вектора $a \in E_n$ считается $\sqrt{\sum_{i=1}^{n} (a^i)^2}$) В общем случае проекций x_y может быть несколько, но когда множество X выпукло, каждой точке y отвечает только одна проекция. Это устанавливает следующая

Лемма 1.1. *Пусть X — замкнутое выпуклое множество, а y — некоторая точка. Тогда существует точка $x_y \in X$ такая, что для любой другой точки $x \in X$ выполнено неравенство*

$$\|x_y - y\| < \|x - y\|. \qquad (1.2)$$

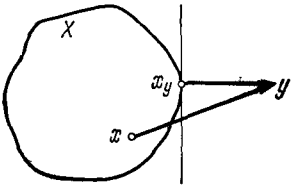


Рис. 1.1.

Доказательство. Возьмем какую-нибудь точку $x' \in X$ и рассмотрим множество X' точек из X, отстоящих от y не далее чем x', т. е.

$$X' = X \cap \{x : \|x - y\| \leqslant \|x' - y\|\}.$$

Это множество непусто (так как содержит x'), замкнуто и ограничено. Поэтому непрерывная функция $f(x) = \|y - x\|$ достигает на нем своего минимума, а, проще говоря, найдется точка $x_y \in X' \subset X$ такая, что из принадлежности x множеству X' следует неравенство

$$\|x_y - y\| \leqslant \|x - y\|. \qquad (1.3)$$

Понятно, что это неравенство справедливо и для точек $x \notin X'$, $x \in X$, так как для них по построению множества X' имеем

$$\|x - y\| \geqslant \|x' - y\| \geqslant \|x_y - y\|.$$

Таким образом, неравенство (1.3) выполнено для всех $x \in X$. Покажем теперь, что оно будет строгим. Доказательство проведем от противного: допустим, что существует точка $x^* \in X$, $x^* \neq x_y$ такая, что

$$\|x_y - y\| = \|x^* - y\|. \qquad (1.4)$$

Тогда

$$\left\|\frac{1}{2}x^*+\frac{1}{2}x_y-y\right\|^2=\left\|\left(\frac{1}{2}x^*-\frac{1}{2}y\right)+\left(\frac{1}{2}x_y-\frac{1}{2}y\right)\right\|^2=$$
$$=\left\|\frac{1}{2}x^*-\frac{1}{2}y\right\|^2+\left\|\frac{1}{2}x_y-\frac{1}{2}y\right\|^2+\frac{1}{2}(x^*-y,\ x_y-y)=$$
$$=\frac{1}{2}\|x_y-y\|^2+\frac{1}{2}\left(\left(\frac{1}{2}x^*+\frac{1}{2}x_y-y\right)+\right.$$
$$\left.+\left(\frac{1}{2}x^*-\frac{1}{2}x_y\right),\ \left(\frac{1}{2}x^*+\frac{1}{2}x_y-y\right)-\left(\frac{1}{2}x^*-\frac{1}{2}x_y\right)\right)=$$
$$=\frac{1}{2}\|x_y-y\|^2+\frac{1}{2}\left\|\frac{1}{2}x^*+\frac{1}{2}x_y-y\right\|^2-\frac{1}{2}\left\|\frac{1}{2}x^*-\frac{1}{2}x_y\right\|^2,$$

откуда

$$\left\|\frac{1}{2}x^*+\frac{1}{2}x_y-y\right\|^2=\|x_y-y\|^2-\frac{1}{4}\|x^*-x_y\|^2<\|x_y-y\|^2.$$

Но в силу выпуклости множества X точка $\frac{1}{2}x^*+\frac{1}{2}x_y$ принадлежит ему и, следовательно, полученное неравенство противоречит (1.3). Лемма доказана.

Кроме утверждения о существовании и единственности проекции x_y произвольной точки y на замкнутое выпуклое множество X нам понадобится в дальнейшем тот факт, что вектор $y-x_y$ составляет тупой угол с любым из векторов вида $x-x_y$, где $x\in X$ (рис. 1 2).

Лемма 1 2 *Точка x_y будет проекцией точки y на замкнутое выпуклое множество X в том и только в том случае, если для любой точки $x\in X$ справедливо неравенство*

$$(x-x_y,\ y-x_y)\leqslant 0 \tag{1.5}$$

Доказательство Пусть x_y — проекция y на X и x — некоторая точка из X. Тогда при любом $\lambda\in[0,\ 1]$ точка $z=(1-\lambda)x_y+\lambda x$ принадлежит X и, по определению проекции,

$$\|y-x_y\|^2\leqslant\|y-z\|^2.$$

Это неравенство можно переписать так:

$$\|y-x_y\|^2\leqslant\|(y-x_y)-\lambda(x-x_y)\|^2=$$
$$=\|y-x_y\|^2+\lambda^2\|x-x_y\|^2-2\lambda(x-x_y,\ y-x_y).$$

Отсюда следует, что

$$\lambda^2\|x-x_y\|^2-2\lambda(x-x_y,\ y-x_y)\geqslant 0$$

для любых $\lambda \in [0, 1]$, а это возможно лишь тогда, когда

$$(x - x_y,\ y - x_y) \leqslant 0.$$

Необходимость доказана Достаточность установить еще проще. Действительно, пусть неравенство (1.5) выполнено

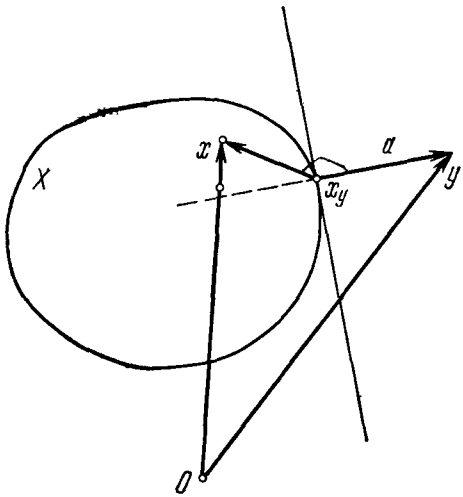


Рис. 1.2.

для некоторой точки $x_y \in X$ при любых $x \in X$. Тогда для $x \in X$ имеем

$$\|x - y\|^2 = \|(x - x_y) + (x_y - y)\|^2 =$$
$$= \|x - x_y\|^2 + 2(x - x_y,\ x_y - y) + \|x_y - y\|^2 \geqslant \|x_y - y\|^2,$$

что соответствует определению проекции

2. Теоремы отделимости. Леммы предыдущего пункта позволяют доказать целый ряд очень полезных утверждений, называемых теоремами отделимости Начнем с простейшей.

Теорема 1.1. *Пусть X — замкнутое выпуклое множество, а y — внешняя по отношению к нему точка Тогда существуют положительное число ε и вектор a такие, что* $\|a\| = 1$ *и*

$$(a,\ x) \leqslant (a,\ y) - \varepsilon \qquad (1.6)$$

при любых $x \in X$.

Доказательство Поскольку точка y не принадлежит множеству X, ее проекция x_y на X не совпадает с y, т. е. $\|y-x_y\|\neq 0$. Соответственно, можно ввести вектор

$$a=\frac{(y-x_y)}{\|y-x_y\|},\quad \|a\|=1,$$

для которого имеем

$$(a,\ y-x_y)=\|y-x_y\|. \tag{1.7}$$

Кроме того, в силу леммы 1.2 при любом $x\in X$ будет выполнено неравенство

$$(a,\ x-x_y)=\frac{(x-x_y,\ y-x_y)}{\|y-x_y\|}\leqslant 0. \tag{1.8}$$

Вычитая (1.8) из (1.7), получим

$$(a,\ y)-(a,\ x)\geqslant\|y-x_y\|,$$

что эквивалентно (1.6) при $\varepsilon=\|y-x_y\|>0$. Теорема доказана.

Установленный факт называют *сильной отделимостью* замкнутого выпуклого множества X от не принадлежащей ему точки y. Термин «отделимость» отражает геометрическую суть неравенства (1.6), которое показывает, что можно построить гиперплоскость (например, с уравнением $(a,\ x-x_y)=0$) такую, что точка y и множество X окажутся лежащими по разные стороны от нее (см. рис. 1.2). Эпитет «сильная» означает, что расстояние между точками y и $x\in X$ всегда больше некоторого положительного числа. Легко понять, что незамкнутое выпуклое множество X тоже сильно отделимо от точки $y\notin X$, если только она не является граничной для X, т. е. не принадлежит его замыканию $\bar{X}$. Действительно, в этом случае по теореме 1.1 мы можем подобрать a и ε такие, что неравенство (1.6) будет выполняться для любого $x\in\bar{X}$ и, в частности, для всех $x\in X\subset\bar{X}$

Когда точка y находится на границе выпуклого множества X (замкнутого или нет — безразлично), сильно отделить ее от X нельзя, а просто отделить, т е. построить проходящую через y гиперплоскость так, чтобы все множество X лежало по одну ее сторону, — можно. Формально это означает, что справедлива

Теорема 1.2. *Пусть точка y является граничной для выпуклого множества X. Тогда существует вектор a такой, что* $\|a\|=1$ *и*

$$(a,\ x)\leqslant(a,\ y) \tag{1.9}$$

при любом $x\in X$.

Доказательство. Так как y — граничная точка множества X, можно построить последовательность внешних по отношению к его замыканию $\bar{X}$ точек y_i, сходящуюся к y. Для каждой y_i в силу теоремы 1.1 найдутся вектор a_i, $\|a_i\|=1$, и положительное число ε_i, при которых нер венство

$$(a_i,\ x)\leqslant(a_i,\ y_i)-\varepsilon_i$$

и, соответственно, неравенства

$$(a_i,\ x)<(a_i,\ y) \tag{1.10}$$

справедливы для всех $x\in\bar{X}$, и в том числе для $x\in X$. Без ограничения общности предположим, что последовательность векторов a_i сходится к некоторому пределу a (иначе мы просто выбрали бы из нее сходящуюся подпоследовательность, а не попавшие в нее векторы a_i отбросили бы). Тогда, поскольку $\|a_i\|=1$, $i=1,\ 2,\ \ldots$, должно быть $\|a\|=1$, а переходя в (1.10) к пределу при $i\to\infty$ для каждого $x\in X$, получим неравенство

$$(a,\ x)\leqslant(a,\ y).$$

Теорема доказана.

Коль скоро выпуклое множество X открыто, утверждение теоремы 1.2 можно усилить, так как равенство в (1.9) при этом невозможно. Действительно, из равенства

$$(a,\ x)=(a,\ y)$$

при некотором x из открытого множества X следовало бы существование $x'\in X$ такого, что

$$(a,\ x')>(a,\ y).$$

Чтобы убедиться в этом, достаточно взять $x'=x+\delta\cdot a$, где δ — положительное число, причем настолько малое по модулю, что $x+\delta\cdot a\in X$. Таким образом, точка y, принадлежащая границе открытого выпуклого множества X, *строго отделима* от X: существует вектор a, $\|a\|=1$, такой, что для любого $x\in X$

$$(a,\ x)<(a,\ y).$$

Перейдем теперь к теоремам об отделимости двух выпуклых множеств.

Теорема 1.3. *Пусть X и Y — выпуклые множества, не имеющие общих точек. Тогда существует вектор a, $\|a\|=1$, такой, что для любых $x\in X$, $y\in Y$ выполнено неравенство*

$$(a,\ x)\leqslant(a,\ y).$$

Доказательство. Рассмотрим множество Z точек вида $z=x-y$, где $x\in X$, $y\in Y$. Легко показать, что оно выпукло. Действительно, пусть z_1 и z_2 — произвольные точки, принадлежащие Z. Тогда $z_1=x_1-y_1$, $z_2=x_2-y_2$, где x_1, x_2 и y_1, y_2 — некоторые пары точек из множеств X, Y, соответственно. Линейная комбинация z_1 и z_2 с коэффициентами $\lambda_1\geqslant 0$, $\lambda_2\geqslant 0$, $\lambda_1+\lambda_2=1$ выражается через x_1, x_2, y_1, y_2 так:

$$\lambda_1z_1+\lambda_2z_2=\lambda_1x_1+\lambda_2x_2-\lambda_1y_1-\lambda_2y_2=\tilde{x}-\tilde{y},$$

где

$$\tilde{x}=\lambda_1x_1+\lambda_2x_2,\quad \tilde{y}=\lambda_1y_1+\lambda_2y_2.$$

Поскольку множества X, Y выпуклы, точка $\tilde{x}$ принадлежит первому из них, а $\tilde{y}$ — второму. Отсюда, в свою очередь, следует, что $\lambda_1z_1+\lambda_2z_2=\tilde{x}-\tilde{y}$ принадлежит множеству Z, т. е. оно тоже выпукло.

По условию теоремы у множеств X, Y нет общих точек. Это значит, что множество Z не содержит начала координат — точки нуль. Поэтому последнюю в силу теоремы 1.2 можно отделить от Z, т. е. найти вектор a, $\|a\|=1$, такой, что

$$(a,\ z)\leqslant(a,\ 0)=0$$

при любом $z\in Z$. Вспоминая определение множества Z, отсюда получаем, что

$$(a,\ x)\leqslant(a,\ y)$$

при любых $x\in X$, $y\in Y$, что и требовалось доказать.

Итак, два произвольных непересекающихся выпуклых множества отделимы друг от друга. Если одно из них открыто, они отделимы строго (это устанавливается точно так же, как была показана строгая отделимость открытого выпуклого множества от непринадлежащей ему точки) Если же оба они замкнуты, и хотя бы одно из них огра-

ничено, их можно отделить сильно (см. рис. 1.3), т. е. справедлива

Теорема 1.4 *Пусть X и Y — выпуклые замкнутые множества, пересечение которых пусто, и пусть, кроме того, множество X ограничено. Тогда существуют вектор a, $\|a\|=1$, и число $\varepsilon>0$ такие, что*

$$(a,\ x)\leqslant(a,\ y)-\varepsilon$$

при любых $x\in X$, $y\in Y$.

Доказательство. Рассмотрим, как и в предыдущей теореме, множество Z точек $z=x-y$, где $x\in X$,

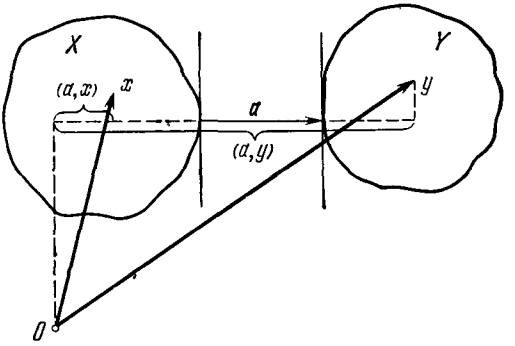


Рис 1 3.

$y\in Y$. То, что оно выпукло, мы уже знаем. Покажем, что оно еще и замкнуто. Действительно, пусть $\{z_i\}$, $i=1$, 2, . ., сходящаяся, и потому ограниченная последовательность точек $z_i\in Z$. При этом $z_i=x_i-y_i$, $i=1, 2, \ldots$, где $x_i\in X$, $y_i\in Y$. В силу ограниченности $\{z_i\}$, $i=1, 2, \ldots$, и множества X последовательности $\{x_i\}$, $\{y_i\}$, $i=1, 2, \ldots$, тоже будут ограниченными Поэтому, не умаляя общности, можно считать, что у них есть пределы (иначе мы перешли бы к сходящимся подпоследовательностям) $\tilde{y}=\lim\limits_{i\to\infty} y_i$, $\tilde{x}=\lim\limits_{i\to\infty} x_i$ Тогда предел $\tilde{z}$ последовательности $\{z_i\}$, $i=1$, 2, ..., есть разность этих пределов

$$\tilde{z}=\lim_{i\to\infty} z_i=\lim_{i\to\infty}(x_i-y_i)=\lim_{i\to\infty} x_i-\lim_{i\to\infty} y_i=\tilde{x}-\tilde{y},$$

а отсюда, поскольку множества X, Y замкнуты и, соответственно, точка $\tilde{x}$ принадлежит первому, а $\tilde{y}$ — второму

из них, следует, что $\tilde{z}$ принадлежит Z. Таким образом, множество Z содержит свои предельные точки, т. е. является замкнутым.

У множеств X, Y нет общих точек и, следовательно, выпуклое замкнутое множество Z не содержит нуля. Значит, по теореме 1.1 его можно сильно отделить от нуля, т. е. найти вектор a, $\|a\|=1$, и число $\varepsilon>0$ такие, что

$$(a,\ z)\leqslant(a,\ 0)-\varepsilon=-\varepsilon$$

для любого $z\in Z$ или, что то же самое,

$$(a,\ x)\leqslant(a,\ y)-\varepsilon$$

для любых $x\in X$, $y\in Y$. Теорема доказана.

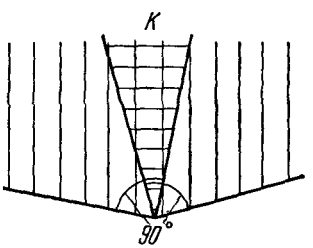


Рис. 1.4.

Если снять требование ограниченности одного из замкнутых выпуклых непересекающихся множеств, сильную отделимость их друг от друга гарантировать нельзя. Рассмотрим простой

Пример 1.1. Пусть X, Y — подмножества двумерного евклидова пространства:

$$X=\left\{x: x^1\geqslant 0,\quad x^2\geqslant\frac{1}{x^1}\right\},$$
$$Y=\{x: -\infty\leqslant x^1\leqslant+\infty,\ x^2=0\}.$$

Они выпуклы, замкнуты и не имеют общих точек, но из-за того, что и X и Y — неограниченные множества, их можно отделить только строго, а сильно отделить X от Y не удается.

3. Выпуклые конусы. Особую роль в теории экстремума при наличии ограничений играют выпуклые множества специального вида, а именно — выпуклые конусы. Дадим соответствующее

Определение 1.2. *Конусом* называется множество K, содержащее вместе с любой своей точкой x всевозможные точки $\lambda\cdot x$, где $0<\lambda<+\infty$.

Если такое множество выпукло, его называют выпуклым конусом, если замкнуто — замкнутым конусом, и так далее. Конус, имеющий внутренние точки (т. е. точки,

принадлежащие ему вместе с некоторыми своими окрестностями), будем называть *телесным*.

Центральным является понятие *двойственного* или *сопряженного* конуса. Вводится оно следующим образом. Рассмотрим множество K^* точек a таких, что для любого x из некоторого конуса K выполнено неравенство

$$(a,\ x) \geqslant 0.$$

Поскольку умножение на положительное число знака этого неравенства не изменит, ясно, что K^* — конус (возможно, состоящий из одной точки — нуля). Его-то и называют *двойственным* к K конусом. Данное определение проиллюстрировано рисунком 1.4, где горизонтальной штриховкой выделен исходный, а вертикальной — двойственный конусы.

Двойственные конусы обладают целым рядом интересных качеств. В частности, конус K^* является выпуклым и замкнутым независимо от того, обладает теми же свойствами исходный конус K или нет, причем

$$K^* = \bar{K}^*,$$

т. е. замыкание исходного конуса K никоим образом не отражается на двойственном. Доказательства этих утверждений очень просты, и мы предлагаем читателю проделать их самостоятельно, в качестве упражнений. Столь же просто показать, что сумма и пересечение нескольких конусов тоже будут конусами. Нам потребуются и более сложные свойства конусов. Первое из них устанавливает

Лемма 1.3. *Пусть K — замкнутый выпуклый конус. Тогда*

$$K^{**} = K.$$

Доказательство. Легко проверить, что любая точка из K принадлежит K^{**}. Действительно, возьмем произвольную точку x' из K. По определению двойственного конуса K^*, какова бы ни была точка $a \in K^*$, неравенство $(a,\ x) \geqslant 0$ будет справедливо для всех $x \in K$ и, в частности, для $x = x'$. Таким образом, $(a,\ x') \geqslant 0$ для всех $a \in K^*$. Но это и означает, что x' содержится в конусе K^{**}.

Покажем теперь, что никаких иных точек, кроме принадлежащих K, конус K^{**} в рассматриваемом случае включать не может. Допустим противное, т. е. предположим, что нашлась точка $y \in K^{**}$, $y \notin K$. Тогда, поскольку

множество K выпукло и замкнуто, y можно строго отделить от K (больше того — мы знаем, что точка y сильно отделима от K; однако нам достаточно строгой отделимости), т. е. найдется вектор a, $\|a\|=1$, такой что

$$(a,\ x)<(a,\ y) \tag{1.11}$$

при любых $x\in K$. Поскольку множество K наряду с x содержит все точки λx, $\lambda>0$, это неравенство возможно лишь в случае, когда

$$(a,\ x)\leqslant 0,$$

и, следовательно, точка $-a$ принадлежит K^*. Значит, для y как точки конуса K^{**} должно выполняться неравенство

$$(-a,\ y)\geqslant 0. \tag{1.12}$$

В то же время, полагая в (1.11) $x=0$ (что можно сделать, так как конус K замкнут и, соответственно, включает нуль), получим

$$(a,\ y)>0.$$

Это противоречит (1.12). Значит, исходная посылка неверна: точки $y\in K^{**}$, $y\notin K$, не существует. Лемма доказана.

Далее нам понадобятся еще два вспомогательных утверждения.

Лемма 1.4 *Конус, двойственный к сумме конусов K_1 и K_2 (т. е. к конусу, состоящему из всевозможных сумм вида x_1+x_2, где $x_1\in K_1$, $x_2\in K_2$), есть пересечение двойственных к K_1 и K_2 конусов:*

$$(K_1+K_2)^*=K_1^*\cap K_2^*.$$

Доказательство. Пусть точка a принадлежит $(K_1+K_2)^*$. Это значит, что

$$(a,\ x_1+x_2)\geqslant 0$$

при любых $x_1\in K_1$, $x_2\in K_2$. Рассмотрев наряду с x_1, x_2 пары точек $\lambda x_1\in K_1$, x_2 и x_1, $\lambda x_2\in K_2$, где $\lambda>0$, и устремляя λ к бесконечности, отсюда получим, что

$$(a,\ x_1)\geqslant 0,\quad (ax_2)\geqslant 0$$

при всех $x_1\in K_1$, $x_2\in K_2$, т. е. $a\in K_1^*\cap K_2^*$.

Пусть теперь a — некоторая точка пересечения $K_1^*\cap K_2^*$. Тогда для любых $x_1\in K_1$, $x_2\in K_2$ выполнены неравенства

$$(a,\ x_1)\geqslant 0,\quad (a,\ x_2)\geqslant 0.$$

Складывая их, получим, что

$$(a,\ x_1+x_2)\geqslant 0,$$

если только $x_1\in K_1$, $x_2\in K_2$. Следовательно, a принадлежит конусу $(K_1+K_2)^*$. Таким образом, точек, принадлежащих одному из конусов $(K_1+K_2)^*$, $K_1^*\cap K_2^*$ и не принадлежащих другому, не существует. Лемма доказана.

Лемма 1.5. *Сумма двух замкнутых конусов K_1 и K_2 таких, что равенство*

$$x_1=-x_2,\quad x_1\in K_1,\quad x_2\in K_2,$$

возможно лишь при $x_1=x_2=0$, замкнута.

Доказательство. Рассмотрим сходящуюся к некоторому пределу $\bar z$ последовательность точек z_i, $i=1, 2, \ldots$, конуса K_1+K_2, т. е.

$$z_i=x_i+y_i,\quad x_i\in K_1,\quad y_i\in K_2,\quad i=1,\ 2,\ \ldots,$$

$$\lim_{i\to\infty} z_i=\lim_{i\to\infty}(x_i+y_i)=\bar z.$$

Покажем, что последовательности $\{x_i\}$, $\{y_i\}$, $i=1, 2, \ldots$, ограничены. Действительно, в противном случае, не ограничивая общности, можно считать, что

$$\lim_{i\to\infty}\|x_i\|=+\infty,$$

и существует $\lim\limits_{i\to\infty}\dfrac{x_i}{\|x_i\|}=\bar x$. Тогда

$$\lim_{i\to\infty}\left(\frac{x_i}{\|x_i\|}+\frac{y_i}{\|x_i\|}\right)=\lim_{i\to\infty}\frac{z_i}{\|x_i\|}=0,$$

откуда следует, что

$$\lim_{i\to\infty}\frac{y_i}{\|x_i\|}=-\bar x.$$

Точки $\dfrac{x_i}{\|x_i\|}$ принадлежат конусу K_1, а точки $\dfrac{y_i}{\|x_i\|}$ — конусу K_2. Так как оба они замкнуты, предельные точки $\bar x$, $-\bar x$ также содержатся в K_1, K_2, соответственно, причем, по построению, $\|\bar x\|=1$. Но это в силу условий леммы невозможно. Следовательно, предположение о неограниченности, на основании которого мы пришли к противоречию, неверно.

Итак, последовательности $\{x_i\}$, $\{y_i\}$, $i=1, 2, \ldots$, ограничены. Значит, из них можно выделить сходящиеся

к некоторым $\bar{x} \in K_1$, $\bar{y} \in K_2$ подпоследовательности $\{x_{i_s}\}$, $\{y_{i_s}\}$, $s = 1, 2, \ldots$ При этом должно выполняться равенство

$$\bar{z} = \bar{x} + \bar{y},$$

т. е. $\bar{z}$ принадлежит $K_1 + K_2$. Таким образом, конус $K_1 + K_2$ содержит свои предельные точки, что и требовалось доказать.

Отметим, что обобщить лемму 1.5 для замкнутых выпуклых множеств произвольной природы нельзя. Так, в примере 1.1, приведенном в конце предыдущего пункта, сумма множеств X, Y, на которые утверждение леммы не распространяется только потому, что X — не конус, не замкнута: множество $X + Y$ не содержит нуля, хотя нуль и является его предельной точкой.

Приведенные в данном пункте леммы позволяют доказать следующую теорему.

Теорема 1.5. *Пусть K_1, K_2 — выпуклые конусы, причем пересечение их представляет собой телесный конус. Тогда справедливо равенство*

$$(K_1 \cap K_2)^* = K_1^* + K_2^*.$$

Доказательство. Конусы, двойственные к пересечению $K_1 \cap K_2$ и к замыканию этого пересечения $\overline{K_1 \cap K_2}$, совпадают. Используя теорему об отделимости выпуклых множеств, нетрудно показать, что замыкание телесного конуса $K_1 \cap K_2$ и пересечение замыканий конусов K_1 и K_2 — это одно и то же. Поэтому, учитывая лемму 1.3, можно утверждать, что

$$(K_1 \cap K_2)^* = (\overline{K_1 \cap K_2})^* = (\bar{K}_1 \cap K_2)^* = (\bar{K}_1^{**} \cap \bar{K}_2^{**})^* = \\ = (K_1^{**} \cap K_2^{**})^*.$$

Кроме того, в силу леммы 1.4 имеем

$$K_1^{**} \cap K_2^{**} = (K_1^* + K_2^*)^*,$$

т. е.

$$(K_1 \cap K_2)^* = (K_1^* + K_2^*)^{**}. \qquad (1.13)$$

Конусы K_1^*, K_2^* замкнуты, причем, как легко проверить, ненулевого вектора a такого, что

$$a \in K_1^*, \quad -a \in K_2^*,$$

не существует. Действительно, в противном случае для

любой точки x пересечения $K_1 \cap K_2$ должны были бы выполняться неравенства

$$(a, x) \geqslant 0 \qquad (x \in K_1, \qquad a \in K_1^*),$$
$$-(a, x) \geqslant 0 \qquad (x \in K_2, \quad -a \in K_2^*)$$

или, что то же самое,

$$(a, x) = 0$$

для всех $x \in K_1 \cap K_2$. Но это невозможно, так как $K_1 \cap K_2$ — телесный конус.

Таким образом, конусы K_1^*, K_2^* удовлетворяют условиям леммы 1.5. Значит, их сумма $K_1^* + K_2^*$ замкнута, а точнее — представляет собой замкнутый выпуклый конус. Поэтому по лемме 1.3

$$(K_1^* + K_2^*)^{**} = K_1^* + K_2^*,$$

откуда с учетом (1.13) видим, что

$$(K_1 \cap K_2)^* = K_1^* + K_2^*.$$

Теорема доказана.

В дальнейшем, при выводе необходимых условий экстремума, будет использоваться очевидное *следствие теоремы* 1.5: если K_i, $i = 1, 2, \ldots, s$, — выпуклые конусы и их пересечение $\bigcap_{i=1}^{s} K_i$ телесно, выполнено равенство

$$\left(\bigcap_{i=1}^{s} K_i\right)^* = \sum_{i=1}^{s} K_i^*. \tag{1.14}$$

На этом мы закончим изучение выпуклых множеств и перейдем к выпуклым функциям.

§ 2. Выпуклые функции и опорные функционалы

1. Определение выпуклых функций и их основные свойства. Рассмотрим функцию одной переменной $y = f(x)$, заданную на всей вещественной оси. Точки $\{x_1, f(x_1)\}$ и $\{x_2, f(x_2)\}$ в плоскости x, y соединим прямолинейным отрезком. Если, независимо от выбора x_1, x_2, этот отрезок лежит над графиком функции $f(x)$, будем говорить, что она *выпукла* (рис. 2.1). Координаты $\tilde{x}$, $\tilde{y}$ произвольной точки отрезка прямой, соединяющей $\{x_1, f(x_1)\}$ и $\{x_2, f(x_2)\}$,

вычисляются по формулам

$$\tilde{x} = \lambda_1 x_1 + \lambda_2 x_2, \quad \tilde{y} = \lambda_1 f(x_1) + \lambda_2 f(x_2),$$

где $\lambda_1 \geqslant 0$, $\lambda_2 \geqslant 0$, $\lambda_1 + \lambda_2 = 1$. Соответственно, формальное определение выпуклой функции, не зависящее от размерности ее аргумента, звучит так:

Определение 2.1. Функция $f(x)$ называется *выпуклой* на E_n, если для произвольных значений ее аргу-

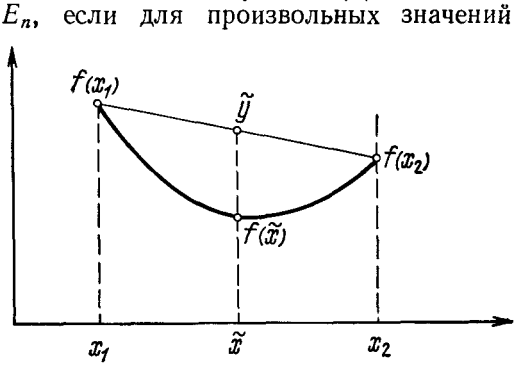


Рис. 2.1.

мента x_1, x_2 и при любых $\lambda_1 \geqslant 0$, $\lambda_2 \geqslant 0$, $\lambda_1 + \lambda_2 = 1$ выполнено неравенство

$$f(\lambda_1 x_1 + \lambda_2 x_2) \leqslant \lambda_1 f(x_1) + \lambda_2 f(x_2). \qquad (2.1)$$

Если же для некоторой функции справедливо неравенство с обратным знаком, ее называют *вогнутой*.

Легко проверить, что линейная функция $f(x) = (c, x)$ и квадратичная функция $f(x) = (Gx, x) + (c, x) + d$, где G — неотрицательно определенная матрица, выпуклы. Первая одновременно является и вогнутой.

Нетрудно убедиться также, что сумма двух выпуклых (вогнутых) функций выпукла (вогнута). Действительно, пусть

$$F(x) = f(x) + \varphi(x),$$

где $f(x)$, $\varphi(x)$ — выпуклые функции. Тогда при любых x_1, x_2, $\lambda_1 \geqslant 0$, $\lambda_2 \geqslant 0$, $\lambda_1 + \lambda_2 = 1$ имеем

$$F(\lambda_1 x_1 + \lambda_2 x_2) = f(\lambda_1 x_1 + \lambda_2 x_2) + \varphi(\lambda_1 x_1 + \lambda_2 x_2) \leqslant$$
$$\leqslant \lambda_1 f(x_1) + \lambda_2 f(x_2) + \lambda_1 \varphi(x_1) + \lambda_2 \varphi(x_2) = \lambda_1 F(x_1) + \lambda_2 F(x_2),$$

что и требовалось доказать. Отсюда по индукции следует, что сумма любого числа выпуклых (вогнутых) функций выпукла (вогнута).

Методом математической индукции можно установить также справедливость для выпуклой функции $f(x)$ при любых x_i, $\lambda_i \geqslant 0$, $i = 1, \ldots, k$, $\sum_{i=1}^{k} \lambda_i = 1$ неравенства

$$f\left(\sum_{i=1}^{k} \lambda_i x_i\right) \leqslant \sum_{i=1}^{k} \lambda_i f(x_i). \tag{2.2}$$

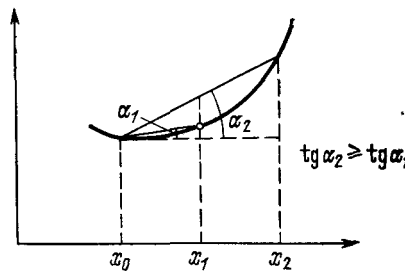


Рис. 2 2

Выпуклость в значительной степени определяет свойства гладкости функции. Мы покажем, что заданная на всем пространстве E_n выпуклая функция имеет в любой его точке производную по любому направлению. Для этого нам потребуются два вспомогательных утверждения.

Лемма 2.1. *Пусть $f(x)$ — выпуклая функция скалярного аргумента x и $x_0 < x_1 < x_2$. Тогда*

$$\frac{f(x_2)-f(x_0)}{x_2-x_0} \geqslant \frac{f(x_1)-f(x_0)}{x_1-x_0}. \tag{2.3}$$

Доказательство. Возьмем (рис. 2.2)

$$\lambda_1 = \frac{x_1-x_0}{x_2-x_0} > 0, \quad \lambda_2 = 1-\lambda_1 = 1-\frac{x_1-x_0}{x_2-x_0} = \frac{x_2-x_1}{x_2-x_0} > 0.$$

При этом

$$\lambda_1 x_2 + \lambda_2 x_0 = \frac{x_1-x_0}{x_2-x_0} x_2 + x_0 - \frac{x_1-x_0}{x_2-x_0} x_0 =$$
$$= \frac{x_1x_2 - x_0x_2 + x_0x_2 - x_0^2 - x_1x_0 + x_0^2}{x_2-x_0} = \frac{x_1x_2 - x_1x_0}{x_2-x_0} = x_1.$$

Следовательно,

$$f(x_1) = f(\lambda_1 x_2 + \lambda_2 x_0) \leqslant \lambda_1 f(x_2) + \lambda_2 f(x_0) =$$
$$= \frac{x_1-x_0}{x_2-x_0} f(x_2) + \left(1 - \frac{x_1-x_0}{x_2-x_0}\right) f(x_0),$$

или, что то же самое,

$$f(x_1) - f(x_0) \leqslant \frac{x_1-x_0}{x_2-x_0} (f(x_2) - f(x_0))$$

и

$$\frac{f(x_2)-f(x_0)}{x_2-x_0} \geqslant \frac{f(x_2)-f(x_0)}{x_1-x_0}.$$

Лемма доказана.

Неравенство (2.3) означает, что функция

$$\varphi(x)=\frac{f(x)-f(x_0)}{x-x_0}$$

монотонно убывает при $x \to +x_0$. Покажем, что $\varphi(x)$, кроме того, ограничена снизу для $x > x_0$.

Лемма 2.2. *Пусть $f(x)$ — выпуклая функция скалярного аргумента и $x^* < x_0 < x$. Тогда*

$$\frac{f(x)-f(x_0)}{x-x_0} \geqslant \frac{f(x_0)-f(x^*)}{x_0-x^*}. \tag{2.4}$$

Доказательство. Точки x^*, x_0, x связаны между собой теми же неравенствами, что и точки x_0, x_1, x_2 в лемме 2.1. Поэтому, заменив в утверждении этой леммы x_0 на x^*, x_1 на x_0 и x_2 на x, получим

$$\frac{f(x)-f(x^*)}{x-x^*} \geqslant \frac{f(x_0)-f(x^*)}{x_0-x^*}. \tag{2.5}$$

Но из неравенства

$$\frac{a_1}{b_1} \geqslant \frac{a_2}{b_2},$$

справедливого при некоторых a_1, a_2, $b_1 > b_2 > 0$, следует неравенство

$$\frac{a_1-a_2}{b_1-b_2} \geqslant \frac{a_2}{b_2},$$

что применительно к (2.5) дает

$$\frac{(f(x)-f(x^*))-(f(x_0)-f(x^*))}{(x-x^*)-(x_0-x^*)} \geqslant \frac{f(x_0)-f(x^*)}{x_0-x^*},$$

или

$$\frac{f(x)-f(x_0)}{x-x_0} \geqslant \frac{f(x_0)-f(x^*)}{x_0-x^*}.$$

Лемма доказана.

Итак, мы установили, что функция

$$\varphi(x)=\frac{f(x)-f(x_0)}{x-x_0}$$

скалярного аргумента x в случае, если $f(x)$ выпукла, монотонно убывает и ограничена снизу при $x \to +x_0$. От-

сюда следует существование предела

$$\lim_{x \to +x_0} \frac{f(x)-f(x_0)}{x-x_0},$$

т. е. существование производной справа у функции $f(x)$ в произвольной точке x_0. Совершенно аналогично устанавливается, что выпуклая функция $f(x)$ в каждой точке имеет производную слева. Отметим, что существование односторонних производных еще не означает дифференцируемости функции $f(x)$. Действительно, нетрудно построить пример выпуклой функции, график которой имеет изломы (рис. 2.3).

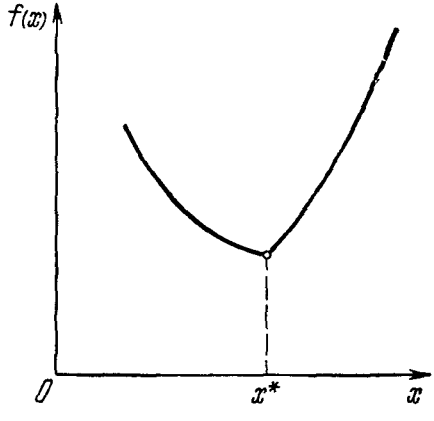


Рис 2.3.

Обобщением полученного результата на многомерный случай является следующая

Теорема 2.1. *Функция $f(x)$, заданная и выпуклая в пространстве E_n, имеет в каждой точке из E_n производную по любому направлению, т. е. при любых $x \in E_n$, $e \in E_n$, $\|e\| = 1$ существует предел*

$$\frac{\partial f}{\partial e}(x) = \lim_{t \to +0} \frac{f(x+te)-f(x)}{t}.$$

Доказательство. Возьмем произвольные $x \in E_n$, $e \in E_n$, $\|e\| = 1$ и рассмотрим функцию одной переменной $\psi(t) = f(x+te)$. Она, как легко убедиться, выпукла и поэтому в силу доказанного ранее имеет производные справа при любых t. В частности, у нее есть производная справа при $t = 0$:

$$\left(\frac{d\psi}{dt}\right)^+_{t=0} = \lim_{t \to +0} \frac{\psi(t)-\psi(0)}{t} = \lim_{t \to +0} \frac{f(x+te)-f(x)}{t}.$$

Таким образом, предел

$$\lim_{t \to +0} \frac{f(x+te)-f(x)}{t}$$

существует, что и требовалось доказать

В одномерном случае наличие у функции производных справа и слева гарантирует ее непрерывность. Таким образом, мы можем утверждать, что функция скалярного аргумента, определенная и выпуклая на всей вещественной оси, непрерывна в каждой ее точке. Свойством непрерывности обладают и выпуклые функции многих переменных, но это уже не следует из их дифференцируемости по любым направлениям — легко построить функцию, дифференцируемую по всем направлениям в некоторой точке и при этом разрывную в ней.

Пример 2.1. У функции двух переменных, которая в полярных координатах r, φ имеет вид

$$f(r, \varphi) = \frac{r}{1 - \sin\frac{\varphi}{4}}, \quad r \geqslant 0, \quad 0 \leqslant \varphi < 2\pi,$$

в нуле существует производная по любому направлению, но нуль — точка разрыва функции $f(r, \varphi)$.

Однако среди выпуклых функций такого примера не найти, поскольку справедлива

Теорема 2.2. *Функция $f(x)$, заданная и выпуклая в пространстве E_n, непрерывна в каждой его точке.*

Доказательство. Допустим противное. Тогда существуют число $\varepsilon > 0$, точка $x_0 \in E_n$ и сходящаяся к ней последовательность $\{x_i\}$, $i = 1, 2, \ldots$, такие, что

$$|f(x_i) - f(x_0)| \geqslant \varepsilon, \qquad i = 1, 2, \ldots$$

Построим при каждом i функцию

$$\psi_i(t) = f(x_0 + t(x_0 - x_i)).$$

Она выпукла и поэтому в силу леммы 2.2 при любых $t^* < t_0 < t$ выполнено неравенство

$$\frac{\psi_i(t) - \psi_i(t_0)}{t - t_0} \geqslant \frac{\psi_i(t_0) - \psi_i(t^*)}{t_0 - t^*}. \tag{2.6}$$

Полагая здесь

$$t^* = -1, \quad t_0 = 0, \quad t = \frac{1}{\sqrt{\|x_0 - x_i\|}},$$

получим

$$(f(y_i) - f(x_0))\sqrt{\|x_0 - x_i\|} \geqslant f(x_0) - f(x_i), \tag{2.7}$$

где

$$y_i = x_0 + (x_0 - x_i)/\sqrt{\|x_0 - x_i\|},$$

а при

$$t^* = -1 - 1/\sqrt{\|x_0 - x_i\|}, \quad t_0 = -1, \quad t = 0$$

неравенство (2.6) принимает вид

$$f(x_0) - f(x_i) \geqslant (f(x_i) - f(z_i))\sqrt{\|x_0 - x_i\|}, \tag{2.8}$$

где

$$z_i = x_i - (x_0 - x_i)/\sqrt{\|x_0 - x_i\|}.$$

Таким образом, либо $f(x_0) \geqslant f(x_i) + \varepsilon$, и тогда из (2.7) следует, что

$$f(y_i) \geqslant f(x_0) + \frac{\varepsilon}{\sqrt{\|x_0 - x_i\|}}, \tag{2.9}$$

либо $f(x_i) \geqslant f(x_0) + \varepsilon$, и тогда из (2.8) видно, что

$$f(z_i) \geqslant f(x_i) + \frac{\varepsilon}{\sqrt{\|x_0 - x_i\|}} \geqslant f(x_0) + \frac{\varepsilon}{\sqrt{\|x_0 - x_i\|}}. \tag{2.10}$$

Рассмотрим теперь содержащее точку x_0 вместе с некоторой ее окрестностью множество K точек вида

$$x = \sum_{k=1}^{2n} \lambda_k v_k, \quad v_k \in E_n, \quad \lambda_k \geqslant 0, \quad \sum_{i=1}^{2n} \lambda_k = 1,$$

где $v_k = \{v_k^1, \ldots, v_k^n\}$, $v_k^s = x_0^s$ при $k \neq 2s$, $k \neq 2s-1$, $v_k^s = x_0^s + 1$ при $k = 2s$, $v_k^s = x_0^s - 1$ при $k = 2s-1$. Для произвольной точки x из этого множества имеем

$$f(x) = f\left(\sum_{k=1}^{2n} \lambda_k v_k\right) \leqslant \sum_{k=1}^{2n} \lambda_k f(v_k) \leqslant \max_{k=1, 2, \ldots, 2n} \{f(v_k)\}.$$

Точки y_i, z_i, как видно из их определения, сходятся к x_0 и поэтому при достаточно больших i принадлежат K. Но отсюда следует, что при достаточно больших i выполнены неравенства

$$f(y_i) \leqslant \max_{k=1, 2, \ldots, 2n} f(v_k), \quad f(z_i) \leqslant \max_{k=1, \ldots, 2n} f(v_k),$$

несовместимые с (2.9), (2.10). Полученное противоречие доказывает теорему.

До сих пор речь шла о свойствах только таких выпуклых функций, областью определения которых является пространство E_n. Однако, используя те же доказательства, что и прежде, можно установить существование в точке x_0 производных по всем направлениям и непрерывность для

функции, выпуклой на произвольном выпуклом множестве, содержащем x_0 как внутреннюю точку (на границе области своего определения выпуклая функция может иметь разрывы).

Наконец, следует отметить, что выпуклость функции позволяет сформулировать для нее простой критерий существования безусловного минимума.

Теорема 2.3. *Для того чтобы функция $f(x)$, определенная и выпуклая на пространстве E_n, достигала в x_0 безусловного минимума, необходимо и достаточно, чтобы производные от $f(x)$ по всем направлениям, вычисленные в точке x_0, были неотрицательны.*

Доказательство. Из отрицательности производной от $f(x)$ по некоторому направлению следовала бы возможность, сделав из x_0 малый шаг вдоль этого направления, уменьшить значение $f(x)$. Следовательно, в точке безусловного минимума производные по всем направлениям должны быть не меньше нуля. Необходимость доказана.

Пусть теперь при всех $e \in E_n$, $\|e\| = 1$ выполнено неравенство

$$\frac{\partial f}{\partial e}(x_0) = \lim_{t \to +0} \frac{f(x_0 + te) - f(x_0)}{t} \geqslant 0.$$

В силу леммы 2.2 отсюда следует, что

$$\frac{f(x_0 + te) - f(x_0)}{t} \geqslant 0$$

и, соответственно,

$$f(x_0 + te) - f(x_0) \geqslant 0$$

при любых $t > 0$, $e \in E_n$, $\|e\| = 1$. В частности, полагая

$$e = \frac{x - x_0}{\|x - x_0\|}, \quad t = \|x - x_0\|,$$

где x — некоторая не равная x_0 точка из E_n, отсюда получим

$$f(x) - f(x_0) \geqslant 0.$$

Таким образом, x_0 — точка глобального безусловного минимума $f(x)$. Теорема доказана.

Если выпуклая функция $f(x)$ дифференцируема, неотрицательность в некоторой точке x_0 всех ее производных по направлениям означает, что градиент $f(x)$ в точке x_0 равен нулю. Поэтому для дифференцируемых выпуклых

функций критерий существования минимума звучит так: безусловный минимум дифференцируемой выпуклой функции $f(x)$ достигается в некоторой точке x_0 в том и только в том случае, если $f'(x_0)=0$. Для вогнутых функций равенство нулю градиента будет необходимым и достаточным условием безусловного максимума.

2. Опорные функционалы. В нелинейном программировании большую роль играет понятие опорного функционала.

Определение 2.2. Вектор $c \in E_n$ является *опорным функционалом* для некоторой функции $f(x)$ в точке $x_0 \in E_n$, если при всех $x \in E_n$ справедливо неравенство

$$f(x)-f(x_0) \geqslant (c,\ x-x_0). \tag{2.11}$$

Множество опорных функционалов в точке x_0 будем обозначать $M(x_0)$.

Рассмотрим два примера, поясняющих смысл данного определения.

Пример 2.2. Пусть $y=f(x)$ — скалярная функция скалярного аргумента. Предположим, что она выпукла и в точке $x=x_0$ дифференцируема. Тогда для всех x справедливо неравенство

$$f(x)-f(x_0) \geqslant f'(x_0)(x-x_0),$$

причем для любого $c \neq f'(x_0)$ можно подобрать x так, чтобы было

$$f(x)-f(x_0) < c(x-x_0).$$

Следовательно, в данном случае множество $M(x_0)$ состоит из единственного опорного функционала $c=f'(x_0)$. Если же x_0 — точка, в которой правая и левая производные от $f(x)$ не равны между собой, неравенство (2.11) будет выполнено как при c, равном одной из величин $f'_-(x_0)$ и $f'_+(x_0)$:

$$f(x)-f(x_0) \geqslant f'_+(x_0)(x-x_0),$$
$$f(x)-f(x_0) \geqslant f'_-(x_0)(x-x_0),$$

так и для любого c, удовлетворяющего неравенствам

$$f'_-(x_0) \leqslant c \leqslant f'_+(x_0).$$

Таким образом, в этом случае множество $M(x_0)$ содержит континуум опорных функционалов.

Пример 2.3. Пусть теперь $x \in E_2$, т. е. является двумерным вектором с компонентами x^1 и x^2 и $f(x) = f(x^1, x^2)$ — выпуклая дифференцируемая функция. Тогда при любом $x \neq x_0$ имеем

$$\frac{f(x)-f(x_0)}{\|x-x_0\|} = \frac{f(x_0+\|x-x_0\|e)-f(x_0)}{\|x-x_0\|} \geqslant \frac{\partial f}{\partial e}(x_0),$$

где e — вектор единичной длины, совпадающий по направлению с $x-x_0$. Это неравенство можно переписать в виде

$$f(x)-f(x_0) \geqslant \frac{\partial f}{\partial e}(x_0)\|x-x_0\|,$$

а поскольку

$$\frac{\partial f}{\partial e}(x_0) = (f'(x_0), e), \quad x-x_0 = \|x-x_0\|e,$$

окончательно получим

$$f(x)-f(x_0) \geqslant (f'(x_0), x-x_0).$$

Следовательно, множество $M(x_0)$ в этом случае включает вектор $c = f'(x_0)$ и никаких других c, как легко проверить, в нем нет.

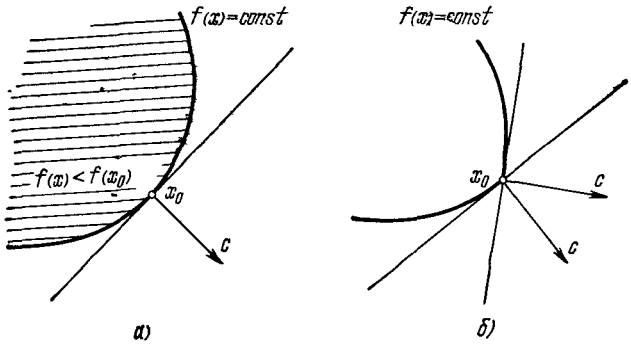


Рис. 2.4.

Этот вывод, очевидно, не зависит от размерности вектора x. Значит, если выпуклая функция $f(x)$ дифференцируема, множество $M(x_0)$ непусто: оно содержит (и притом единственный) элемент $c = f'(x_0)$. Если же в точке x_0 выпуклая функция n переменных не дифференцируема, то подобно тому, как это было в случае одной переменной

(пример 2.1), множество $M(x_0)$ содержит континуум векторов c.

С каждым вектором $c \in M(x_0)$ можно связать некоторую плоскость, ортогональную c и проходящую через точку x_0 (см. рис. 2.4). Если функция $f(x)$ дифференцируема в точке x_0, эта плоскость единственна (рис. 2.4, *а*). В противном случае мы получим целый пучок плоскостей (рис. 2.4, *б*). Их принято называть опорными. Множество точек x, где $f(x) < f(x_0)$, лежит по одну сторону от каждой из этих плоскостей.

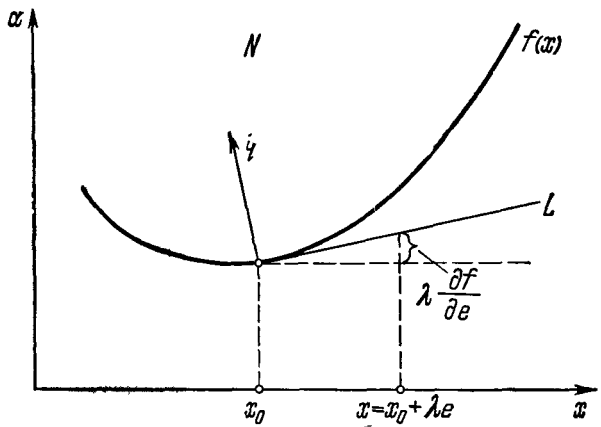


Рис. 2.5.

3. Опорные функционалы и производные по направлению. Мы видели, что, когда выпуклая функция дифференцируема в точке x_0, множество опорных функционалов $M(x_0)$ состоит из единственного вектора — ее градиента. В противном случае связь между производными от $f(x)$ и опорными функционалами более сложна. Ее устанавливает следующая

Теорема 2.4. *Если $f(x)$ — выпуклая функция, определенная на E_n, x_0 — некоторая точка из E_n и $e \in E_n$ — произвольный вектор единичной длины, то множество $M(x_0)$ непусто и выполняется равенство*

$$\frac{\partial f}{\partial e}(x_0) = \max(c, e), \quad c \in M(x_0).$$

Заметим, что это равенство в случае дифференцируемости функции $f(x)$ переходит в определение производной по направлению:

$$\frac{\partial f}{\partial e}(x_0) = (f'(x_0),\ e).$$

Доказательство теоремы следует из нескольких вспомогательных утверждений.

Введем $(n+1)$-мерное пространство векторов z, имеющих компоненты (рис. 2.5)

$$z^i = x^i, \quad i = 1, \ \ldots, \ n,$$
$$z^{n+1} = \alpha.$$

В этом пространстве рассмотрим два множества, N и L. В множество N включим все векторы $z = \{x, \ \alpha\}^T$, у которых компоненты $x^1, \ \ldots, \ x^n$ и α связаны соотношением

$$\alpha > f(x).$$

(На рис. 2.5 множество N — совокупность точек плоскости, лежащих выше кривой $\alpha = f(x)$.) Множество L — это луч, проведенный из точки $\{x_0, \ f(x_0)\}^T$ в направлении $\left\{e, \ \frac{\partial f}{\partial e}(x_0)\right\}^T$, где e — некоторый вектор единичной длины. Координаты произвольной точки $z = \{x, \ \alpha\}^T \in L$ определяются по формулам

$$x = x_0 + \lambda e,$$
$$\alpha = f(x_0) + \lambda \frac{\partial f}{\partial e}(x_0), \quad \lambda \geqslant 0.$$

Справедлива

Лемма 2.3. *Множество N выпукло и не имеет общих точек с множеством L.*

Доказательство. Установим сначала выпуклость множества N. Возьмем точку

$$z = \lambda_1 z_1 + \lambda_2 z_2 = \{\lambda_1 x_1 + \lambda_2 x_2, \ \lambda_1 \alpha_1 + \lambda_2 \alpha_2\}^T,$$

где

$$z_1 = \{x_1, \ \alpha_1\}^T \in N, \quad z_2 = \{x_2, \ \alpha_2\}^T \in N,$$
$$\lambda_1 + \lambda_2 = 1, \quad \lambda_1 \geqslant 0, \quad \lambda_2 \geqslant 0.$$

Так как $f(x)$ — выпуклая функция, то

$$f(\lambda_1 x_1 + \lambda_2 x_2) \leqslant \lambda_1 f(x_1) + \lambda_2 f(x_2),$$

а поскольку $f(x_i) < \alpha_i$, $i = 1, \ 2$, отсюда следует, что

$$f(\lambda_1 x_1 + \lambda_2 x_2) < \lambda_1 \alpha_1 + \lambda_2 \alpha_2.$$

Таким образом, точка z принадлежит N, т. е. N — выпуклое множество.

Предположим теперь, что на луче L есть точка, определяемая значением $\tilde{\lambda}$, которая принадлежит множеству N. Это значит, что

$$f(x_0)+\tilde{\lambda}\frac{\partial f}{\partial e}(x_0) > f(x_0+\tilde{\lambda}e).$$

Видно, что $\tilde{\lambda}$ не может быть равным нулю, и поэтому данное неравенство эквивалентно следующему:

$$\frac{f(x_0+\tilde{\lambda}e)-f(x_0)}{\tilde{\lambda}} < \frac{\partial f}{\partial e}(x_0). \tag{2.12}$$

Но ранее было доказано, что если $f(x)$ — выпуклая функция, то

$$\frac{\partial f}{\partial e}(x_0) = \lim_{\tilde{\lambda}\to+0}\frac{f(x_0+\tilde{\lambda}e)-f(x_0)}{\tilde{\lambda}},$$

причем отношение

$$\frac{f(x_0+\tilde{\lambda}e)-f(x_0)}{\tilde{\lambda}}$$

стремится к своему пределу, монотонно убывая. Неравенство (2.12) противоречит этому утверждению, что и доказывает нашу лемму.

Итак, установлено, что множества N и L не пересекаются. Поэтому, согласно теореме об отделимости, найдется вектор $q=\{a,\ b\}^T \in E_{n+1}$, $q\neq 0$, такой, что

$$(z_N,\ q) \geqslant (z_L,\ q) \tag{2.13}$$

для любых $z_N \in N$, $z_L \in L$. В силу определения множеств N и L это неравенство означает, что

$$b\alpha+(a,\ x) \geqslant b\left(f(x_0)+\lambda\frac{\partial f}{\partial e}\right)+(a,\ x_0+\lambda e) \tag{2.14}$$

при всех $\lambda \geqslant 0$ и $\alpha > f(x)$, причем сразу видно, что $b\neq 0$. Действительно, при $b=0$, во-первых, $a\neq 0$ и, во-вторых, из (2.14) следовало бы, что

$$(a,\ x) \geqslant (a,\ x_0+\lambda e)$$

для всех x, а это невозможно.

На основании неравенства (2.14) доказывается следующая

Лемма 2.4. *Вектор* $c_0 = -a/b$ *является опорным функционалом для* $f(x)$ *в точке* x_0, *причем*

$$(c_0,\ e) \geqslant \frac{\partial f}{\partial e}(x_0).$$

Доказательство. Покажем, прежде всего, что $b > 0$. В самом деле, полагая в (2.14) $x = x_0$, $\lambda = 0$, получим

$$b(\alpha - f(x_0)) \geqslant 0,$$

и это неравенство должно выполняться для всех $\alpha > f(x_0)$, что возможно только при $b \geqslant 0$. Но $b \neq 0$, поэтому $b > 0$. Далее, возьмем произвольную точку $x \in E_n$. Неравенство (2.14) выполнено при любых сколь угодно близких к $f(x)$ значениях $\alpha > f(x)$, а значит, и для $\alpha = f(x)$. При этом, если взять $\lambda = 0$, оно принимает вид

$$b(f(x) - f(x_0)) \geqslant -(a,\ x - x_0)$$

или, что то же самое,

$$f(x) - f(x_0) \geqslant \left(-\frac{a}{b},\ x - x_0\right).$$

Таким образом, вектор $c_0 = -\frac{a}{b}$ является опорным функционалом для $f(x)$ в точке x_0. Наконец, подстановка в (2.14) $\alpha = f(x)$ и $\lambda = 1$ дает неравенство

$$b(f(x) - f(x_0)) \geqslant -(a,\ x - x_0) + b\frac{\partial f}{\partial e} + (a,\ e),$$

откуда при $x = x_0$ имеем

$$b\left(\frac{\partial f}{\partial e}(x_0) - (c_0,\ e)\right) \leqslant 0,$$

т. е.

$$\frac{\partial f}{\partial e}(x_0) \leqslant (c_0,\ e). \tag{2.15}$$

Лемма доказана.

Непустоту множества опорных функционалов для выпуклой функции $f(x)$ в произвольной точке x_0 мы установили. В этом состояло первое утверждение теоремы 2.4. Чтобы доказать второе, нужна

Лемма 2.5. *Для любого* $e \in E_n$, $\|e\| = 1$, *справедливо неравенство*

$$\frac{\partial f}{\partial e}(x_0) \geqslant \max_{c \in M(x_0)} (c,\ e).$$

Доказательство. По определению опорного функционала, для любого $c \in M(x_0)$ и любого $x \in E_n$ имеет место неравенство

$$f(x) - f(x_0) \geqslant (c,\ x - x_0).$$

Положим

$$x = x_0 + te, \quad t \geqslant 0,$$

где e — произвольный вектор единичной длины. Тогда

$$\frac{f(x_0 + te) - f(x_0)}{t} \geqslant (c,\ e),$$

откуда, переходя к пределу при $t \to +0$, получим

$$\frac{\partial f}{\partial e}(x_0) \geqslant (c,\ e)$$

Поскольку это неравенство получено для произвольного $c \in M(x_0)$, должно быть

$$\frac{\partial f}{\partial e}(x_0) \geqslant \max_{c \in M(x_0)} (c,\ e). \tag{2.16}$$

Лемма доказана.

Сопоставляя неравенства (2.15), (2.16), легко видеть, что

$$\frac{\partial f}{\partial e}(x_0) = (c_0,\ e) = \max_{c \in M(x_0)} (c,\ e).$$

Тем самым теорема 2.4 доказана полностью.

На этом мы закончим изучение свойств выпуклых функций и перейдем к анализу условий экстремума при наличии ограничений.

§ 3. Условия экстремума в задачах нелинейного программирования

1. Основное необходимое условие оптимальности. В этом пункте будет получено необходимое условие минимума функции $f(x)$ n-мерного векторного аргумента x на пересечении некоторых множеств G_i, $i = 1, 2, \ldots, m+1$, т. е. условие, которое должно выполняться в решении задачи

$$\min f(x), \quad x \in \bigcap_{i=1}^{m+1} G_i. \tag{3.1}$$

Никаких требований к функции $f(x)$ и множествам G_i здесь предъявляться не будет. Поэтому условие, о котором идет речь, совершенно неконструктивно и само по себе практической ценности не имеет. Однако на его основе удается строить практичные необходимые условия минимума для достаточно широкого класса задач с ограничениями вида нелинейных равенств и неравенств. Каждую из таких задач можно записать в форме (3.1), подразумевая, что G_i при $i = 1, \ldots, m$ есть множество точек, удовлетворяющих ее i-му ограничению — неравенству, а G_{m+1} — множество точек, подчиняющихся всем ограничениям — равенствам (если среди ограничений неравенств нет, то в записи (3.1) будет $m = 1$, $G_1 = E_n$; если все ограничения заданы неравенствами, надо взять $G_{m+1} = E_n$). Неравноправие равенств и неравенств в том отношении, что каждому неравенству в записи (3.1) сопоставляется свое множество G_i, а равенствам — одно множество G_{m+1} на всех, не случайно: иначе общее условие оптимальности для задачи типа (3.1) не удается расшифровать в эквивалентные ему более конструктивные необходимые условия минимума при ограничениях.

В дальнейшем нам понадобятся следующие определения.

Определение 3.1. Вектор e называется *направлением убывания* функции $f(x)$ в точке x_0, если существуют числа $\delta > 0$, $\varepsilon_0 > 0$ такие, что при всех e', ε, удовлетворяющих условиям

$$\|e' - e\| < \delta, \quad 0 < \varepsilon < \varepsilon_0,$$

выполнено неравенство

$$f(x_0 + \varepsilon e') < f(x_0).$$

Множество направлений убывания $f(x)$ в точке x_0 либо пусто, либо представляет собой открытый конус, который в дальнейшем будем обозначать через $\Gamma^0_{x_0}$. Если поверхность с уравнением $f(x) = f(x_0)$ не вырождается в точку и является гладкой в x_0, конус $\Gamma^0_{x_0}$ есть открытое полупространство (рис. 3.1) Когда x_0 — точка строгого безусловного максимума функции $f(x)$, конус $\Gamma^0_{x_0}$ совпадает с пространством E_n.

Определение 3.2. Говорят, что вектор e будет для множества G_i, $i \leqslant m$, *возможным направлением* в точке $x_0 \in G_i$, если найдутся $\delta > 0$, $\varepsilon_0 > 0$ такие, что при всех

e', ε, удовлетворяющих условиям

$$\|e' - e\| < \delta, \quad 0 < \varepsilon < \varepsilon_0,$$

точка

$$x_0 + \varepsilon e'$$

принадлежит множеству G_t.

Нетрудно понять, что возможные направления для множества G_t в точке $x_0 \in G_t$ могут существовать (но не обязательно существуют) только в том случае, когда

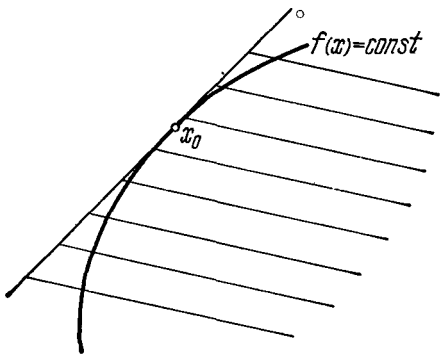


Рис. 3.1.

в этом множестве есть внутренние точки. Коль скоро возможные направления найдутся, они составят открытый конус. Последний обозначим через $\Gamma^t_{x_0}$. В частности, когда точка x_0 является внутренней для G_t, конус $\Gamma^t_{x_0}$ есть пространство E_n.

Определение 3.3. Вектор e называется *касательным направлением* к множеству G_{m+1} в точке $x_0 \in G_{m+1}$, если существует функция $o(\varepsilon)\left(\lim\limits_{\varepsilon \to +0} \frac{o(\varepsilon)}{\varepsilon} = 0\right)$ такая, что точка

$$x(\varepsilon) = x_0 + \varepsilon e + o(\varepsilon)$$

принадлежит G_{m+1} при всех $\varepsilon > 0$.

Касательные направления к множеству G_{m+1} в точке $x_0 \in G_{m+1}$, если они есть, образуют конус, который впредь обозначается через $\Gamma^{m+1}_{x_0}$. Обычно $\Gamma^{m+1}_{x_0}$ — некоторое подпространство в E_n. Так будет, например, если G_{m+1} представляет собой множество точек, удовлетворяющих

системе ограничений — равенств, функции которых дифференцируемы и имеют в точке x_0 линейно независимые градиенты.

Справедлива следующая

Лемма 3.1 *Для того чтобы точка x_0 была решением задачи* (3.1), *необходимо выполнение равенства*

$$\bigcap_{i=0}^{m+1} \Gamma^i_{x_0} = \emptyset,$$

где символ $\emptyset$ означает пустое множество.

Доказательство. Допустим, что лемма неверна, т. е. найдется вектор $e \in \Gamma^i_{x_0}$, $i = 0, 1, \ldots, m+1$. Тогда, по определению конуса убывания $\Gamma^0_{x_0}$ и конусов возможных направлений $\Gamma^i_{x_0}$, существуют числа $\delta_i > 0$, $\varepsilon_i > 0$, $i = 0, \ldots, m$, такие, что для каждого $i = 1, \ldots, m$ при любых e', ε, удовлетворяющих условиям $\|e' - e\| < \delta_i$, $0 < \varepsilon < \varepsilon_i$, точка $x_0 + \varepsilon e'$ принадлежит G_i, а при $\|e' - e\| < \delta_0$, $0 < \varepsilon < \varepsilon_0$, выполнено неравенство

$$f(x_0 + \varepsilon e') < f(x_0).$$

Следовательно, если положить

$$\underline{\delta} = \min_{i=0,\ldots,m} \{\delta_i\}, \quad \underline{\varepsilon} = \min_{i=0,\ldots,m} \{\varepsilon_i\},$$

точка $x_0 + \varepsilon e'$ будет принадлежать множеству $\bigcap_{i=1}^{m} G_i$ и удовлетворит неравенству

$$f(x_0 + \varepsilon e') < f(x_0)$$

при любых $\|e' - e\| < \underline{\delta}$; $0 < \varepsilon < \underline{\varepsilon}$.

Возьмем теперь функцию $o(\varepsilon)$, фигурирующую в определении e как элемента конуса касательных направлений $\Gamma^{m+1}_{x_0}$, и подберем для нее $\varepsilon^* > 0$ так, чтобы при всех $0 < \varepsilon < \varepsilon^*$ для $e^* = e + \frac{o(\varepsilon)}{\varepsilon}$ было

$$\|e^* - e\| = \left\| \frac{o(\varepsilon)}{\varepsilon} \right\| < \underline{\delta}.$$

Тогда при $0 < \varepsilon < \min\{\underline{\varepsilon}, \varepsilon^*\}$ точка

$$x(\varepsilon) = x_0 + \varepsilon e^* = x_0 + \varepsilon e + o(\varepsilon),$$

во-первых, принадлежит пересечению $\bigcap_{i=1}^{m+1} G_i$, т. е. допустима для задачи (3.1), а во-вторых, для нее справедливо неравенство

$$f(x(\varepsilon)) < f(x_0).$$

Но это невозможно, так как x_0 — решение задачи (3.1). Полученное противоречие доказывает лемму.

Таким образом, пересечение конусов убывания, возможных и касательных направлений, построенных в решении задачи (3.1), должно быть пустым. Это и есть *основное необходимое условие экстремума при наличии* ограничений. Оно получено без каких-либо предположений относительно свойств задачи (3.1). Однако уже на первом этапе расшифровки этого условия (при выводе эквивалентных ему более конструктивных условий) предположение такого сорта необходимо и состоит в следующем: задача (3.1) должна быть такой, чтобы конусы $\Gamma^i_{x_0}$, $i=0, \ldots, m+1$, были непустыми и выпуклыми. Условия, к которым приводит данное предположение, рассмотрены в следующем пункте.

2. Уравнения Эйлера — Лагранжа. В случае, когда x_0 — решение задачи (3.1), в котором конусы $\Gamma^i_{x_0}$, $i=0, 1, \ldots, m+1$, непусты и выпуклы, условия, эквивалентные требованию пустоты пересечения этих конусов, устанавливает

Теорема Милютина — Дубовицкого. *Для того чтобы пересечение непустых выпуклых конусов* $\Gamma^i_{x_0}$, $i=0, \ldots, m+1$, *было пустым, необходимо и достаточно, чтобы нашлись векторы* c_i *из сопряженных конусов, т. е.*

$$c_i \in (\Gamma^i_{x_0})^*, \quad i=0, 1, 2, \ldots, m+1,$$

не все равные нулю и удовлетворяющие равенству

$$\sum_{i=0}^{m+1} c_i = 0. \tag{3.2}$$

Доказательство. Установим сначала необходимость. Пусть пересечение выпуклых конусов $\Gamma^i_{x_0} \neq \emptyset$, $i=0, 1, \ldots, m+1$, пусто. Тогда, рассматривая после-

довательно пересечения $\bigcap_{i=0}^{k-1}\Gamma_{x_0}^i$ для $k=m+1,\ m,\ \ldots$, мы рано или поздно придем к некоторому $k_0\leqslant m+1$ такому, что

$$\bigcap_{i=0}^{k_0-1}\Gamma_{x_0}^i\neq\emptyset,\quad \bigcap_{i=0}^{k_0}\Gamma_{x_0}^i=\emptyset.$$

Два непересекающихся непустых выпуклых конуса $\bigcap_{i=0}^{k_0-1}\Gamma_{x_0}^i$, $\Gamma_{x_0}^{k_0}$ можно отделить друг от друга, т. е. найти вектор $c\neq 0$, для которого

$$(c,\ e_1)\geqslant(c,\ e_2)$$

при любых $e_1\in\Gamma_{x_0}^{k_0}$, $e_2\in\bigcap_{i=0}^{k_0-1}\Gamma_{x_0}^i$. Поскольку наряду с e_1, e_2 конусы $\Gamma_{x_0}^{k_0}$, $\bigcap_{i=0}^{k_0-1}\Gamma_{x_0}^i$ включают векторы $\lambda_1 e_1$, $\lambda_2 e_2$, где $\lambda_1>0$, $\lambda_2>0$ произвольны, это неравенство возможно лишь в том случае, когда

$$(c,\ e_1)\geqslant 0,\quad (c,\ e_2)\leqslant 0$$

для всех $e_1\in\Gamma_{x_0}^{k_0}$, $e_2\in\bigcap_{i=0}^{k_0-1}\Gamma_{x_0}^i$. Последнее означает, что вектор c принадлежит конусу $(\Gamma_{x_0}^{k_0})^*$, а вектор $(-c)$ — конусу $\left(\bigcap_{i=0}^{k_0-1}\Gamma_{x_0}^i\right)^*$. Но конусы $\Gamma_{x_0}^i$, $i=0,\ 1,\ \ldots,\ k_0-1$, открыты, а потому их пересечение телесно и, соответственно, в силу теоремы 1.5 § 1 настоящей главы выполнено равенство

$$\left(\bigcap_{i=0}^{k_0-1}\Gamma_{x_0}^i\right)^*=\sum_{i=0}^{k_0-1}(\Gamma_{x_0}^i)^*,$$

т. е. вектор $(-c)$ представим в виде

$$-c=\sum_{i=0}^{k_0-1}c_i,\quad c_i\in(\Gamma_{x_0}^i)^*,\quad i=0,\ 1,\ \ldots,\ k_0-1.$$

Отсюда, полагая $c_{k_0}=c\in(\Gamma_{x_0}^{k_0})^*$ и $c_i=0$, $c_i\in(\Gamma_{x_0}^i)^*$ для

всех $i \geqslant k_0+1$, получим для c_i, $i=0, 1, \ldots, m+1$, равенство (3.2). Необходимость доказана.

Докажем достаточность. Пусть нашлись не все равные нулю векторы $c_i \in (\Gamma^i_{x_0})^*$, $i=0, \ldots, m+1$, такие, что

$$\sum_{i=0}^{m+1} c_i = 0,$$

и допустим, что при этом существует ненулевой вектор $e \in \bigcap_{i=0}^{m+1} \Gamma^i_{x_0}$. Тогда в силу определения c_i выполнены неравенства

$$(c_i, e) \geqslant 0, \quad i=0, 1, \ldots, m+1,$$

причем

$$\sum_{i=0}^{m+1} (c_i, e) = \left(\sum_{i=0}^{m+1} c_i, e\right) = 0.$$

Отсюда следует, что

$$(c_i, e) = 0, \quad i=0, \ldots, m+1.$$

Но это невозможно, так как среди векторов c_i, $i=0, \ldots, m$, есть ненулевой, $c_s \neq 0$, $s \leqslant m$, а поскольку конус $\Gamma^s_{x_0}$ открыт и потому наряду с вектором e включает при достаточно малом $\varepsilon > 0$ вектор $e - \varepsilon c_s$, из равенства

$$(c_s, e) = 0$$

вытекает противоречащее определению c_s неравенство

$$(c_s, e - \varepsilon c_s) = -\varepsilon (c_s, c_s) < 0,$$

$$e - \varepsilon c_s \in \Gamma^s_{x_0}.$$

Полученное противоречие означает, что $\bigcap_{i=0}^{m+1} \Gamma^i_{x_0} = \emptyset$. Теорема доказана.

Итак, для того чтобы точка x_0, в которой конусы $\Gamma^i_{x_0}$, $i=0, 1, \ldots, m+1$, непусты и выпуклы, была решением задачи (3.1), необходимо существование не равных нулю одновременно векторов $c_i \in (\Gamma^i_{x_0})^*$, $i=0, 1, \ldots, m+1$, таких, что

$$c_0 + c_1 + \ldots + c_{m+1} = 0. \tag{3.3}$$

Это, весьма общее условие экстремума при наличии ограничений в конкретных случаях расшифровывается с исполь-

зованием множителей Лагранжа Поэтому векторное равенство (3.3) называют уравнением Эйлера — Лагранжа.

В заключение данного пункта выделим задачи нелинейного программирования вида

$$\begin{gathered} \min f(x), \\ \varphi_i(x) \leqslant 0, \quad i = 1, 2, \ldots, s, \\ \varphi_i(x) = 0, \quad i = s+1, \ldots, k, \\ x \in G \subset E_n, \end{gathered} \tag{3.4}$$

в которых конусы убывания, возможных и касательных направлений, построенных в решении, будут непустыми и выпуклыми и для которых, соответственно, применимы необходимые условия оптимальности в форме уравнений Эйлера — Лагранжа. Запись (3.4) преобразуется в (3.1), если ввести следующие обозначения: $m = s+1$; $G_i = \{x: \varphi_i(x) \leqslant 0\}$, $i = 1, 2, \ldots, m-1$; $G_m = G$; $G_{m+1} = \{x: \varphi_i(x) = 0$ для $i = s+1, \ldots, k\}$. Таким образом, нам нужно понять, каким требованием должны подчиняться множество G и функции $f(x)$, $\varphi_i(x)$, $i = 1, \ldots, k$, чтобы в решении x_0 непустыми и выпуклыми были: а) конусы возможных направлений для множеств G, $\{x: \varphi_i(x) \leqslant 0\}$, $i = 1, \ldots, s$; б) конус касательных направлений для множества $\{x: \varphi_i(x) = 0, \ i = s+1, \ldots, k\}$; в) конус направлений убывания функции $f(x)$.

Пусть $\varphi_i(x)$, $i = 1, \ldots, s$, — непрерывно дифференцируемые функции и в решении x_0 задачи (3.4) либо $\varphi_i(x_0) < 0$, либо $\varphi_i(x_0) = 0$, $\varphi_i'(x_0) \neq 0$. Тогда конусы возможных направлений для множеств $\{x: \varphi_i(x) \leqslant 0\}$, $i = 1, \ldots, s$, в точке x_0 непусты и выпуклы. Действительно, если $\varphi_i(x_0) < 0$, конус $\Gamma^t_{x_0}$, как легко понять, совпадает с E_n. Если же $\varphi_i(x_0) = 0$, $\varphi_i'(x_0) \neq 0$, то нетрудно показать, что $\Gamma^t_{x_0} = \{e: (\varphi_i'(x_0), e) < 0\}$, т. е. $\Gamma^t_{x_0}$ есть открытое полупространство — множество выпуклое. Конус возможных направлений $\Gamma^t_{x_0}$ для множества $\{x: \varphi_i(x) \leqslant 0\}$ будет непустым и выпуклым также в том случае, когда $\varphi_i(x)$ — выпуклая функция и существуют точки, в которых ее значения отрицательны Тогда $\Gamma^t_{x_0} = \{e: e = \nu(x - x_0), \nu > 0, \varphi_i(x) < 0\}$ Для доказательства этого утверждения не требуется ничего, кроме соответствующих определений, и читатель без труда проделает его самостоятельно.

Достаточным условием непустоты и выпуклости конуса Γ_{x_0} возможных направлений для множества G в точке x_0 является выпуклость этого множества и существование у него внутренних точек. При этом $\Gamma_{x_0}=\{e: e=\nu(x-x_0),\ \nu>0,\ x\in G^0\}$, где G^0 — внутренность множества G.

Непустоту и выпуклость конуса направлений убывания функции $f(x)$ в точке x_0 можно гарантировать, если эта функция выпукла и не имеет в x_0 глобального безусловного минимума, либо непрерывно дифференцируема и $f'(x_0)\neq 0$. В первом случае $\Gamma^1_{x_0}=\{e: e=\nu(x-x_0),\ \nu>0, f(x)<f(x_0)\}$, а во втором $\Gamma^1_{x_0}=\{e: (f'(x_0), e)<0\}$.

Наконец, конус касательных к множеству $\{x\cdot \varphi_i(x)=0$ для $i=s+1, \ldots, k\}$ направлений будет в x_0 выпуклым, если градиенты $\varphi_i'(x_0)$ линейно независимы. Тогда

$$\Gamma^{m+1}_{x_0}=\{e: (\varphi_i'(x_0), e)=0 \text{ при } i=s+1, \ldots, k\}.$$

3. Обобщенное правило множителей Лагранжа. В предыдущем пункте мы проделали первый этап расшифровки основного необходимого условия экстремума при наличии ограничений, а именно — требования, чтобы пересечение конусов возможных направлений, касательных направлений и конуса направлений убывания было пустым. В предположении непустоты и выпуклости всех перечисленных конусов удалось доказать эквивалентность данного условия утверждения о существовании нетривиального решения уравнений Эйлера — Лагранжа, т. е. о существовании не равных нулю одновременно и в сумме дающих нуль элементов сопряженных конусов. Здесь мы проведем второй этап расшифровки: выпишем сопряженные конусы и на основе уравнений Эйлера — Лагранжа получим конкретные необходимые условия минимума для задачи нелинейного программирования вида

$$\begin{gathered}\min f(x),\\ \varphi_i(x)\leqslant 0,\quad i=1, \ldots, m,\\ \varphi_i(x)=0,\quad i=m+1, \ldots, k,\\ x\in G\subset E_n,\end{gathered}\tag{3.5}$$

где f, $\varphi_i(x)$ — непрерывно дифференцируемые функции, а G — выпуклое замкнутое множество, содержащее внут-

ренние точки (в приложениях, чаще всего, $G=\{x\colon x^i\geqslant 0,\ i=1,\ \ldots,\ n\}$).

Итак, пусть x_0 — решение задачи (3.5), причем

$$\left.\begin{array}{ll} \text{а)} & f'(x_0)\neq 0; \\ \text{б)} & \text{при каждом } i=1,\ \ldots,\ m \text{ либо } \varphi_i(x_0)<0, \\ & \text{либо } \varphi_i'(x_0)\neq 0; \\ \text{в)} & \text{градиенты } \varphi_i'(x_0) \text{ для } i=m+1,\ \ldots,\ k \\ & \textit{линейно независимы.} \end{array}\right\} \quad (3.6)$$

Тогда конусы направлений убывания $\Gamma^0_{x_0}$ функции $f(x)$, конусы возможных направлений $\Gamma^i_{x_0}$ для множеств $\{x\colon \varphi_i(x)\leqslant 0\}$, $i=1,\ \ldots,\ m$, конус касательных к множеству $\{x\colon \varphi_i(x)=0$ для $i=m+1,\ \ldots,\ k\}$ направлений $\Gamma^{m+1}_{x_0}$ и конус возможных направлений Γ_{x_0} для множества G выглядят следующим образом:

$$\Gamma^0_{x_0}=\{e\colon (f'(x_0),\ e)<0\},$$

$$\Gamma^i_{x_0}=\begin{cases} E_n, & \text{если } \varphi_i(x_0)<0, \\ \{e\colon (\varphi_i'(x_0),\ e)<0\} & \text{в противном случае,} \end{cases}$$

$$\Gamma^{m+1}_{x_0}=\{e\colon (\varphi_i'(x_0),\ e)=0 \text{ при } i=m+1,\ \ldots,\ k\},$$

$$\Gamma_{x_0}=\{e\colon e=\nu(x-x_0),\ \nu>0,\ x-\text{внутренняя точка } G\}.$$

Все они непусты и выпуклы. Поэтому существуют не равные нулю одновременно векторы $c_0\in(\Gamma^0_{x_0})^*$, $c_i\in(\Gamma^i_{x_0})^*$ для $i=1,\ \ldots,\ m$, $c_{m+1}\in(\Gamma^{m+1}_{x_0})^*$ и $c\in\Gamma^*_{x_0}$ такие, что

$$c_0+c_1+\ldots+c_{m+1}+c=0. \quad (3.7)$$

Посмотрим, что представляют собой в рассматриваемом случае сопряженные конусы. Начнем с $(\Gamma^0_{x_0})^*$. По определению — это множество векторов c_0, для которых неравенство

$$(c_0,\ e)\geqslant 0$$

выполнено для всех e из $\Gamma^0_{x_0}$, а потому и при всех e, принадлежащих замыканию $\Gamma^0_{x_0}$, т. е. удовлетворяющих неравенству

$$(f'(x_0),\ e)\leqslant 0.$$

Сразу видно, что конус $(\Gamma^0_{x_0})^*$ содержит всевозможные векторы вида $(-\lambda_0 f'(x_0))$, где $\lambda_0\geqslant 0$ Нетрудно убедиться и в том, что никаких других векторов в нем нет. Дейст-

вительно, любой вектор c, не принадлежащий **выпуклому** замкнутому множеству

$$\{c_0:\ c_0 = -\lambda_0 f'(x_0),\ \lambda_0 \geqslant 0\},$$

можно строго отделить от него — найти такой **ненулевой** вектор e, что

$$(-\lambda_0 f'(x_0),\ e) > (c,\ e) \tag{3.8}$$

при любом $\lambda_0 \geqslant 0$. Полагая здесь $\lambda_0 = 0$, получим

$$(c,\ e) < 0,$$

а будучи поделенным на λ_0 при $\lambda_0 \to \infty$, неравенство (3.8) дает

$$(f'(x_0),\ e) \leqslant 0.$$

Значит, c не принадлежит $(\Gamma^0_{x_0})^*$.

Мы показали, что

$$\Gamma^{0*}_{x_0} = \{c_0:\ c_0 = -\lambda_0 f'(x_0),\ \lambda_0 \geqslant 0\}. \tag{3.9}$$

Точно так же

$$(\Gamma^i_{x_0})^* = \{c_i:\ c_i = -\lambda_i \varphi_i'(x_0),\ \lambda_i \geqslant 0\}$$

для тех $i \leqslant m$, при которых $\varphi_i(x_0) = 0$. Если же $\varphi_i(x_0) < 0$, т. е. конус $\Gamma^i_{x_0}$ совпадает со всем пространством E_n, сопряженный конус $(\Gamma^i_{x_0})^*$ состоит из единственного вектора — нуля. Следовательно, при всех $i \leqslant m$ имеем

$$(\Gamma^i_{x_0})^* = \{c_i:\ c_i = -\lambda_i \varphi_i'(x_0),\ \lambda_i \geqslant 0,\ \lambda_i \varphi_i(x_0) = 0\}. \tag{3.10}$$

Далее, используя рассуждения, аналогичные тем, которые привели к равенству (3.9), легко проверить, что

$$(\Gamma^{m+1}_{x_0})^* = \left\{c_{m+1}:\ c_{m+1} = -\sum_{i=m+1}^{k} \lambda_i \varphi_i'(x_0)\right\}. \tag{3.11}$$

Таким образом, учитывая (3.9) — (3.11), равенство (3.7) можно переписать так:

$$\begin{gathered} c = \lambda_0^0 f'(x_0) + \sum_{i=1}^{k} \lambda_i^0 \varphi_i'(x_0) \in \Gamma^*_{x_0}, \\ \lambda_i^0 \geqslant 0, \quad i = 0,\ 1,\ \ldots,\ m, \\ \lambda_i^0 \varphi_i(x_0) = 0, \quad i = 1,\ \ldots,\ m, \end{gathered} \tag{3.12}$$

где λ_i^0, $i = 0,\ \ldots,\ k$, — некоторые множители, которые не **должны быть** равны нулю одновременно (данное условие

эквивалентно требованию нетривиальности набора векторов $c_0, \ldots, c_{m+1}, c$).

Рассмотрим теперь конус $\Gamma^*_{x_0}$. Он состоит из векторов c, удовлетворяющих неравенству

$$\nu(c, x - x_0) \geqslant 0$$

при всех $\nu > 0$ и любых x из внутренности множества G. Поделив это неравенство на $\nu > 0$ и учитывая, что в любой окрестности граничной точки выпуклого множества G найдутся его внутренние точки, видим, что для $c \in \Gamma^*_{x_0}$ при всех $x \in G$ будет

$$(c, x - x_0) \geqslant 0, \tag{3.13}$$

т. е.

$$\Gamma^*_{x_0} = \{c: (c, x) \geqslant (c, x_0), x \in G\}.$$

Если x_0 — внутренняя точка G, конус $\Gamma^*_{x_0}$ включает нуль и ничего больше, а в противном случае содержит также векторы, совпадающие по направлению с нормалями опорных к множеству G в точке x_0 плоскостей (т. е. плоскостей, проходящих через x_0 и лежащих «под» множеством G в смысле неравенства (3.13)). Эти векторы называют опорными функционалами множества G. Проще всего их совокупность (конус $\Gamma^*_{x_0}$) описывается в случае, когда

$$G = \{x: x^i \geqslant 0, \quad i = 1, \ldots, n\}.$$

При этом

$$\Gamma^*_{x_0} = \{c: c^i \geqslant 0, i = 1, \ldots, n; (c, x_0) = 0\}.$$

Итак, установлено, что для решения x_0 задачи (3.5), удовлетворяющего требованиям (3.6), должны найтись не все равные нулю множители λ_i^0, $i = 0, 1, \ldots, k$, такие, что

$$\begin{gathered} \lambda_i^0 \geqslant 0, \quad i = 0, 1, \ldots, m, \\ \lambda_i^0 \varphi_i(x_0) = 0, \quad i = 1, \ldots, m, \\ \left(\lambda_0^0 f'(x_0) + \sum_{i=1}^{k} \lambda_i^0 \varphi_i'(x_0), x - x_0\right) \geqslant 0 \end{gathered} \tag{3.14}$$

для всех x, принадлежащих множеству G. В частности,

если $G=\{x:\ x^i\geqslant 0,\ i=1,\ \ldots,\ n\}$, будет

$$\left(\lambda_0^0 f'(x_0)+\sum_{i=1}^{k}\lambda_i^0\varphi_i'(x_0)\right)_j\geqslant 0,\quad j=1,\ \ldots,\ n,$$

$$\left(\lambda_0^0 f'(x_0)+\sum_{i=1}^{k}\lambda_i^0\varphi_i'(x_0)\right)_j=0,\quad \text{если } x_0^j>0,$$

где индексом j внизу обозначена j-я компонента вектора $\lambda_0^0 f'(x_0)+\sum\limits_{i=1}^{k}\lambda_0^0\varphi_i'(x_0)$.

Сформулированные необходимые условия оптимальности называются обобщенным правилом множителей Лагранжа, а величины λ_i^0 — обобщенными множителями Лагранжа. Эти условия мы получили, исходя из более общего утверждения, применить которое к задаче (3.5) удается, вообще говоря, только при выполнении ограничений (3.6). Однако коль скоро конкретные условия выписаны, можно попытаться обобщить их на случаи, в которых ограничения (3.6) нарушаются, и легко убедиться, что это действительно можно сделать. Возьмем, к примеру, случай, когда $f'(x_0)=0$. Тогда при $\lambda_0^0=1$, $\lambda_i^0=0$, $i=1,\ \ldots,\ k$, во-первых, выполнены соотношения (3.14), и, во-вторых, вектор $\lambda_0^0 f'(x_0)+\sum\limits_{i=1}^{k}\lambda_i^0\varphi_i'(x_0)=0$ принадлежит конусу $\Gamma_{x_0}^*$ опорных к множеству G функционалов. Подобным же образом можно снять и ограничения б), в) в (3.6). Следовательно, справедлива

Теорема 3.1 (*обобщенное правило множителей Лагранжа*). *Пусть x_0 — решение задачи* (3.5). *Тогда существуют не все равные нулю множители λ_i^0, $i=0,\ \ldots,\ k$, такие, что*

$$\lambda_i^0\geqslant 0,\quad i=0,\ \ldots,\ m,$$
$$\lambda_i^0\varphi_i(x_0)=0,\quad i=1,\ \ldots,\ m,$$

и при всех $x\in G$ выполнено неравенство

$$\left(\lambda_0^0 f'(x_0)+\sum_{i=1}^{k}\lambda_i^0\varphi_i'(x_0),\ x-x_0\right)\geqslant 0.$$

Проводя рассуждения, совершенно аналогичные тем, которые привели к теореме 3.1, можно получить следую-

щие условия оптимальности для более простых, чем (3.5), задач с *однотипными ограничениями*.

Теорема 3.2. *Для того чтобы точка x_0 была решением задачи*

$$\min f(x),$$
$$\varphi_i(x) \leqslant 0, \quad i = 1, \ldots, m,$$
$$x \in G \subset E_n,$$

где f, φ_i — непрерывно дифференцируемые функции, а G — замкнутое выпуклое множество с непустой внутренностью, необходимо существование не равных нулю одновременно множителей $\lambda_i^0 \geqslant 0$, $i = 0, 1, \ldots, m$, таких, что

$$\left(\lambda_0^0 f'(x_0) + \sum_{i=1}^{m} \lambda_i^0 \varphi_i'(x_0),\ x - x_0\right) \geqslant 0$$

при всех $x \in G$ и

$$\lambda_i^0 \varphi_i(x_0) = 0, \quad i = 1, \ldots, m.$$

Теорема 3.3. *Для того чтобы точка x_0 была решением задачи*

$$\min f(x),$$
$$\varphi_i(x) = 0, \quad i = 1, \ldots, m,$$
$$x \in G \subset E_n,$$

где f, φ_i — непрерывно дифференцируемые функции, а G — замкнутое выпуклое множество с непустой внутренностью, необходимо существование не равных нулю одновременно множителей λ_i^0, $i = 0, 1, \ldots, m$, таких, что $\lambda_0^0 \geqslant 0$ и

$$\left(\lambda_0^0 f'(x_0) + \sum_{i=1}^{m} \lambda_i^0 \varphi_i'(x_0),\ x - x_0\right) \geqslant 0$$

для всех $x \in G$.

Заметим, что в приведенных условиях экстремума множитель λ_0^0 при градиенте целевой функции может оказаться нулем. Это означает, что соответствующая задача в определенном смысле является вырожденной — для нее необходимые условия не отражают того, что́ именно оптимизируется. Такие задачи встречаются довольно редко, но все же встречаются и нужно уметь отличать их от невырожденных. Один из рецептов, позволяющих описывать классы невырожденных задач, дает теорема Милютина —

Дубовицкого. В частности, применительно к задаче (3.5), удовлетворяющей требованиям (3.6), равенство нулю коэффициента λ_0^0 в обобщенном правиле множителей Лагранжа эквивалентно равенству нулю вектора $c_0 \in (\Gamma_{x_0}^0)^*$ в условиях Эйлера — Лагранжа (3.7), которые, соответственно, принимают вид

$$c_1 + c_2 + \ldots + c_{m+1} + c = 0,$$

причем среди векторов $c_i \in (\Gamma_{x_0}^i)^*$, $i = 1, \ldots, m$, $c_{m+1} \in (\Gamma_{x_0}^{m+1})^*$, $c \in \Gamma_{x_0}^*$ есть ненулевые. Отсюда, в точности повторив рассуждения второй половины доказательства теоремы Милютина — Дубовицкого для системы конусов $\Gamma_{x_0}^i$, $i = 1, \ldots, m$, Γ_{x_0}, $\Gamma_{x_0}^{m+1}$, получим равенство

$$\Gamma_{x_0}^1 \cap \Gamma_{x_0}^2 \cap \ldots \cap \Gamma_{x_0}^m \cap \Gamma_{x_0} \cap \Gamma_{x_0}^{m+1} = \emptyset.$$

Значит, если известно, что пересечение конусов возможных и касательных направлений для рассматриваемой задачи непусто, равенство $\lambda_0^0 = 0$ исключается. Всевозможные достаточные условия непустоты этого пересечения будут выделять классы невырожденных задач (3.5).

Наиболее просто условия, гарантирующие отличие λ_0^0 от нуля в обобщенном правиле множителей Лагранжа, получаются для тех задач вида (3.5), в которых $G = E_n$. Само правило при этом формулируется так: для того чтобы точка x_0 была решением задачи

$$\begin{aligned} &\min f(x), \\ &\varphi_i(x) \leqslant 0, \quad i = 1, \ldots, m, \\ &\varphi_i(x) = 0, \quad i = m+1, \ldots, k, \end{aligned} \tag{3.15}$$

необходимо существование множителей λ_i^0, $i = 0, 1, \ldots, k$, таких, что

$$\begin{aligned} &\lambda_i^0 \geqslant 0, \quad i = 0, \ldots, m, \\ &\lambda_i^0 \varphi_i(x_0) = 0, \quad i = 1, \ldots, m, \\ &\lambda_0^0 f'(x_0) + \sum_{i=1}^{k} \lambda_i^0 \varphi_i'(x_0) = 0 \end{aligned} \tag{3.16}$$

(это — прямое следствие теоремы 3.1, поскольку при $G = E_n$ неравенство

$$(c,\ x - x_0) \geqslant 0$$

может быть выполнено для всех $x \in G$ только при $c = 0$). Множитель λ_0^0 в (3.16) не может быть равен нулю (т. е.

$\lambda_0^0 > 0$), например, если векторы $\varphi_i'(x_0)$, $i = m+1, \ldots, k$, и $\varphi_i'(x_0)$ для тех $i \leqslant m$, при которых $\varphi_i(x_0) = 0$, линейно независимы. Данное утверждение проверяется непосредственно — легко убедиться, что при $\lambda_0^0 = 0$ равенства (3.16) в рассматриваемом случае несовместны. Однако его можно трактовать и как результат применения изложенного выше общего рецепта выработки гарантий соблюдения неравенства $\lambda_0^0 > 0$: дело в том, что линейная независимость перечисленных выше градиентов есть достаточное условие непустоты пересечения конусов $\Gamma_{x_0}^i$, $i = 1, \ldots, m$, $\Gamma_{x_0} = E_n$, $\Gamma_{x_0}^{m+1}$.

Используя полученное условие, обеспечивающее выполнение неравенства $\lambda_0^0 > 0$, и разделив все соотношения в (3.16) на λ_0^0, приходим к следующей теореме.

Теорема 3.4 *(необходимые условия Куна — Таккера). Пусть x_0 — решение задачи (3.15), причем градиенты $\varphi_i'(x_0)$, $i = m+1, \ldots, k$, и $\varphi_i'(x_0)$ для $i \leqslant m$ таких, что $\varphi_i(x_0) = 0$, линейно независимы. Тогда существуют множители λ_i^0, $i = 1, \ldots, k$, удовлетворяющие соотношениям*

$$\begin{aligned} &\lambda_i^0 \geqslant 0, \quad i = 1, \ldots, m, \\ &\lambda_i^0 \varphi_i(x_0) = 0, \quad i = 1, \ldots, m, \\ &f'(x_0) + \sum_{i=1}^{k} \lambda_i^0 \varphi_i'(x_0) = 0. \end{aligned} \tag{3.17}$$

Аналогичные утверждения для задач с однотипными ограничениями формулируются на базе теорем 3.2 и 3.4, причем для задачи с равенствами это будет не что иное, как обычное правило множителей Лагранжа (см. § 3 гл. 1).

Следует отметить, что, хотя условия — равенства в (3.17) и ограничения $\varphi_i(x_0) = 0$, $i = m+1, \ldots, k$, образуют систему из $(k+n)$ уравнений с $(k+n)$ неизвестными λ_i^0, $i = 1, \ldots, k$, x_0^i, $i = 1, \ldots, n$, попытки найти отимальную точку x_0 как решение этой системы, применяя к ней, скажем, метод Ньютона, иногда оканчиваются неудачно и это связано с природой уравнений $\lambda_i \varphi_i(x) = 0$. Поэтому прикладное значение доказанных теорем в основном состоит в том, что они позволяют исследовать свойства сходимости различных методов решения задач типа (3.5). Необходимые условия оптимальности, непосредственно используемые для построения численных методов, удается

получить только для особого класса задач поиска экстремума при наличии ограничений. Это — задачи выпуклого программирования, которыми мы сейчас и займемся.

4. Теорема Куна—Таккера. Рассмотрим задачу выпуклого программирования вида

$$\begin{gathered}\min f(x),\\ \varphi_i(x)\leqslant 0,\quad i=1,\ \ldots,\ m,\\ x\in G\subset E_n,\end{gathered}\tag{3.18}$$

где f, φ_i — выпуклые функции, а G — выпуклое замкнутое множество, имеющее внутренние точки. Обозначим через x_0 ее решение и предположим, что для каждой из функций $f(x)-f(x_0)$, $\varphi_i(x)$, $i=1,\ \ldots,\ m$, существует непустое подмножество пространства E_n, на котором она принимает отрицательные значения. Конусы направлений убывания $\Gamma^0_{x_0}$ функции $f(x)$ и возможных направлений $\Gamma^i_{x_0}$, Γ_{x_0} для множеств $G_i=\{x\colon \varphi_i(x)\leqslant 0\}$, G в данном случае выглядят следующим образом:

$$\Gamma^0_{x_0}=\{e\colon e=\nu(x-x_0),\ \nu>0,\ f(x)<f(x_0)\},$$

$$\Gamma^i_{x_0}=\begin{cases}E_n, & \text{если } \varphi_i(x_0)<0,\\ \{e\colon e=\nu(x-x_0),\ \nu>0,\ \varphi_i(x)<0\}, & \text{если } \varphi_i(x_0)=0,\end{cases}$$

$$\Gamma_{x_0}=\{e\colon e=\nu(x-x_0),\ \nu>0,\ x\in G^0\},$$

где G^0 — внутренность множества G. Их пересечение в силу леммы 3.1 должно быть пустым (конус касательных направлений $\Gamma^{m+1}_{x_0}$ при отсутствии ограничений — равенств, по определению, есть E_n), и так как все они выпуклы и не пусты, теорема Милютина — Дубовицкого гарантирует существование не равных одновременно нулю векторов

$$c_0\in(\Gamma^0_{x_0})^*,\quad c_i\in(\Gamma^i_{x_0})^*,\quad i=1,\ \ldots,\ m,\quad c\in\Gamma^*_{x_0}$$

таких, что

$$c_0+c_1+\ldots+c_m+c=0.\tag{3.19}$$

Чтобы получить из этого равенства конструктивные условия оптимальности для задачи (3.18), нужно понять, из чего состоят сопряженные конусы. Начнем, как всегда, с $(\Gamma^1_{x_0})^*$.

Конус $(\Gamma^1_{x_0})^*$ представляет собой множество векторов c_0, для которых при всех x, удовлетворяющих неравенству

$f(x) < f(x_0)$, и всех $\nu > 0$ будет

$$\nu(c_0,\ x - x_0) \geqslant 0,$$

т. е. для любого $c_0 \in (\Gamma^0_{x_0})^*$ из

$$f(x) < f(x_0)$$

следует

$$(c_0,\ x - x_0) \geqslant 0.$$

Возьмем какой-нибудь один вектор $c_0 \in (\Gamma^0_{x_0})^*$ и для него построим в пространстве E_2 множество Y, куда включим те точки $y = \{y^1,\ y^2\}^T$, к каждой из которых можно подобрать x из E_n так, чтобы выполнялись соотношения

$$y^1 = (c_0,\ x - x_0),$$
$$y^2 \geqslant f(x) - f(x_0).$$

Нетрудно проверить, что Y — выпуклое множество, причем в силу определения c_0 оно не пересекается с отрицательным ортантом $E_2^- = \{\alpha:\ \alpha \in E_2,\ \alpha^1 < 0,\ \alpha^2 < 0\}$. Значит, его можно отделить от E_2^-, т. е. найти ненулевой вектор $\mu = \{\mu^1,\ \mu^2\}^T$ такой, что

$$\mu^1\alpha^1 + \mu^2\alpha^2 \leqslant \mu^1 y^1 + \mu^2 y^2$$

при всех $\alpha^1 < 0$, $\alpha^2 < 0$ и всех $y = \{y^1,\ y^2\}^T \in Y$. Положив здесь $y^1 = (c_0,\ x - x_0)$, $y^2 = f(x) - f(x_0)$, получим неравенство

$$\mu^1\alpha^1 + \mu^2\alpha^2 \leqslant \mu^1(c_0,\ x - x_0) + \mu^2(f(x) - f(x_0)),$$

которое должно выполняться при любых $x \in E_n$, $\alpha^1 < 0$, $\alpha^2 < 0$, а это возможно лишь в том случае, когда $\mu^1 \geqslant 0$, $\mu^2 \geqslant 0$ и

$$\mu^1(c_0,\ x - x_0) + \mu^2(f(x) - f(x_0)) \geqslant 0 \qquad (3.20)$$

для всех $x \in E_n$.

Если $c_0 \neq 0$, неравенство (3.20) выполняется для всех $x \in E_n$ только при $\mu^2 > 0$. Действительно, коль скоро $\mu^2 = 0$, величина μ^1 должна быть больше нуля, и из (3.20) при $x = x_0 - c_0$ имеем

$$\mu^1(c_0,\ x - x_0) + \mu^2(f(x) - f(x_0)) = -\mu^1 \| c_0 \|^2 \geqslant 0,$$

т. е. $\| c_0 \|^2 = 0$. Следовательно, при $c_0 \neq 0$ (3.20) можно переписать так:

$$f(x) - f(x_0) \geqslant \left(-\frac{\mu^1}{\mu^2} c_0,\ x - x_0\right).$$

Это означает, что вектор $-\frac{\mu^1}{\mu^2}c_0=\tilde{c}_0$ принадлежит множеству $M_0(x_0)$ опорных к $f(x)$ в точке x_0 функционалов, причем, так как мы предположили существование точек x, в которых $f(x)<f(x_0)$, это множество не содержит нуля. Поэтому $\mu^1\neq 0$ и имеет смысл равенство

$$c_0=-\frac{\mu^2}{\mu^1}\tilde{c}_0=-\lambda_0\tilde{c}_0,\quad \tilde{c}_0\in M_0(x_0),\quad \lambda_0\geqslant 0.$$

Таким образом, произвольный ненулевой вектор $c_0\in (\Gamma^0_{x_0})^*$ представим в виде произведения неположительного множителя на некоторый опорный функционал из $M_0(x_0)$. Понятно, что такое представление возможно и для $c_0=0$. Стало быть,

$$(\Gamma^0_{x_0})^*=\{c_0:\ c_0=-\lambda_0\tilde{c}_0,\ \tilde{c}_0\in M_0(x_0),\ \lambda_0\geqslant 0\}.$$

Совершенно аналогично устанавливается, что

$$(\Gamma^i_{x_0})^*=\{c_i:\ c_i=-\lambda_i\tilde{c}_i,\ \tilde{c}_i\in M_i(x_0),\ \lambda_i\geqslant 0\}$$

для тех $i\leqslant m$, при которых $\varphi_i(x_0)=0$, где через $M_i(x_0)$ обозначено множество опорных к $\varphi_i(x)$ в точке x_0 функционалов. Отсюда, учитывая, что из $\varphi_i(x_0)<0$ вытекает равенство

$$(\Gamma^i_{x_0})^*=\{0\},$$

для любого $i=1,\ \ldots,\ m$ получаем

$$(\Gamma^i_{x_0})^*=\{c_i:\ c_i=-\lambda_i\tilde{c}_i,\ \tilde{c}_i\in M_i(x_0),\ \lambda_i\geqslant 0,\ \lambda_i\varphi_i(x_0)=0\}.$$

Теперь, воспользовавшись представлением конуса $\Gamma^*_{x_0}$, найденным в предыдущем пункте, мы можем равенство (3.19) заменить следующими, эквивалентными ему, условиями: существуют не все равные нулю множители λ^0_i, $i=0,\ 1,\ \ldots,\ m$, такие, что

$$\begin{gathered}(\lambda^0_0\tilde{c}_0+\lambda^0_1\tilde{c}_1+\ldots+\lambda^0_m\tilde{c}_m,\ x-x_0)\geqslant 0\quad\text{при всех}\quad x\in G,\\ \lambda^0_i\geqslant 0,\quad i=0,\ \ldots,\ m,\\ \lambda^0_i\varphi_i(x_0)=0,\quad i=1,\ \ldots,\ m,\\ \tilde{c}_i\in M_i(x_0),\quad i=0,\ \ldots,\ m.\end{gathered}\tag{3.21}$$

Вектор

$$\lambda_0\tilde{c}_0+\lambda_1\tilde{c}_1+\ldots+\lambda_m\tilde{c}_m$$

при любых $\lambda_i\geqslant 0$, $\tilde{c}_i\in M_i(x_0)$, $i=0,\ 1,\ \ldots,\ m$, является опорным функционалом в точке x_0 для обобщенной

функции Лагранжа

$$L(x, \lambda_0, \lambda) = \lambda_0 f(x) + \sum_{i=1}^{m} \lambda_i \varphi_i(x),$$

где $\lambda = \{\lambda_1, \ldots, \lambda_m\}^T$. Поэтому из первого неравенства в (3.21) следует, что для всех $x \in G$ будет

$$L(x, \lambda_0^0, \lambda^0) \geqslant L(x_0, \lambda_0^0, \lambda^0).$$

Равенства же $\lambda_i^0 \varphi_i(x_0) = 0$, $i = 1, \ldots, m$, эквивалентны утверждению о том, что вектор $\lambda^0 = \{\lambda_1^0, \ldots, \lambda_m^0\}^T$ есть решение задачи максимизации $L(x_0, \lambda_0^0, \lambda)$ по $\lambda_i \geqslant 0$, $i = 1, \ldots, m$. Следовательно, пара x_0, λ^0 есть седловая точка функции Лагранжа $L(x, \lambda_0^0, \lambda)$ по $x \in G$ и $\lambda_i \geqslant 0$, $i = 1, \ldots, m$, т. е.

$$L(x_0, \lambda_0^0, \lambda) \leqslant L(x_0, \lambda_0^0, \lambda^0) \leqslant L(x, \lambda_0^0, \lambda^0) \quad (3.22)$$

при любых $x \in G$, $\lambda_i \geqslant 0$, $i = 1, \ldots, m$. Этот результат получен при определенных предположениях относительно задачи (3.18). Однако теперь их можно снять. Действительно, если, например, $f(x) \geqslant f(x_0)$ при всех $x \in E_n$, нужно взять $\lambda_0^0 = 1$, $\lambda_i^0 = 0$, $i = 1, \ldots, m$, и неравенства (3.22) выполняются. Таким образом, справедлива

Теорема 3.5. *Пусть x_0 — решение задачи* (3.18). *Тогда найдутся не все равные нулю множители $\lambda_i^0 \geqslant 0$, $i = 0, 1, \ldots, m$, такие, что*

$$L(x_0, \lambda_0^0, \lambda) \leqslant L(x_0, \lambda_0^0, \lambda^0) \leqslant L(x, \lambda_0^0, \lambda^0) \quad (3.23)$$

при всех $x \in G$, $\lambda_i \geqslant 0$, $i = 1, \ldots, m$.

Для того чтобы множитель λ_0^0 при целевой функции в (3.23) был больше нуля, достаточно существования точки $\tilde{x} \in G$, в которой $\varphi_i(\tilde{x}) < 0$ для $i = 1, \ldots, m$. Это — так называемое *условие Слейтера*. Доказать его достаточность совсем несложно. В самом деле, подстановка $x = \tilde{x}$ в правое неравенство в (3.23) дает

$$\lambda_0^0 f(x_0) + \sum_{i=1}^{m} \lambda_i^0 \varphi_i(x_0) = \lambda_0^0 f(x_0) \leqslant \lambda_0^0 f(\tilde{x}) + \sum_{i=1}^{m} \lambda_i^0 \varphi_i(\tilde{x}),$$

откуда при $\lambda_0^0 = 0$ в силу неравенств $\lambda_i^0 \geqslant 0$, $\varphi_i(\tilde{x}) < 0$, $i = 1, \ldots, m$, следовало бы, что множители λ_i^0, $i = 1, \ldots, m$, тоже равны нулю, а это невозможно, так как среди чисел λ_i^0, $i = 0, \ldots, m$, должны быть положительные.

Достаточность условия Слейтера можно установить и по-другому: оно гарантирует непустоту пересечения конусов $\Gamma^i_{x_0}$, $i=1, \ldots, m$, Γ_{x_0}, что, как мы знаем, исключает возможность равенства нулю вектора c_0 в уравнениях Эйлера — Лагранжа (3.19) и, соответственно, равенства нулю λ_0^0 в условиях (3.23).

Допустим теперь, что некоторая задача (3.18) удовлетворяет условию Слейтера, а x_0 — ее решение. Ему отвечают множители Лагранжа $\lambda_0^0 > 0$, $\lambda_i^0 \geqslant 0$, $i=1, \ldots, m$, обеспечивающие неравенства (3.23). Поделив эти неравенства на λ_0^0 и обозначив через новые λ_i^0, $i=1, \ldots, m$, частные от деления старых λ_i^0 на λ_0^0, придем в рассматриваемом случае к следующим необходимым условиям оптимальности: существуют множители λ_i^0, $i=1, \ldots, m$, такие, что для любых $x \in G$, $\lambda_i \geqslant 0$, $i=1, \ldots, m$, будет

$$L(x_0, \lambda) \leqslant L(x_0, \lambda^0) \leqslant L(x, \lambda^0), \tag{3.24}$$

где

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i \varphi_i(x).$$

Покажем, что соблюдение этих условий не только необходимо, но и достаточно для оптимальности x_0. Действительно, если для каких-нибудь x_0, $\lambda_i^0 \geqslant 0$, $i=1, \ldots, m$, при всех $\lambda_i \geqslant 0$, $i=1, \ldots, m$, оказывается, что

$$L(x_0, \lambda) = f(x_0) + \sum_{i=1}^{m} \lambda_i \varphi_i(x_0) \leqslant$$

$$\leqslant f(x_0) + \sum_{i=1}^{m} \lambda_i^0 \varphi_i(x_0) = L(x_0, \lambda^0)$$

или, что то же самое,

$$\sum_{i=1}^{m} \lambda_i \varphi_i(x_0) \leqslant \sum_{i=1}^{m} \lambda_i^0 \varphi_i(x_0), \tag{3.25}$$

то ясно, что $\varphi_i(x_0) \leqslant 0$, $i=1, \ldots, m$ (иначе левая часть неравенства (3.25) не ограничена сверху на множестве неотрицательных λ_i), и, соответственно,

$$\sum_{i=1}^{m} \lambda_i^0 \varphi_i(x_0) \leqslant 0.$$

С другой стороны, подстановка в (3.25) $\lambda_i = 0$, $i = 1, \ldots, m$, дает

$$\sum_{i=1}^{m} \lambda_i^0 \varphi_i(x_0) \geqslant 0.$$

Таким образом,

$$\sum_{i=1}^{m} \lambda_i^0 \varphi_i(x_0) = 0,$$

откуда, в предположении, что для x_0, $\lambda_i^0 \geqslant 0$, $i = 1, \ldots, m$, при всех $x \in G$ выполнено правое неравенство из (3.24), следует, что

$$L(x_0, \lambda^0) = f(x_0) \leqslant f(x) + \sum_{i=1}^{m} \lambda_i^0 \varphi_i(x) = L(x, \lambda^0)$$

для любой точки $x \in G$. Если эта точка к тому же удовлетворяет условиям $\varphi_i(x) \leqslant 0$, $i = 1, \ldots, m$, т. е. является допустимой для задачи (3.18), отсюда получим

$$f(x_0) \leqslant f(x) + \sum_{i=1}^{m} \lambda_i^0 \varphi_i(x) \leqslant f(x),$$

что и требовалось доказать.

Полученный результат занимает центральное место в теории выпуклого программирования и известен как

Теорема Куна—Таккера. *Точка x_0 является решением задачи выпуклого программирования* (3.18), *удовлетворяющей условию Слейтера, в том и только в том случае, когда существуют ненулевые множители $\lambda_i^0 \geqslant 0$, $i = 1, \ldots, m$, обеспечивающие соблюдение неравенств*

$$L(x_0, \lambda) \leqslant L(x_0, \lambda^0) \leqslant L(x, \lambda^0)$$

при любых $x \in G$, $\lambda_i \geqslant 0$, $i = 1, \ldots, m$.

Помимо разобранных выше задач объектом исследований в выпуклом программировании — разделе теории оптимизации, посвященном проблематике поиска минимумов выпуклых функций на выпуклых множествах, — являются задачи типа

$$\begin{gathered} \min f(x), \\ \varphi_i(x) \leqslant 0, \quad i = 1, \ldots, m, \\ \varphi_i(x) = (a_i, x) - b^i = 0, \ a_i \in E_n, \ i = m+1, \ldots, k, \\ x \in G, \end{gathered} \tag{3.26}$$

где f, φ_i, $i=1, \ldots, m$, — выпуклые функции, G — выпуклое замкнутое множество с непустой внутренностью, a_i, $i=m+1, \ldots, k$, — линейно независимые векторы. Обозначив через x_0 решение этой задачи и предполагая, что для функций f, φ_i, $i=1, \ldots, m$, существуют точки, в которых значения f меньше, чем в x_0, а значения φ_i отрицательны, на основании теоремы Милютина — Дубовицкого и полученных выше представлений конусов $(\Gamma^i_{x_0})^*$, $(\Gamma^i_{x_0})^*$, $i=1, \ldots, m$, $\Gamma^*_{x_0}$ придем к следующему результату: найдутся множители $\lambda^0_i \geqslant 0$, $i=1, \ldots, m$, и вектор $c_{m+1} \in (\Gamma^{m+1}_{x_0})^*$ (это — конус, сопряженный к конусу $\Gamma^{m+1}_{x_0}$ касательных к множеству $\{x\colon (a_i, x)-b^i=0, i=m+1, \ldots, k\}$ направлений), не равные нулю одновременно и удовлетворяющие условиям

$$(\lambda^0_0\tilde{c}_0+\lambda^0_1\tilde{c}_1+\ldots+\lambda^0_m\tilde{c}_m+c_{m+1},\ x-x_0)\geqslant 0$$

при всех $x \in G$,

$$\begin{aligned} &\lambda^0_i\varphi_i(x_0)=0, \quad i=1, \ldots, m,\\ &\tilde{c}_i \in M_i(x_0), \quad i=0, \ldots, m. \end{aligned} \tag{3.27}$$

Здесь $M_0(x_0)$, $M_i(x_0)$, $i=1, \ldots, m$, — множества опорных к функциям f, φ_i, $i=1, \ldots, m$, в точке x_0 функционалов.

Конус $\Gamma^{m+1}_{x_0}$ в рассматриваемом случае совпадает с множеством

$$\{x\colon (a_i, x)-b^i=0, \quad i=m+1, \ldots, k\}.$$

Соответственно,

$$(\Gamma^{m+1}_{x_0})^*=\left\{e\colon e=\sum_{i=m+1}^{k}\lambda_i a_i\right\},$$

причем ясно, что при любых x_0, λ_i, $i=m+1, \ldots, k$, вектор $\sum\limits_{i=m+1}^{k}\lambda_i a_i$ является опорным функционалом для выпуклой функции $\sum\limits_{i=m+1}^{k}\lambda_i((a_i, x)-b^i)$. Поэтому условия (3.27) можно переформулировать следующим образом: найдутся не все равные нулю множители λ^0_i, $i=0, \ldots, k$, такие, что

$$\begin{aligned} &\lambda^0_i \geqslant 0, \quad i=0, \ldots, m,\\ &\lambda^0_i\varphi_i(x_0)=0, \quad i=1, \ldots, m, \end{aligned}$$

и при всех $x \in G$ выполнено неравенство

$$L(x_0, \lambda_0^0, \lambda^0) = \lambda_0^0 f(x_0) + \sum_{i=1}^{k} \lambda_i^0 \varphi_i(x_0) \leqslant \lambda_0^0 f(x) + \sum_{i=1}^{k} \lambda_i^0 \varphi_i(x) = L(x, \lambda_0^0, \lambda^0).$$

Как и прежде, ограничения, первоначально наложенные на задачу, можно снять и, как и прежде, равенства $\lambda_i^0 \varphi_i(x_0) = 0$, $i = 1, \ldots, m$, можно заменить утверждением о том, что λ_i^0, $i = 1, \ldots, m$, доставляют максимум функции $L(x_0, \lambda_0^0, \lambda)$ по $\lambda_i \geqslant 0$. При этом равенства $(a_i, x_0) - b^i = 0$, $i = m+1, \ldots, k$, — необходимые и достаточные условия максимальности $L(x_0, \lambda_0^0, \lambda)$ на множестве произвольных значений λ_i, $i = m+1, \ldots, k$. Следовательно, справедлива

Теорема 3.6. *Пусть x_0 — решение задачи* (3.26). *Тогда существуют не все равные нулю множители λ_i^0, $i = 0, \ldots, k$, такие, что $\lambda_i^0 \geqslant 0$, $i = 0, 1, \ldots, m$, и для функции*

$$L(x, \lambda_0^0, \lambda) = \lambda_0^0 f(x) + \sum_{i=1}^{k} \lambda_i \varphi_i(x)$$

при всех $x \in G$, $\lambda_i \geqslant 0$, $i = 1, \ldots, m$, и любых λ_i, $i = m+1, \ldots, k$, выполнены неравенства

$$L(x_0, \lambda_0^0, \lambda) \leqslant L(x_0, \lambda_0^0, \lambda^0) \leqslant L(x, \lambda_0^0, \lambda^0). \qquad (3.28)$$

Условием, гарантирующим неравенство $\lambda_0^0 > 0$ в (3.28), является существование точки $\tilde{x}$, принадлежащей внутренности множества G, удовлетворяющей ограничениям — равенствам задачи (3.26) и обеспечивающей неравенства $\varphi_i(\tilde{x}) < 0$, $i = 1, \ldots, m$. Такая точка $\tilde{x}$ будет общей для всех конусов $\Gamma_{x_0}^i$, $i = 1, \ldots, m$, Γ_{x_0}, $\Gamma_{x_0}^{m+1}$, и поэтому равенства $c_0 = 0$ в уравнениях Эйлера — Лагранжа и, соответственно, $\lambda_0^0 = 0$ в (3.28) исключаются. Сформулированное условие легко доказать и непосредственно, примерно так же, как доказывалось выше условие Слейтера. При $\lambda_0^0 > 0$ неравенства (3.28) можно разделить на λ_0^0, причем они становятся достаточными условиями оптимальности x_0. Последнее устанавливается путем рассуждений, совершенно аналогичных использовованным выше при доказательстве теоремы Куна — Таккера. Таким образом, справедлива

Теорема 3.7. *Пусть для задачи* (3.26) *найдется такая точка* $\tilde{x} \in G^0$, *что* $\varphi_i(\tilde{x}) < 0$, $i = 1, \ldots, m$, $\varphi_i(\tilde{x}) = 0$, $i = m+1, \ldots, k$. *Тогда* x_0 — *решение этой задачи в том и только в том случае, когда существуют множители* λ_i^0, $i = 1, \ldots, k$, $\lambda_i^0 \geqslant 0$, $i = 1, \ldots, m$, *обеспечивающие при любых* $x \in G$, λ_i, $i = 1, \ldots, k$, $\lambda_i \geqslant 0$, $i = 1, \ldots, m$, *неравенства*

$$L(x_0, \lambda) \leqslant L(x_0, \lambda^0) \leqslant L(x, \lambda^0), \tag{3.29}$$

где $L(x, \lambda) = f(x) + \sum_{i=1}^{k} \lambda_i \varphi_i(x)$.

Теорема Куна — Таккера и ее обобщение — теорема 3.7 — позволяют перейти от задачи выпуклого программирования к задаче поиска седловой точки функции Лагранжа $L(x, \lambda)$ на множестве, заданном простыми ограничениями (включающими условия неотрицательности λ_i и принадлежности x множеству G — в приложениях это множество, как правило, задается условиями неотрицательности всех либо части переменных x^i, $i = 1, \ldots, n$). Возможность такой редукции используется во многих алгоритмах решения задач вида (3.18), (3.26). Кстати сказать, если эти задачи линейны, упомянутые теоремы — не что иное, как соответствующие версии первой теоремы двойственности в линейном программировании.

Полученные выше условия оптимальности для задач выпуклого программирования имеют нелокальный характер и не предполагают дифференцируемость целевой функции и функций ограничений — неравенств. Однако если эти функции дифференцируемы, соответствующие локальные (т. е. формулируемые в терминах производных) условия вытекают из доказанных теорем как элементарные следствия. К примеру, правое неравенство в (3.29) в силу выпуклости по x функции $L(x, \lambda^0)$ возможно при дифференцируемых f, φ_i, $i = 1, \ldots, m$, тогда и только тогда, когда

$$\left(\frac{\partial L}{\partial x}(x_0, \lambda^0), \; x - x_0\right) \geqslant 0$$

для любых $x \in G$. Следовательно, имея в виду, что левое неравенство в (3.29) эквивалентно условиям дополняющей нежесткости, на основании теоремы 3.7 можно утверждать:

Теорема 3.8. *Пусть в задаче выпуклого программирования* (3.26) *функции* f, φ_i, $i=1, \ldots, m$, *дифференцируемы и найдется точка* $\tilde{x} \in G^0$ *такая, что* $\varphi_i(\tilde{x}) < 0$, $i=1, \ldots, m$, $\varphi_i(\tilde{x}) = 0$, $i=m+1, \ldots, k$. *Тогда* x_0 — *решение этой задачи в том и только в том случае, если найдутся множители* λ_i^0, $i=1, \ldots, k$, $\lambda_i^0 \geqslant 0$, $i=1, \ldots, m$, *такие, что*

$$\left(f'(x_0) + \sum_{i=1}^{k} \lambda_i^0 \varphi_i'(x_0),\ x - x_0\right) \geqslant 0$$

для всех $x \in G$ *и*

$$\lambda_i^0 \varphi_i(x_0) = 0, \qquad i=1, \ldots, m.$$

Аналогичным образом на язык производных переводятся и все остальные условия оптимальности, выведенные в данном пункте.

§ 4. Дискретный принцип максимума

1. Постановка задачи. При планировании экономики, производственных процессов, в автоматическом управлении и т. п. возникают задачи оптимального управления дискретного типа, в которых оптимизируемый процесс описывается системой уравнений

$$x_{k+1} = f_k(x_k, u_k), \quad k=0, 1, \ldots,$$

где u_k — управляющее воздействие, а x_k — состояние исследуемой системы в дискретный момент времени k. Дискретность может возникать при оптимизации непрерывных процессов, так как при использовании ЭВМ дифференциальные уравнения заменяются разностными. Кроме того, имеется большое количество задач, дискретных по существу, в которых либо управляющие воздействия поступают в дискретные моменты времени, либо дискретна во времени информация о состоянии процесса, либо исследуемый процесс является многоэтапным (как это имеет место, например, при управлении химическими реакциями). Дискретные задачи оптимального управления есть не более чем специальные задачи нелинейного программирования. В связи с этим при выводе для них условий оптимальности мы будем использовать результаты, полученные в предыдущем

параграфе. Правда, терминологию мы возьмем из общей теории оптимального управления.

Итак, рассмотрим управляемый процесс, который описывается системой разностных уравнений

$$x(k+1)=f_k(x(k),\ u(k)),\quad k=0,\ 1,\ \ldots,\ N-1, \tag{4.1}$$

причем в начальный момент состояние процесса задано:

$$x(0)=a. \tag{4.2}$$

Здесь $x(k)=\{x^1(k),\ \ldots,\ x^n(k)\}^T$ — вектор-столбец из пространства E_n, определяющий состояние системы на k-ом шаге, а вектор $u(k)=\{u^1(k),\ \ldots,\ u^m(k)\}^T$ отражает текущие управляющие воздействия. Мы будем предполагать, что $u(k)$ может принимать значения из некоторого, зависящего от k, множества G_k евклидова пространства E_m, т. е.

$$u(k)\in G_k\subset E_m,\quad k=0,\ \ldots,\ N-1. \tag{4.3}$$

Вектор-функции $f_k=\{f_k^1,\ \ldots,\ f_k^n\}^T$ определены, соответственно, на прямом произведении $E_n\times G_k$.

Число шагов N исследуемого многошагового процесса считаем фиксированным. Составной вектор $u=\{u(0), u(1),\ \ldots,\ u(N-1)\}$ назовем управлением процесса, а вектор $x=\{x(0),\ \ldots,\ x(N)\}$ — его фазовой траекторией. Будем говорить, что управление u допустимо, если его составляющие $u(k)$ удовлетворяют ограничению (4.3).

В принятых обозначениях задача оптимального управления формулируется так: найти такие управления и фазовую траекторию, чтобы удовлетворить системе разностных уравнений (4.1), начальным условиям (4.2), ограничениям (4.3) и получить минимальное на множестве пар $\{x,\ u\}$, подчиняющихся (4.1) — (4.3), значение критерия

$$I(u,\ x)=f_N^0(x_N)+\sum_{k=0}^{N-1} f_k^0(x(k),\ u(k)), \tag{4.4}$$

где $f_k^0,\ k=0,\ 1,\ \ldots,\ N,$ — скалярные функции.

Найденные в результате решения задачи (4.1) — (4.4) управление $\bar{u}$ и траекторию $\bar{x}$ будем называть, соответственно, оптимальным управлением и оптимальной фазовой траекторией.

2. Необходимые условия оптимальности. Принцип максимума. Введем пространство $\tilde{E}$ векторов $y=\{x, u\}$, представляющее собой прямое произведение вида

$$\underbrace{E_n\times E_n\times\ldots\times E_n}_{N+1}\times\underbrace{E_m\times E_m\times\ldots\times E_m}_{N}.$$

В этом пространстве задача (4.1) — (4.4) ставится следующим образом:

$$\begin{gathered}\min F(y),\\ \varphi_k^i(y)=0,\quad i=1,\ \ldots,\ m,\quad k=0,\ \ldots,\ N-1,\\ y^i-a^i=0,\quad i=1,\ \ldots,\ m,\\ y\in G=\underbrace{E_n\times E_n\times\ldots\times E_n}_{N+1}\times G_0\times G_1\times\ldots\times G_{N-1}\subset\tilde{E},\end{gathered}\tag{4.5}$$

где

$$F(y)=f_N^0(y^{n\cdot N+1},\ \ldots,\ y^{n(N+1)})+$$

$$+\sum_{k=0}^{N-1}f_k^0(y^{nk+1},\ \ldots,\ y^{n(k+1)},\ y^{n(N+1)+m\cdot k+1},\ \ldots$$

$$\ldots,\ y^{n(N+1)+m(k+1)}),$$

$$\varphi_k^i(y)=y^{n(k+1)+i}-$$

$$-f_k^i(y^{n\cdot k+1},\ \ldots,\ y^{nk+n},\ y^{n(N+1)+m\cdot k+1},\ \ldots,\ y^{n(N+1)+mk+m}),$$

$$i=1,\ \ldots,\ m,\quad k=0,\ 1,\ \ldots,\ N-1.$$

Предполагая, что функции f_k^0, $k=0,\ \ldots,\ N$, f_k, $k=0,\ \ldots$ $\ldots,\ N-1$, непрерывно дифференцируемы, а множества G_k — замкнуты, выпуклы и имеют внутренние точки, в соответствии с теоремой 3.3 предыдущего параграфа можно утверждать, что если $\tilde{y}$ — решение задачи (4.5), то найдутся не все равные нулю множители $\tilde{\psi}_0\geqslant 0$, $\tilde{\psi}_k^i$, $i=1,\ \ldots,\ m$, $k=0,\ \ldots,\ N$, такие, что

$$\left(\tilde{\psi}_0F'(\tilde{y})+\sum_{k=0}^{N-1}\sum_{i=1}^{m}\tilde{\psi}_{k+1}^i\varphi_k^{i\prime}(\tilde{y})+\sum_{i=1}^{m}\tilde{\psi}_0^i(y^i-a^i)',\ y-\tilde{y}\right)\geqslant 0\tag{4.6}$$

при любом $y\in G$.

Сказать, что $\tilde{y}=\{\tilde{x},\ \tilde{u}\}$ — решение задачи (4.5), или сказать, что $\tilde{x}$, $\tilde{u}$ — оптимальные фазовая траектория и управление в задаче (4.1) — (4.4) — это одно и то же. Поэтому неравенство (4.6) есть необходимое условие оптимальности для задачи (4.1) — (4.4) и осталось только рас-

шифровать его в терминах данной задачи. Чтобы сделать это, нужно поочередно рассматривать (4.6) для тех векторов $y \in G$, которые отличаются от $\tilde{y}$ компонентами то из одного, то из другого сомножителя в образующем G прямом произведении множеств. Начнем с векторов $y \in G$, которые отличаются от $\tilde{y}$ только первыми n компонентами, т. е. составляющей $\tilde{x}(0)$ траектории $\tilde{x}$. Поскольку эти компоненты в силу определения G могут быть любыми, ясно, что для $j = 1, \ldots, n$ неравенство (4.6) дает

$$\tilde{\psi}_0 \frac{\partial F}{\partial y^j}(\tilde{y}) + \sum_{k=0}^{N-1} \sum_{i=1}^{m} \tilde{\psi}_{k+1}^i \frac{\partial \varphi_k^i}{\partial y^j}(\tilde{y}) + \sum_{i=1}^{m} \tilde{\psi}_0^i \frac{\partial (y^i - a^i)}{\partial y^j} = 0$$

или, учитывая вид функций F, φ_k^i,

$$\frac{\partial f_0^0(x(0), u(0))}{\partial x^j(0)}\Bigg|_{\substack{x(0)=\tilde{x}(0) \\ u(0)=\tilde{u}(0)}} - \sum_{i=1}^{m} \tilde{\psi}_1^i \frac{\partial f_0^i(x(0), u(0))}{\partial x^j(0)}\Bigg|_{\substack{x(0)=\tilde{x}(0) \\ u(0)=\tilde{u}(0)}} + \tilde{\psi}_0^j = 0. \qquad (4.7)$$

Точно так же, подставляя в (4.6) всевозможные y, отличающиеся от $\tilde{y}$ компонентами с номерами j, $n \cdot k + 1 < j < n(k+1)$, где $1 \leqslant k \leqslant N-1$, получим

$$\tilde{\psi}_0 \frac{\partial f_k^0(x(k), u(k))}{\partial x^j(k)}\Bigg|_{\substack{x(k)=\tilde{x}(k) \\ u(k)=\tilde{u}(k)}} + \tilde{\psi}_k^j - \sum_{i=1}^{m} \tilde{\psi}_{k+1}^i \frac{\partial f_k^i(x(k), u(k))}{\partial x^j(k)}\Bigg|_{\substack{x(k)=\tilde{x}(k) \\ u(k)=\tilde{u}(k)}} = 0, \qquad (4.8)$$

где $1 \leqslant j \leqslant n$. Наконец, из соблюдения неравенства (4.6) при всех y, которые отличаются от $\tilde{y}$ только компонентами с индексами j, $n \cdot N + 1 \leqslant j \leqslant n(N+1)$, следует, что

$$\tilde{\psi}_0 \frac{\partial f_N^0(x(N))}{\partial x^j(N)}\Bigg|_{x(N)=\tilde{x}(N)} + \tilde{\psi}_N^j = 0 \qquad (4.9)$$

для $1 \leqslant j \leqslant n$. Обозначив через $\tilde{\psi}(k)$ вектор-строку с координатами $\tilde{\psi}_k^i$, $i = 1, \ldots, m$, через $\frac{\partial f_k^0}{\partial x(k)}$ — вектор-

строку с координатами

$$\left.\frac{\partial f_k^0(x(k),\ u(k))}{\partial x^j(k)}\right|_{\substack{x(k)=\tilde{x}(k)\\ u(k)=\tilde{u}(k)}},$$

$j=1,\ \ldots,\ n$, а через $\frac{\partial f_k}{\partial x(k)}$ — матрицу, $(i,\ j)$-й элемент которой равен

$$\left.\frac{\partial f_k^i(x(k),\ u(k))}{\partial x^j(k)}\right|_{\substack{x(k)=\tilde{x}(k)\\ u(k)=\tilde{u}(k)}},$$

равенства (4.9), (4.8), (4 7) можно переписать так:

$$\begin{gathered}\tilde{\psi}(N)=-\tilde{\psi}_0\frac{\partial f_N^0}{\partial x(N)},\\ \tilde{\psi}(k)=-\tilde{\psi}_0\frac{\partial f_k^0}{\partial x(k)}+\tilde{\psi}(k+1)\frac{\partial f_k}{\partial x(k)},\\ k=N-1,\ N-2,\ \ldots,\ 0.\end{gathered}\tag{4.10}$$

Заметим, что при заданных $\tilde{\psi}_0$, $\tilde{x}$, $\tilde{u}$ они определяют $\tilde{\psi}(k)$, $k=N,\ N-1,\ \ldots,\ 0$, единственным образом, причем как уравнения для расчета $\tilde{\psi}(k)$ они разрешены относительно своих неизвестных и при $\tilde{\psi}_0=0$ дают $\tilde{\psi}(k)=0$, $k=0,\ 1,\ \ldots,\ N$. Поскольку среди величин $\tilde{\psi}_0$, $\tilde{\psi}(k)$, $k=0,\ \ldots,\ N$, должны быть ненулевые, последнее означает, что $\tilde{\psi}_0>0$, и можно принять $\tilde{\psi}_0=1$, что мы и сделаем Рассмотрим теперь векторы

$$y'=\{\tilde{x}(0),\ \ldots,\ \tilde{x}(N),\ \tilde{u}(0),\ \ldots$$
$$\ldots,\ \tilde{u}(k-1),\ u(k),\ \tilde{u}(k+1),\ \ldots,\ \tilde{u}(N-1)\}^T,$$

где $u(k)\in G_k$. Каждый из таких y' принадлежит множеству G и, следовательно, удовлетворяет неравенству (4.6), т. е. при любом $u(k)\in G_k$ (с учетом выбора $\tilde{\psi}_0=1$) будет

$$\sum_{j=n(N+1)+km+1}^{n(N+1)+(k+1)m}\left(\frac{\partial F}{\partial y^j}(\tilde{y})+\sum_{k=0}^{N-1}\sum_{i=1}^{m}\tilde{\psi}_{k+1}^i\frac{\partial\varphi_k^i}{\partial y^j}(\tilde{y})+\right.$$
$$\left.+\sum_{i=1}^{m}\tilde{\psi}_0^i\frac{\partial(y^i-a^i)}{\partial y^j}\right)(u(k)-\tilde{u}(k))^{j-n(N+1)-km}\geqslant 0$$

или, что то же самое, выполнится неравенство

$$\left(\frac{\partial f_k^0}{\partial u(k)} - \tilde{\psi}(k+1)\frac{\partial f_k}{\partial u(k)}\right)(u(k) - \tilde{u}(k)) \geqslant 0. \qquad (4.11)$$

Здесь $\frac{\partial f_k^0}{\partial u(k)}$ — вектор-строка с компонентами

$$\left.\frac{\partial f_k^0(x(k), u(k))}{\partial u^j(k)}\right|_{\substack{x(k)=\tilde{x}(k)\\ u(k)=\tilde{u}(k)}}, \quad j=1, \ldots, m,$$

а $\frac{\partial f_k}{\partial u(k)}$ — матрица, (i, j)-й элемент которой равен

$$\left.\frac{\partial f_k^i(x(k), u(k))}{\partial u^j(k)}\right|_{\substack{x(k)=\tilde{x}(k)\\ u(k)=\tilde{u}(k)}}$$

Вводя так называемую функцию Гамильтона

$$H_k(\psi(k+1), x(k), u(k)) =$$
$$= -f_k^0(x(k), u(k)) + \sum_{i=1}^{m} \psi_{k+1}^i f_k^i(x(k), u(k)) =$$
$$= -f_k^0(x(k), u(k)) + \psi(k+1) f_k(x(k), u(k)),$$

неравенство (4.11) можно переписать в виде

$$\frac{\partial H_k}{\partial u(k)}(\tilde{\psi}(k+1), \tilde{x}(k), \tilde{u}(k))(\tilde{u}(k) - u(k)) \geqslant 0,$$

а равенства (4.10) с использованием нового определения записываются так:

$$\tilde{\psi}(N) = \left.\frac{\partial f_N^0(x(N))}{\partial x(N)}\right|_{x(N)=\tilde{x}(N)},$$
$$\tilde{\psi}(k) = \frac{\partial H_k}{\partial x(k)}(\tilde{\psi}(k+1), \tilde{x}(k), \tilde{u}(k)), \quad k = N-1, \ldots, 0. \qquad (4.12)$$

Следовательно, справедлива

Теорема 4.1. *Пусть $\tilde{x}$, $\tilde{u}$ — оптимальные фазовая траектория и управление в задаче* (4.1) — (4.4) *с непрерывно дифференцируемыми функциями f_k^0, f_k и телесными выпуклыми замкнутыми множествами G_k, а $\psi(1), \ldots, \psi(N)$ — решение связанных с $\tilde{x}$, $\tilde{u}$ уравнений* (4.12). *Тогда при каждом $k = 0, \ldots, N-1$ функция*

$$\frac{\partial H_k}{\partial u(k)}(\tilde{\psi}(k+1), \tilde{x}(k), \tilde{u}(k))\, u(k)$$

достигает своего максимума на множестве G_k *при* $u(k) = \tilde{u}(k)$.

В случае, когда при всех $k = 0, 1, \ldots, N-1$ функции f_k выпуклы, а функции f_k линейны относительно управлений $u(k)$, гамильтонианы $H_k(\tilde{\psi}(k+1), \tilde{x}(k), u(k))$ будут вогнутыми по $u(k)$. При этом из

$$\frac{\partial H_k}{\partial u(k)}(\tilde{\psi}(k+1), \tilde{x}(k), \tilde{u}(k))(\tilde{u}(k) - u(k)) \geqslant 0$$

вытекает неравенство

$$H_k(\tilde{\psi}(k+1), \tilde{x}(k), \tilde{u}(k)) - H_k(\tilde{\psi}(k+1), \tilde{x}(k), u(k)) \geqslant 0,$$

т. е. имеет место

Теорема 4.2 (*принцип максимума*). *Пусть* $\tilde{x}$, $\tilde{u}$ — *оптимальные фазовая траектория и управление в задаче* (4.1) — (4.4) *с непрерывно дифференцируемыми выпуклыми и линейными по* $u(k)$ *функциями* f_k^0, f_k, *соответственно, и выпуклыми замкнутыми телесными множествами* G_k, *а* $\tilde{\psi}(1), \ldots, \tilde{\psi}(N)$ — *решение связанных с* $\tilde{x}$, $\tilde{u}$ *уравнений* (4.12). *Тогда при любом* $k = 0, 1, \ldots, N-1$ *гамильтониан*

$$H_k(\tilde{\psi}(k+1), \tilde{x}(k), u(k))$$

достигает своего максимума по $u(k) \in G_k$ *в точке* $\tilde{u}(k)$.

В следующей части данного курса лекций будет рассмотрен аналог задачи (4.1) — (4.4) в непрерывном времени. Мы увидим, что для этого случая утверждение о максимальности гамильтониана на оптимальном управлении справедливо вне предположений о выпуклости и линейности по управлениям функций задачи и о выпуклости и телесности множеств, из которых эти управления выбираются. Однако для дискретных задач эти предположения существенны. В частности, необходимость первого из них иллюстрирует следующий

Пример 4.1. Пусть управляемый процесс описывается скалярными уравнениями вида

$$\begin{aligned} x^1(k+1) &= x^1(k) + 2u(k), \\ x^2(k+1) &= x^2(k) - (x^1(k))^2 + (u(k))^2, \\ k &= 0, 1, \end{aligned} \tag{4.13}$$

и его начальное состояние таково

$$x^1(0) = 3, \quad x^2(0) = 0.$$

Выбор управлений ограничим неравенствами

$$|u(k)| \leqslant 5, \quad k = 0, 1,$$

и будем искать минимум функционала

$$I = -x^2(2).$$

В силу уравнений (4.13) нетрудно получить

$$x^1(1) = 3 + 2u(0),$$
$$x^2(1) = -9 + (u(0))^2,$$
$$x^2(2) = (u(1))^2 - 3(u(0))^2 - 12u(0) - 18,$$

откуда видно, что составляющими оптимального управления $\tilde{u}$ будут

$$\tilde{u}(0) = -2, \quad \tilde{u}(1) = \pm 5.$$

Проверим выполнение необходимых условий. Для этого составим для $k = 0, 1$ функции Гамильтона:

$$H_k(\psi(k+1), x(k), u(k)) = \psi^1(k+1)(x^1(k) + 2u(k)) + \\ + \psi^2(k+1)(x^2(k) - (x^1(k))^2 + (u(k))^2).$$

Уравнения для множителей Лагранжа $\tilde{\psi}^i(k+1)$, соответственно, выглядят так:

$$\tilde{\psi}^i(2) = \frac{\partial x^2(2)}{\partial x^i(2)}, \quad i = 1, 2,$$

$$\tilde{\psi}^1(1) = \frac{\partial H_1(\tilde{\psi}(2), \tilde{x}(1), \tilde{u}(1))}{\partial x^1(1)} = \tilde{\psi}^1(2) - 2\tilde{x}^1(1)\tilde{\psi}^2(2),$$

$$\tilde{\psi}^2(1) = \frac{\partial H_1(\tilde{\psi}(2), \tilde{x}(1), \tilde{u}(1))}{\partial x^2(1)} = \tilde{\psi}^2(2).$$

Из этих уравнений, подставив в них оптимальное значение фазовой координаты $\tilde{x}^1(1) = -1$, получаем

$$\tilde{\psi}^1(2) = 0, \quad \tilde{\psi}^2(2) = 1,$$
$$\tilde{\psi}^2(1) = 1, \quad \tilde{\psi}^1(1) = 2.$$

Отсюда

$$H_0(\tilde{\psi}(1), \tilde{x}(0), u(0)) = (u(0) + 2)^2 - 7,$$
$$H_1(\tilde{\psi}(2), \tilde{x}(1), u(1)) = (u(1))^2 - 6$$

и видно, что $\tilde{u}(0) = -2$ есть точка минимума, а не максимума гамильтониана $H_0(\tilde{\psi}(1), \tilde{x}(0), u(0))$ по $|u(0)| \leqslant 5$: второе уравнение в (4.13) нелинейно по управлению, поэтому

теорема 4.2 в рассматриваемом случае неприменима. Утверждение же теоремы 4.1, разумеется, подтверждается:

$$\frac{\partial H_0}{\partial u(0)}(\tilde{\psi}(1),\ \tilde{x}(0),\ \tilde{u}(0))(\tilde{u}(0)-u(0))=0\cdot(\tilde{u}(0)-u(0))\geqslant 0$$

для всех $|u(0)|\leqslant 5$ и

$$\frac{\partial H_1}{\partial u(1)}(\tilde{\psi}(2),\ \tilde{x}(1),\ \tilde{u}(1))(\tilde{u}(1)-u(1))=$$
$$=2\tilde{u}(1)(\tilde{u}(1)-u(1))\geqslant 0$$

для всех $|u(1)|\leqslant 5$ как при $\tilde{u}(1)=5$, так и при $\tilde{u}(1)=-5$.

3. Достаточные условия оптимальности. В предыдущем пункте было показано, что соблюдение при всех $k=0, 1, \ldots, N-1$ и $u(k)\in G_k$ неравенств

$$H_k(\tilde{\psi}(k+1),\ \tilde{x}(k),\ \tilde{u}(k))\geqslant H_k(\tilde{\psi}(k+1),\ \tilde{x}(k),\ u(k))$$

для фазовой траектории $\tilde{x}$ и управления $\tilde{u}$, удовлетворяющих всем условиям задачи (4.1)—(4.4), в которой функции f_k^0, $k=0, \ldots, N$, f_k, $k=0, \ldots, N-1$, непрерывно дифференцируемы, причем первые выпуклы, а вторые линейны по управлениям, является необходимым условием оптимальности $\tilde{x}$, $\tilde{u}$. Здесь мы установим, что если функции f_k^0 выпуклы, а f_k линейны не только по управлениям, но и по фазовым переменным, эти условия будут также достаточными.

Итак, рассмотрим задачу

$$\min\left(f_N^0(x(N))+\sum_{k=0}^{N-1}f_k^0(x(k),\ u(k))\right), \tag{4.14}$$
$$x(k+1)=A(k)x(k)+B(k)u(k),\quad k=0, \ldots, N-1,$$
$$x(0)=a,$$
$$u(k)\in G_k\subset E_m,\quad k=0, 1, \ldots, N-1,$$

где f_N^0, f_k^0 — непрерывно дифференцируемые выпуклые функции своих аргументов, G_k, $k=0, 1, \ldots, N-1$, — замкнутые выпуклые телесные множества, $A(k)$, $B(k)$, $k=0, 1, \ldots, N-1$, — матрицы размерностей $(n\times n)$ и $(n\times m)$, соответственно Пусть $\tilde{x}$, $\tilde{u}$ — некоторая допустимая пара траекторий этой задачи, а $\tilde{\psi}(1), \ldots, \tilde{\psi}(N)$ — векторы,

найденные из решения уравнений

$$\tilde{\psi}(N) = -\frac{\partial f_N^0}{\partial x(N)},$$

$$\tilde{\psi}(k) = \frac{\partial H_k}{\partial x(k)}(\tilde{\psi}(k+1),\ \tilde{x}(k),\ \tilde{u}(k)) = \tag{4.15}$$

$$= \tilde{\psi}(k+1)\,A(k) - \frac{\partial f_k^1}{\partial x(k)},$$

$$k = 1,\ 2,\ \ldots,\ N-1.$$

(Аргументы у производных $\partial f_k/\partial x(k)$, $k=1, \ldots, N$, здесь, как и выше, опускаем, подразумевая, что они вычислены на траекториях $\tilde{x}$, $\tilde{u}$.) Рассмотрим еще одну допустимую пару траекторий x, u. Для нее имеем

$$\frac{\partial f_N^0}{\partial x(N)}(x(N) - \tilde{x}(N)) +$$

$$+ \sum_{k=0}^{N-1}\left(\frac{\partial f_k^0}{\partial x(k)}(x(k) - \tilde{x}(k)) + \frac{\partial f_k^0}{\partial u(k)}(u(k) - \tilde{u}(k))\right) =$$

$$= \frac{\partial f_N^0}{\partial x(N)}(x(N) - \tilde{x}(N)) + \sum_{k=0}^{N-1}\left(\frac{\partial f_k^0}{\partial x(k)}(x(k) - \tilde{x}(k)) +\right.$$

$$\left. + \frac{\partial f_k^0}{\partial u(k)}(u(k) - \tilde{u}(k))\right) + \sum_{k=0}^{N-1}\tilde{\psi}(k+1)\,(x(k+1) -$$

$$- \tilde{x}(k+1) - A(k)\,(x(k) - \tilde{x}(k)) - B(k)\,(u(k) - \tilde{u}(k))) =$$

$$= \left(\frac{\partial f_N^0}{\partial x(N)} + \tilde{\psi}(N)\right)(x(N) - \tilde{x}(N)) +$$

$$+ \sum_{k=0}^{N-1}\left(\frac{\partial f_k^0}{\partial x(k)} + \tilde{\psi}(k) - \tilde{\psi}(k+1)\,A(k)\right)(x(k) - \tilde{x}(k)) +$$

$$+ \sum_{k=0}^{N-1}\tilde{\psi}(k+1)\left(\frac{\partial f_k^0}{\partial u(k)} - B(k)\right)(u(k) - \tilde{u}(k)) =$$

$$= \sum_{k=0}^{N-1}\tilde{\psi}(k+1)\left(\frac{\partial f_k^0}{\partial u(k)} - B(k)\right)(u(k) - \tilde{u}(k)) =$$

$$= -\sum_{k=0}^{N-1}\frac{\partial H_k}{\partial u(k)}(\tilde{\psi}(k+1),\ \tilde{x}(k),\ \tilde{u}(k))\,(u(k) - \tilde{u}(k)). \tag{4.16}$$

Если вогнутый по $u(k)$ гамильтониан $H_k(\tilde{\psi}(k+1), \tilde{x}(k), u(k))$ достигает своего максимума на выпуклом множестве G_k при $u(k)=\tilde{u}(k)$, ясно, что при любом $u(k)\in G_k$ должно быть

$$\frac{\partial H_k}{\partial u(k)}(\tilde{\psi}(k+1), \tilde{x}(k), \tilde{u}(k))(u(k)-\tilde{u}(k))\leqslant 0.$$

Тогда в силу (4.16) выполняется неравенство

$$\frac{\partial f_N^0}{\partial x(N)}(x(N)-\tilde{x}(N))+ \\ +\sum_{k=0}^{N-1}\left(\frac{\partial f_k^0}{\partial x(k)}(x(k)-\tilde{x}(k))+\frac{\partial f_k^0}{\partial u(k)}(u(k)-\tilde{u}(k))\right)\geqslant 0,$$

а так как функции f_k^0, $k=0, \ldots, N$, выпуклы, этого достаточно, чтобы утверждать, что

$$f_N^0(x(N))+\sum_{k=0}^{N-1} f_k^0(x(k), u(k))\geqslant \\ \geqslant f_N^0(\tilde{x}(N))+\sum_{k=0}^{N-1} f_k^0(\tilde{x}(k), \tilde{u}(k)).$$

Таким образом, справедлива

Теорема 4.3. *Для того чтобы траектории $\tilde{x}$, $\tilde{u}$, удовлетворяющие всем ограничениям задачи* (4.14), *были ее решением, необходимо и достаточно, чтобы при $\tilde{\psi}(0)$, ..., $\tilde{\psi}(N)$, найденных из решения системы* (4.15), *для каждого $k=0, \ldots, N-1$ гамильтониан*

$$H_k(\tilde{\psi}(k+1), \tilde{x}(k), u(k))$$

достигал своего максимума по $u(k)\in G_k$ при $u(k)=\tilde{u}(k)$.

В заключение данного пункта следует сказать, что теорему 4.3 можно было доказать как следствие теоремы 3.7 предыдущего параграфа.

4. Задачи с ограничениями на правый конец фазовой траектории. До сих пор в этом параграфе исследовались разновидности задачи (4.1) — (4.4), в которой на неизвестную $x(N)$ не налагалось никаких ограничений, т. е. правый конец траектории был свободен. Введем теперь дополнительное условие вида

$$\Phi(x(N))=0, \qquad (4.17)$$

где $\Phi - r$-мерная непрерывно дифференцируемая вектор-функция и $r < n$, т. е. рассмотрим задачу поиска фазовой траектории и управления, удовлетворяющих ограничениям:

$$\left.\begin{array}{l} x(k+1) = f_k(x(k),\ u(k)), \\ u(k) \in G_k, \end{array}\right\} \quad k = 0,\ \ldots,\ N-1, \tag{4.18}$$

$$x(0) = a,$$

$$\Phi(x(N)) = 0,$$

и минимизирующих критерий

$$f_N^0(x(N)) + \sum_{k=0}^{N-1} f_k^0(x(k),\ u(k)) \tag{4.19}$$

на множестве пар x, u, для которых эти ограничения выполнены. Так, например, выглядит задача о выводе космического аппарата на орбиту: равенства (4.17) задают требуемые значения параметров его движения (координат, скоростей и т. д.) в момент выключения двигателя; уравнения в (4.18) описывают полет на активном участке траектории; условия $u(k) \in G_k$ отражают возможности системы управления; функция (4.19) обычно имеет смысл интегрального расхода топлива.

Сохранив прежние предположения относительно функций f_k^0, $k = 0,\ \ldots,\ N$, f_k, $k = 0,\ \ldots,\ N-1$, и множеств G_k, $k = 0,\ \ldots,\ N-1$, и применяя ту же схему рассуждений, которая привела нас к теореме (4.1), нетрудно убедиться, что справедлива

Теорема 4.4. *Пусть* $\tilde{x}$, $\tilde{u}$ *— оптимальные траектории задачи минимизации функции* (4.19) *при ограничениях* (4.18). *Тогда существуют не равные нулю одновременно число* $\tilde{\psi}_0 \geqslant 0$, *вектор-строки* $\tilde{\psi}(k)$, $k = 1,\ \ldots,\ N$, *и* r*-мерная вектор-строка* μ *такие, что*

$$\tilde{\psi}(N) = -\tilde{\psi}_0 \frac{\partial f_N^0}{\partial x(N)} + \mu \frac{\partial \Phi}{\partial x(N)},$$

$$\tilde{\psi}(k) = -\tilde{\psi}_0 \frac{\partial f_k^0}{\partial x(k)} + \tilde{\psi}(k+1) \frac{\partial f_k}{\partial x(k)}$$

и при каждом $k = 0,\ \ldots,\ N-1$ *для любых* $u(k) \in G_k$ *выполнены неравенства*

$$\frac{\partial H_k}{\partial u(k)}(\tilde{\psi}(k+1),\ \tilde{x}(k),\ \tilde{u}(k))(\tilde{u}(k) - u(k)) \geqslant 0.$$

В формулировке теоремы использованы прежние обозначения и $\frac{\partial \Phi}{\partial x(N)}$ есть $(r \times n)$-матрица, (i, j)-й элемент которой равен $\left.\frac{\partial \Phi_i}{\partial x^j(N)}\right|_{x(N)=\tilde{x}(N)}$. Равенство

$$\tilde{\psi}(N) = -\tilde{\psi}_0 \frac{\partial f_N^0}{\partial x(N)} + \mu \frac{\partial \Phi}{\partial x(N)},$$

в котором эта матрица фигурирует, называется условием трансверсальности. Оно определяет вектор $\tilde{\psi}(N)$, и поскольку его правая часть может быть ненулевой при $\tilde{\psi}_0 = 0$, неравенство $\tilde{\psi}_0 > 0$ в общем случае гарантировать нельзя. Однако оно будет выполнено, если матрица $\frac{\partial \Phi}{\partial x(N)}$ имеет полный ранг r и за счет выбора соответствующих управлений $u(k) \in G_k$, $k = 0, \ldots, N-1$, решая уравнения

$$\delta x(x+1) = \frac{\partial f_k}{\partial x(k)}(\tilde{x}(k), \tilde{u}(k))\,\delta x(k) + \frac{\partial f_k}{\partial u(k)}(\tilde{x}(k), \tilde{u}(k))\,(u(k) - \tilde{u}(k)),$$
$$k = 0, 1, \ldots, N-1,$$

(где δx — так называемая вариация траектории) при начальном условии

$$\delta x(0) = 0,$$

удается получать любые $\delta x(N)$, достаточно близкие к нулю. Последнее говорит об «управляемости» процесса (4.18) в окрестности траекторий $\tilde{x}$, $\tilde{u}$.

На этом мы закончим изучение условий оптимальности для задач нелинейного программирования. Следующая глава будет посвящена методам их решения.

Глава V

ЧИСЛЕННЫЕ МЕТОДЫ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Введение

В данной главе будут рассмотрены алгоритмы отыскания экстремума нелинейной функции при нелинейных ограничениях. С простейшим из них, предназначенным для решения задач, ограничения которых имеют вид равенств, мы уже познакомились ранее. Это — метод множителей Лагранжа. Здесь речь пойдет о более сложных алгоритмах, рассчитанных на задачи с неравенствами. Надо сказать, что в своем большинстве алгоритмы такого сорта представляют собой различные способы воплощения двух подходов к организации поиска условного экстремума. Первый состоит в том, чтобы, непосредственно контролируя соблюдение ограничений задачи, двигаться к ее оптимуму по последовательности допустимых или «почти» допустимых точек с монотонно убывающими либо монотонно возрастающими (это зависит от того, что ищется — минимум или максимум) значениями целевой функции Соответствующие алгоритмы называются методами спуска. Два из них представлены в первом параграфе этой главы. Второй подход заключается в сведении задачи на экстремум при наличии ограничений к последовательности задач безусловной оптимизации конструируемых специальным образом вспомогательных функций. Эти функции принято называть штрафными, а использующие их алгоритмы — методами штрафных функций. Им посвящен второй параграф.

§ 1. Методы спуска

1. Метод проекции градиента. Описанный ниже алгоритм предназначен для решения задач вида

$$\begin{aligned} &\min f(x), \\ &\varphi_i(x) \leqslant 0, \quad i = 1, \quad , m, \\ &\varphi_i(x) = 0, \quad i = m+1, \ \ldots, \ l, \end{aligned} \tag{1.1}$$

где f, φ_i — непрерывно дифференцируемые функции, и является прямым обобщением метода наискорейшего спуска (см. главу II). Принцип работы и у того, и у другого один — идти в направлении быстрейшего убывания минимизируемой функции. Только в методе наискорейшего спуска, осуществляющем поиск безусловного минимума, это направление есть антиградиент, а в методе проекции градиента, решающем задачи условной оптимизации, оно определяется с учетом ограничений и получается в результате ортогонального проектирования антиградиента на некоторое линейное многообразие. Последнее аппроксимирует участок границы допустимой области, «параллельно» которому будет сделан шаг на очередной итерации. Поскольку граница нелинейна, этот шаг, вообще говоря, выведет из допустимого множества, даже если исходная точка принадлежит ему. Таким образом, в методе проекции градиента возможно движение по недопустимым точкам. Однако степень нарушения ограничений строго контролируется и сохраняется малой за счет корректировок и ограничения длин шагов.

Общая схема одной итерации рассматриваемого алгоритма такова:

а) выделяются ограничения задачи, формирующие границу допустимого множества Ω в окрестности текущей точки x_k, и по их функциям строится многогранник K, аппроксимирующий это множесто вблизи x_k;

б) среди векторов, шаги вдоль которых не выводят из K, выбирается ближайший к антиградиенту $(-f'(x_k))$ вектор p_k (указанные векторы образуют многогранный конус и выбор p_k есть проектирование антиградиента целевой функции на этот конус);

в) точка x_k проектируется (это и есть корректировка, о которой говорилось ранее) на линейное многообразие, образующее грань K, в которой лежит вектор p_k;

г) целевая функция минимизируется на луче, исходящем из найденной проекции точки x_k и направленном вдоль вектора p_k при условии соблюдения ограничений задачи с установленной точностью. Эта схема проиллюстрирована рис. 1.1, на котором изображен один шаг метода проекции градиента для задачи с неравенствами (точка x_{k+1} будет лежать на отрезке L). Займемся теперь конкретизацией описанной схемы.

Введем для задачи (1.1) положительное число ε и будем считать, что точка x удовлетворяет ее условиям с достаточной точностью, если

$$\begin{aligned} &\varphi_i(x) \leqslant \varepsilon, \quad i = 1, \ldots, m, \\ &|\varphi_i(x)| \leqslant \varepsilon, \quad i = m+1, \ldots, l. \end{aligned} \tag{1.2}$$

Предполагая, что в x_k данные неравенства выполнены, обозначим через I набор индексов i, $1 \leqslant i \leqslant l$, таких, что $|\varphi_i(x_k)| \leqslant \varepsilon$. В этот набор, разумеется, войдут номера всех ограничений — равенств и, возможно, номера некоторых

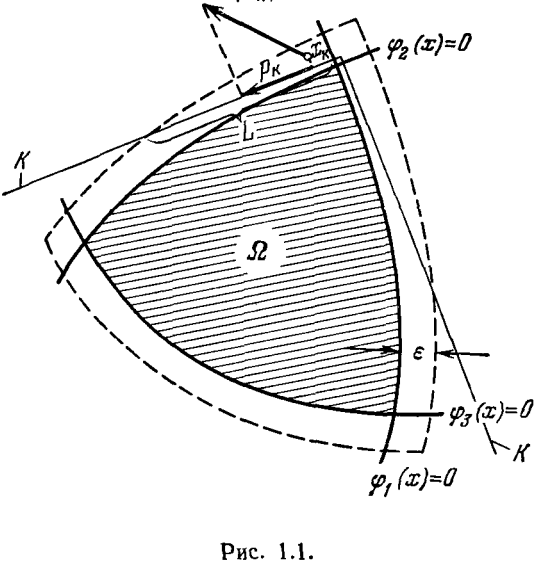


Рис. 1.1.

ограничений — неравенств. Коль скоро все остальные ограничения выполнены в x_k с определенным запасом, граница допустимого множества задачи (1.1) в окрестности x_k будет определяться только функциями $\varphi_i(x)$, $i \in I$. При этом аппроксимирующий многогранник K задается

условиями:

$$\varphi_i(x) \simeq \varphi_i(x_k) + \sum_{j=1}^{n} \frac{\partial \varphi_i}{\partial x^j}(x_k)(x^j - x_k^j) \leqslant 0, \quad i \leqslant m, \ i \in I,$$

$$\varphi_i(x) \simeq \varphi_i(x_k) + \sum_{j=1}^{n} \frac{\partial \varphi_i}{\partial x^j}(x_k)(x^j - x_k^j) = 0, \quad i \geqslant m+1,$$

или, что то же самое,

$$(A_i, \ x - x_k) + \varphi_i(x_k) \leqslant 0, \quad i \leqslant m, \quad i \in I,$$
$$(A_i, \ x - x_k) + \varphi_i(x_k) = 0, \quad i \geqslant m+1.$$

Здесь через A_i обозначен вектор-столбец с компонентами

$$\frac{\partial \varphi_i}{\partial x^1}(x_k), \ \ldots, \ \frac{\partial \varphi_i}{\partial x^n}(x_k).$$

Соответственно, векторы p, шаги вдоль которых не выводят из K, должны обеспечивать соблюдение соотношений

$$(A_i, \ p) \leqslant 0, \quad i \leqslant m, \quad i \in I,$$
$$(A_i, \ p) = 0, \quad i \geqslant m+1.$$

Нам нужно выбрать из них тот, который менее остальных отличается от антиградиента $(-f'(x_k))$, т. е. решить задачу квадратичного программирования

$$\begin{aligned} &\min((f'(x_k)+p), \ (f'(x_k)+p)), \\ &(A_i, \ p) \leqslant 0, \quad i \leqslant m, \quad i \in I, \\ &(A_i, \ p) = 0, \quad i \geqslant m+1. \end{aligned} \tag{1.3}$$

Если векторы A_i, $i \in I$, линейно независимы, а на практике это почти всегда так, для решения задачи (1.3) можно использовать простой алгоритм, на s-м ($s \geqslant 0$) шаге которого решается вспомогательная задача вида

$$\begin{aligned} &\min\{F(p) = ((f'(x_k)+p), \ (f'(x_k)+p))\}, \\ &(A_i, \ p) = 0, \quad i \in I_s \subset I, \end{aligned} \tag{1.4}$$

где множество I_s при $s \geqslant 1$ содержит все $i = m+1, \ \ldots, \ l$ и определяется на предыдущем шаге, а $I_0 = I$. Для решения p_* этой задачи правило множителей Лагранжа дает

такие уравнения:

$$\frac{\partial L}{\partial p}(p_*, \lambda^*) = 2f'(x_k) + 2p_* + \sum_{i \in I_s} \lambda_i^* A_i = \\ = 2f'(x_k) + 2p_* + A\lambda^* = 0, \quad (1.5)$$
$$\frac{\partial L}{\partial \lambda}(p_*, \lambda^*) = A^T p_* = 0.$$

Здесь через A обозначена матрица, составленная из столбцов A_i, $i \in I_s$. Умножая первое из уравнений (1.5) слева на A^T, получим

$$2A^T f'(x_k) + A^T A\lambda^* = 0,$$

а так как в силу предположения о линейной независимости векторов A_i, $i \in I$, и, соответственно, векторов A_i, $i \in I_s$, квадратная матрица $A^T A$ не вырождена, отсюда следует, что

$$\lambda^* = -2(A^T A)^{-1} A^T f'(x_k)$$

и

$$p_* = (A(A^T A)^{-1} A^T - E) f'(x_k).$$

При этом возможны три случая:

а) $(A_i, p_*) \leqslant 0$ для $i \in I$, $i \leqslant m$ и $\lambda_i^* \geqslant 0$ для $i \in I_s$, $i \leqslant m$. Это означает, что p_* — решение задачи (1.3), поскольку в силу выпуклости последней данные неравенства плюс равенства (1.5) являются для нее достаточными условиями оптимальности;

б) существует индекс $i \in I$, $i \leqslant m$ такой, что $(A_i, p_*) > 0$. Тогда положим

$$p_{s+1} = p_s + t_s(p_* - p_s),$$
$$I_{s+1} = \{i: (A_i, p_{s+1}) = 0, i \in I\},$$

где t_s — максимальный шаг, при котором еще будут выполняться неравенства $(A_i, p_{s+1}) \leqslant 0$, $i \in I$, $i \leqslant m$, а p_s — результат предыдущей итерации:

в) $(A_i, p_*) \leqslant 0$ для $i \in I$, $i \leqslant m$ и существует индекс $i_0 \in I_s$, $i_0 \leqslant m$ такой, что $\lambda_{i_0}^* < 0$. В этом случае возьмем $p_{s+1} = p_*$,

$$I_{s+1} = \{i: (A_i, p_{s+1}) = 0, i \in I, i \neq i_0\}.$$

Если реализуется одна из ситуаций б), в), переходим к следующей итерации, и т. д. Нетрудно убедиться, что

на каждом шаге описанного алгоритма значение функции F уменьшается и отсюда следует его конечность. Действительно, случай б) может повторяться подряд не более чем конечное число раз, так как при каждом из таких повторений множество I_s пополняется, по крайней мере, одним новым индексом, а оно всегда содержит не более чем k элементов. Случай же в) не может встретиться бесконечное число раз, так как каждая его реализация в силу убывания $F(p_s)$ означает, что найден новый минимум функции F на некоторой грани допустимого множества задачи (1.3). Но число таких граней конечно и на каждой из них есть только по одной точке минимума $F(p)$. Таким образом, рано или поздно осуществится случай а), т. е. будет найдено решение задачи (1.3).

В соответствии с изложенной схемой метода проекции градиента, прежде чем сделать шаг вдоль вектора p_k, полученного из решения задачи (1.3), нужно спроектировать точку x_k на линейное многообразие, заданное уравнениями

$$(A_i,\ x - x_k) + \varphi_i(x_k) = 0, \quad i \in I_k, \tag{1.6}$$

где I_k есть множество тех индексов $i \in I$, для которых $(A_i,\ p_k) = 0$. Это необходимо, поскольку в точке x_k для некоторых из ограничений с номерами $i \in I_k$ допуски на невязки могут быть исчерпаны и тогда движение из x_k вдоль p_k в рамках требуемой точности соблюдения условий задачи, вообще говоря, окажется невозможным. (Что касается других ограничений, то шаг по направлению p_k только уменьшит их невязки, если таковые имеются, и поэтому они при корректировке не учитываются.) Проекция $\bar{x}_k$ точки x_k на многообразие (1.6) есть решение задачи

$$\min (x - x_k,\ x - x_k),$$
$$(A_i,\ x - x_k) + \varphi_i(x_k) = 0, \quad i \in I_k,$$

применив к которой правило множителей Лагранжа, легко получить такую формулу:

$$\bar{x}_k = x_k + \bar{A}\,(\bar{A}^T A)^{-1}\,\bar{\varphi}(x_k).$$

Здесь $\bar{A}$, $\bar{\varphi}(x_k)$ — матрица и вектор, составленные из столбцов A_i и компонент $\varphi_i(x_k)$ с индексами из множества I_k.

После того как точка $\bar{x}_k$ найдена, для завершения итерации метода проекции градиента остается сделать одно —

решить задачу минимизации вида

$$\min_{\alpha \geqslant 0} f(x_k + \alpha p_k),$$
$$\varphi_i(\bar{x}_k + \alpha p_i) \leqslant \varepsilon, \quad i = 1, \ldots, m,$$
$$|\varphi_i(\bar{x}_k + \alpha p_k)| \leqslant \varepsilon, \quad i = m+1, \ldots, l.$$

В результате будет получена точка $x_{k+1} = \bar{x}_k + \alpha_k p_k$, в которой все повторяется сначала.

Описанный алгоритм при достаточно малых ε, как правило, сходится в точку локального минимума функции $f(x)$ на допустимом множестве (она будет глобальным решением задачи (1.1), если это — задача выпуклого программирования). Признаками сходимости являются, во-первых, стабилизация набора I ограничений, для которых $|\varphi_i(x_k)| \leqslant \varepsilon$, и, во-вторых, стремление к нулю векторов p_k. Однако есть и примеры, в которых метод не сходится (см. [12], стр. 31 — 32). К тому же его быстродействие оставляет желать лучшего (здесь уместно сказать, что иногда эффективнее не решать задачу (1. 3) до конца, ограничившись одной-двумя итерациями представленной выше процедуры).

Существуют значительно более совершенные алгоритмы, относящиеся к тому же, что и рассмотренный, классу методов проектирования, но они достаточно сложны и их изложение выходит за рамки данной книги.

2. Метод возможных направлений. Представленный ниже алгоритм был разработан голландским математиком Г. Зойтендейком и предназначается для поиска экстремума при наличии ограничений только типа неравенств. В отличие от метода проекции градиента, этот алгоритм перебирает не «почти допустимые», а строго допустимые точки задачи, причем, если в методе проекции градиента (применительно к задаче с неравенствами) направление спуска, по сути дела, выбирается из некоторой аппроксимации пересечения конусов возможных направлений и направлений убывания целевой функции, то в методе Зойтендейка оно просто-напросто принадлежит этому пересечению. К тому же в этих алгоритмах реализованы разные подходы к оценке возможностей спуска по различным направлениям: в основе метода проекции градиента лежит представление о том, что направление спуска тем перспективнее, чем бо́льшую локальную скорость убывания

целевой функции оно обеспечит, и при этом рассматриваются направления, не нарушающие ограничений задачи в линейном приближении; в методе Зойтендейка косвенным образом принимается в расчет нелинейность ограничений и фактически делается попытка сравнения направлений не только по локальной скорости убывания целевой функции, но и по длинам шагов, которые удастся сделать вдоль них.

Итак, рассмотрим задачу

$$\min f(x),$$
$$\varphi_i(x) \leqslant 0, \quad i = 1, \ldots, m, \tag{1.7}$$

где f, φ_i — непрерывно дифференцируемые функции. Пусть x_0 — ее допустимая точка, ε_0 — положительное число и I_0 — множество номеров ограничений, выполненных в x_0 с запасом меньшим, чем ε_0:

$$-\varepsilon_0 < \varphi_i(x_0) \leqslant 0, \quad i \in I_0.$$

По смыслу I_0 есть список тех неравенств задачи (1.7), вероятность нарушения которых при перемещениях в окрестности точки x_0 наиболее высока. Они, впрочем, заведомо будут выполняться, если делать не слишком большие шаги из x_0 по тем направлениям p, для которых

$$(\varphi_i'(x_0),\ p) < 0, \quad i \in I_0. \tag{1.8}$$

В случае же, когда найдутся направления, обеспечивающие, помимо (1.8), соблюдение неравенства

$$(f'(x_0),\ p) < 0, \tag{1.9}$$

мы по любому из них наверняка сможем шагнуть из x_0 таким образом, чтобы не покинуть допустимое множество и уменьшить значение функции $f(x)$, т. е. приблизиться к решению задачи (1.7).

Если векторы p, удовлетворяющие (1.8), (1.9), существуют, они образуют телесный конус. Соответственно, приняв решение искать очередное приближение к оптимуму задачи (1.7) в виде $x_1 = x_0 + \alpha_0 p_0$, где p_0 — вектор, подчиняющийся неравенствам (1.8), (1.9), нужно задать правило, по которому p_0 будет выбран из этого конуса. Зойтендейк, в частности, предложил в качестве p_0 брать решение задачи минимизации по p максимальной из

величин

$$(f'(x_0),\ p),\ (\varphi_i'(x_0),\ p),\quad i \in I_0.$$

При этом допустимое множество векторов p должно быть ограниченным — иначе конечного минимума не найти. Чаще всего это множество задают неравенствами

$$\sum_{i=1}^{n} (p^i)^2 \leqslant 1$$

либо

$$|p^i| \leqslant 1,\quad i = 1,\ \ldots,\ n.$$

В последнем случае задача выбора p_0 введением вспомогательной переменной z может быть преобразована к виду

$$\begin{gathered} \min z, \\ (f'(x_0),\ p) \leqslant z, \\ (\varphi_i'(x_0),\ p) \leqslant z,\quad i \in I_0, \\ |p^i| \leqslant 1,\quad i = 1,\ \ldots,\ n, \end{gathered} \tag{1.10}$$

и представляет собой задачу линейного программирования. Следует отметить, что, в отличие от неравенств (1.8), (1.9), она разрешима всегда.

Решение z_0 задачи (1.10) будет удовлетворять одному из двух неравенств:

а) $z_0 < -\varepsilon_0$;

б) $z_0 \geqslant -\varepsilon_0$.

Если реализуется второе, то, обозначив через $\lambda_0^0 \geqslant 0$, $\lambda_i^0 \geqslant 0$, $i \in I_0$, $\underline{\mu}_i \geqslant 0$, $\bar{\mu}_i \geqslant 0$, $i = 1,\ \ldots,\ n$, компоненты решения двойственной к (1.10) задачи, отвечающие ограничениям $(f'(x_0),\ p) - z \leqslant 0$, $(\varphi_i'(x_0),\ p) - z \leqslant 0$, $i \in I_0$, $-p^i \leqslant 1$, $p^i \leqslant 1$, соответственно, получим

$$\lambda_0^0 \frac{\partial f}{\partial x^j}(x_0) + \sum_{i \in I_0} \lambda_i^0 \frac{\partial \varphi_i}{\partial x^j}(x_0) - \underline{\mu}^j + \bar{\mu}^j = 0,$$

$$\lambda_0^0 + \sum_{i \in I_0} \lambda_i^0 = 1,$$

$$\sum_{j=1}^{n} (\underline{\mu}^j + \bar{\mu}^j) = -z_0 \leqslant \varepsilon_0,$$

$$\underline{\mu}^j \cdot \bar{\mu}^j = 0,\quad j = 1,\ \ldots,\ n.$$

Отсюда видно, что

$$\sum_{j=1}^{n}\left|\lambda_0^0\frac{\partial f}{\partial x^j}(x_0)+\sum_{i\in I_0}\lambda_i^0\frac{\partial\varphi_i}{\partial x^j}(x_0)\right|\leqslant\varepsilon_0,$$

и так как в точке x_0 ограничения исходной задачи с номерами $i\in I_0$ с точностью до ε_0 обращаются в равенства, это означает, что x_0 с точностью до ε_0 удовлетворяет необходимым условиям оптимальности, установленным теоремой 3.2 предыдущей главы. Если такой точности достаточно, можно дальше не считать. Если нет — нужно взять новый параметр точности ε_1, меньший чем ε_0, и при выборе направления спуска из точки $x_1=x_0+\alpha_0p_0$ руководствоваться им.

Если же $z_0<-\varepsilon_0$, точность ε_1 для следующей итерации полагают равной ε_0. Что касается величины шага вдоль выбранного направления, то она, как правило, определяется из решения задачи одномерной минимизации вида

$$\min_{\alpha\geqslant0}f(x_k+\alpha p_k),$$
$$\varphi_i(x_k+\alpha p_k)\leqslant0,\quad i=1,\ \ldots,\ m.$$

Таким образом, мы пришли к алгоритму поиска решения задачи (1.7), на k-й итерации которого

а) определяется набор I_k номеров ограничений задачи, для которых в текущей точке x_k выполнены неравенства

$$-\varepsilon_k<\varphi_i(x_k)\leqslant0;$$

б) решается задача линейного программирования

$$\min_{z,\ p}z,$$
$$(f'(x_k),\ p)\leqslant z,$$
$$(\varphi_i'(x_k),\ p)\leqslant z,\quad i\in I_k,$$
$$|p^i|\leqslant1,\quad i=1,\ \ldots,\ n,$$

и если ее решение z_k меньше, чем $(-\varepsilon_k)$, величина ε_{k+1} для следующей итерации полагается равной ε_k, а иначе $\varepsilon_{k+1}=q\varepsilon_k$, где $0<q<1$;

в) определяется точка $x_{k+1}=x_k+\alpha_kp_k$, в которой достигается минимум функции $f(x_k+\alpha p_k)$ по неотрицательным α, удовлетворяющим неравенствам

$$\varphi_i(x_k+\alpha p_k)\leqslant0,\quad i=1,\ \ldots,\ m.$$

Алгоритм может начать работать с любой допустимой точки x_0 и с произвольного значения $\varepsilon_0 > 0$. При этом он сойдется к точке, удовлетворяющей необходимым условиям оптимальности, установленным теоремой 3.2 предыдущей главы. Когда функции f, φ_i выпуклы и выполнено условие Слейтера, эта точка будет решением задачи. В данном случае алгоритм можно использовать и для поиска исходного допустимого приближения x_0, применив его к вспомогательной задаче вида

$$\min_{z,\, x} z,$$

$$\varphi_i(x) \leqslant z, \quad i = 1, \ldots, m.$$

За конечное число шагов он найдет точку x_0, удовлетворяющую неравенствам $\varphi_i(x) < 0$.

Достоинства представленного метода возможных направлений — его надежность и универсальность применительно к задачам с неравенствами. К недостаткам следует отнести невысокую скорость сходимости по числу итераций и большой объем вычислений на каждой итерации. Кстати сказать, хотя в принципе реализуема слегка упрощенная версия метода, в которой параметры ε_k отсутствуют и $I_k = \{i\colon \varphi_i(x_k) = 0\}$, прибегать к такому упрощению не следует — при этом теряются свойства сходимости.

§ 2. Методы штрафных функций

Наряду с алгоритмами типа изложенных в предыдущем параграфе существуют совершенно иные методы поиска условного экстремума, когда решение задачи с ограничениями получается как предел последовательности решений вспомогательных задач безусловной оптимизации подобранных соответствующим образом вспомогательных функций. Они выгодно отличаются от методов спуска простотой реализации и сильными свойствами сходимости. Поэтому на их разработку и разного рода усовершенствования направлялись и продолжают направляться в настоящее время значительные усилия. Первые же из методов, о которых идет речь, появились в самом начале 50-х годов. С них мы и начнем знакомство с большим классом алгоритмов, именуемых методами штрафных функций.

1. Общая схема построения алгоритмов. Рассмотрим задачу отыскания минимума функции $f(x)$ на некотором множестве Ω. Формально она эквивалентна задаче безусловной минимизации суммы

$$f(x)+\delta(x\mid\Omega),$$

где $\delta(x\mid\Omega)$ — так называемая *индикаторная функция*:

$$\delta(x\mid\Omega)=\begin{cases}0, & \text{если } x\in\Omega,\\ +\infty, & \text{если } x\notin\Omega.\end{cases}$$

Последняя, разумеется, совершенно не конструктивна, но, когда множество Ω задано ограничениями типа равенств и неравенств, используя фигурирующие в них функции, совсем нетрудно строить вполне конкретные «штрафы» $\delta_k(x\mid\Omega)$ такие, что при всех $x\in E_n$

$$\lim_{k\to\infty}\delta_k(x\mid\Omega)=\delta(x\mid\Omega). \tag{2.1}$$

Тогда задача, эквивалентная исходной, записывается так:

$$\min\Big(f(x)+\lim_{k\to\infty}\delta_k(x\mid\Omega)\Big),$$
$$x\in E_n.$$

Если операции взятия минимума и предела при этом окажутся перестановочными, мы получим последовательность обычных задач безусловной минимизации вида

$$\min_{x\in E_n}\left(f(x)+\delta_k(x\mid\Omega)\right),$$

пределом решений которых при $k\to\infty$ будет точка минимума функции $f(x)$ на множестве Ω.

Существуют два подхода к построению штрафов $\delta_k(x\mid\Omega)$ и, соответственно, два типа традиционных методов штрафных функций — методы внутренней точки (методы внутренних штрафных или, как их еще называют, барьерных функций) и методы внешней точки (внешних штрафных функций). Мы сначала рассмотрим первые, а потом вторые.

2. Методы внутренних штрафных функций. Когда множество Ω задано с помощью непрерывных функций $\varphi_i(x)$, $i=1,\ldots,m$, ограничениями — неравенствами

$$\varphi_i(x)\geqslant 0,\quad i=1,\ldots,m,$$

причем существуют точки, в которых $\varphi_i(x) > 0$, $i = 1, \ldots, m$, нетрудно построить последовательность штрафов $\delta_k(x \mid \Omega)$, сходящихся при определенных предположениях к индикаторной функции и неограниченно возрастающих при приближении x к границе Ω (рис. 2.1). В данном случае поиск минимума штрафной функции

$$F_k(x) = f(x) + \delta_k(x \mid \Omega) \tag{2.2}$$

нужно начинать с точки, в которой все ограничения исходной задачи выполнены как строгие неравенства. При разумной организации этого поиска последние будут автоматически сохраняться и в дальнейшем: выход на границу Ω, где штраф $\delta_k(x \mid \Omega)$ бесконечен, при минимизации суммы $f(x) + \delta_k(x \mid \Omega)$ смысла не имеет. Таким образом, процесс движения к минимуму функции (2.2) никогда не покинет множества Ω. Отсюда и название — «методы внутренней точки». Сразу отметим, что эти методы неприменимы для решения задач отыскания экстремума на множествах типа того, которое задается в двумерном случае неравенством

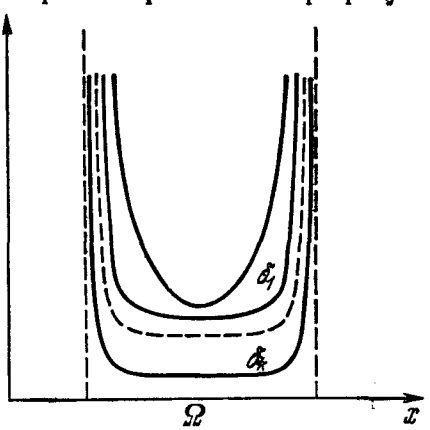


Рис. 2.1.

$$(1 - (x^1)^2 - (x^2)^2)(x^2 - 1)^2 \geqslant 0.$$

Удовлетворяющие ему точки $\{x^1, x^2\}$ лежат в круге единичного радиуса с центром в нуле и на касательной к этому кругу (рис. 2.2). Коль скоро оптимальная точка кругу не принадлежит, ни один из методов внутренних штрафных функций найти ее не сможет.

Чаще всего в качестве штрафов рассматриваемого типа для задач

$$\begin{gathered} \min f(x), \\ \varphi_i(x) \geqslant 0, \quad i = 1, \ldots, m, \end{gathered} \tag{2.3}$$

используют такие функции:

$$\delta_k(x \mid \Omega) = -r_k \sum_{i=1}^{m} \ln \varphi_i(x),$$

$$\delta_k(x \mid \Omega) = r_k \sum_{i=1}^{m} \frac{1}{\varphi_i(x)}.$$

Здесь r_k — положительное число, называемое параметром штрафа. Важно отметить, что если (2.3) — задача выпуклого программирования, т. е. $\varphi_i(x)$ — вогнутые функции, оба предлагаемых штрафа выпуклы. Соотношение (2.1) и для того, и для другого выполняется, если $r_k \to +0$ при $k \to \infty$. Соответственно, алгоритм решения задачи (2.3) с использованием, к примеру, первого из этих штрафов, выглядит так.

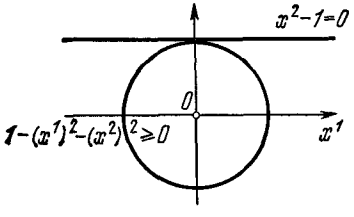


Рис 2 2.

а) выбираются точка x_0, в которой $\varphi_i(x_0) > 0$, $i = 1, \ldots, m$, и монотонно убывающая сходящаяся к нулю последовательность положительных чисел r_k;

б) при $k = 1, 2, \ldots$, начиная с точки x_{k-1}, полученной на предыдущей итерации, решается задача безусловной минимизации по x функции

$$F_k(x) = F(x, r_k) = f(x) - r_k \sum_{i=1}^{m} \ln \varphi_i(x),$$

в результате чего определяется очередная точка x_k.

Если на каждом шаге алгоритма удается найти глобальный минимум $F(x, r_k)$ по x, последовательность $\{x_k\}$ сойдется к глобальному минимуму функции $f(x)$ при ограничениях $\varphi_i(x) \geqslant 0$. Если же x_k — точки локальных безусловных минимумов функции $F(x, r_k)$, то при достаточно естественных предположениях удается установить, что их предел будет локальным решением задачи (2.3). Доказательство первого утверждения приводится ниже. В частности, для задач выпуклого программирования никаких других доказательств сходимости и не требуется, поскольку для них функции $F(x, r_k)$ выпуклы по x и, соответственно, имеют только глобальные минимумы.

3. О сходимости методов внутренних штрафных функций. Для доказательства основного утверждения данного пункта нам потребуется следующая

Лемма 2.1. *Пусть Ω — замкнутое ограниченное множество с непустой внутренностью Ω^0 и $F(x)$ — функция, непрерывная в Ω^0 и неограниченно возрастающая при приближении к границе Ω. Тогда существует точка $\bar{x} \in \Omega^0$ такая, что*

$$F(\bar{x}) = \min_{x \in \Omega^0} F(x).$$

Доказательство. Обозначим через $\bar{v}$ точную нижнюю грань функции $F(x)$ на множестве Ω^0. Тогда найдется последовательность точек $x_k \in \Omega^0$, для которых

$$\lim_{k \to \infty} F(x_k) = \bar{v} < +\infty.$$

Поскольку множество Ω замкнуто и ограничено, не умаляя общности можно считать, что последовательность $\{x_k\}$ сходится к некоторой точке $\bar{x} \in \Omega$. При этом ясно, что $\bar{x}$ не может лежать на границе Ω, поскольку в силу свойств $F(x)$ это означало бы, что

$$\lim_{k \to \infty} F(x_k) = +\infty.$$

Таким образом, $\bar{x}$ принадлежит Ω^0, а отсюда, поскольку функция $F(x)$ непрерывна в Ω^0, следует, что

$$\bar{v} = \lim_{k \to \infty} F(x_k) = F(\bar{x}).$$

Лемма доказана.

Рассмотрим теперь задачу

$$\min_{x \in \Omega} f(x), \tag{2.4}$$

где f — непрерывная функция, а Ω — замкнутое ограниченное множество с непустой внутренностью, причем $\bar{\Omega^0} = \Omega$. Пусть, далее, $\delta_k(x \mid \Omega)$ — неотрицательные функции штрафа, непрерывные в Ω^0, неограниченно возрастающие при приближении к границе множества Ω и сходящиеся при $k \to \infty$ к индикаторной функции. Тогда точки x_k глобальных минимумов штрафных функций

$$F_k(x) = f(x) + \delta_k(x \mid \Omega)$$

на Ω^0, существующие в силу леммы 2.1, сходятся к множеству решений задачи (2.4). Этот факт устанавливает

Теорема 2.1. *Любая предельная точка* $\bar{x}$ *последовательности* $\{x_k\}$ *есть решение задачи* (2.4), *причем, если*

$$\delta_{k+1}(x \mid \Omega) \leqslant \delta_k(x \mid \Omega)$$

при всех k, $x \in \Omega^0$, *то выполнено соотношение*

$$\lim_{k \to \infty} F_k(x_k) = f(\bar{x}).$$

Доказательство. Пусть $\bar{x}$ — некоторая предельная точка последовательности $\{x_k\}$. Поскольку все x_k принадлежат множеству Ω и последнее замкнуто, ясно, что $\bar{x}$ также содержится в Ω. Допустим, что $\bar{x}$ не является при этом решением задачи (2.4). Тогда, учитывая равенство $\bar{\Omega}^0 = \Omega$, можно утверждать, что существует точка $y \in \Omega^0$ такая, что $f(y) < f(\bar{x})$. Обозначив через x_s подпоследовательность точек из $\{x_k\}$, сходящуюся к $\bar{x}$, и используя непрерывность функции $f(x)$, перепишем последнее неравенство так:

$$f(y) < \lim_{s \to \infty} f(x_s).$$

Отсюда, в свою очередь, вытекает существование положительного числа ε, для которого

$$f(y) < f(x_s) - \varepsilon$$

при любых s, а так как $\delta_s(x_s \mid \Omega) \geqslant 0$, это означает, что

$$f(y) < F_s(x_s) - \varepsilon. \tag{2.5}$$

Но для внутренней точки y будет

$$\lim_{s \to \infty} \delta_s(y \mid \Omega) = 0,$$

т. е.

$$\lim_{s \to \infty} F_s(y) = f(y).$$

Поэтому из (2.5) следует, что при достаточно больших s выполнены неравенства

$$F_s(y) < F_s(x_s),$$

а это противоречит определению x_s. Полученное противоречие доказывает оптимальность $\bar{x}$ для задачи (2 4).

Пусть теперь $\delta_{k+1}(x \mid \Omega) \leqslant \delta_k(x \mid \Omega)$ при всех k, $x \in \Omega^0$. Тогда, учитывая определение x_{k+1}, получим

$$F_{k+1}(x_{k+1}) = f(x_{k+1}) + \delta_{k+1}(x_{k+1} \mid \Omega) \leqslant f(x_k) + \delta_{k+1}(x_k \mid \Omega) \leqslant$$
$$\leqslant f(x_k) + \delta_k(x_k \mid \Omega) = F_k(x_k).$$

Таким образом, последовательность $\{F_k(x_k)\}$ является монотонно убывающей. При этом она ограничена снизу величиной $f(\bar{x})$. Значит, существует предел

$$\lim_{k\to\infty} F_k(x_k) = \bar{F} \geqslant f(\bar{x}).$$

Покажем, что неравенство $\bar{F} > f(\bar{x})$ исключено. Действительно, будь оно выполнено, нашлись бы число $\delta > 0$ и близкая к $\bar{x}$ точка $y \in \Omega^0$ такие, что при всех k соблюдались бы неравенства

$$F_k(x_k) > f(y) + \delta.$$

Поскольку $\delta_k(y \mid \Omega) = 0$, отсюда следовало бы, что при достаточно больших k

$$F_k(y) = f(y) + \delta_k(y \mid \Omega) < F_k(x_k).$$

Но это невозможно в силу определения точек x_k. Теорема доказана.

На основании доказанного утверждения можно гарантировать сходимость описанного в предыдущем пункте алгоритма решения задачи (2.3) с использованием любой из двух представленных там же разновидностей внутреннего штрафа в случае, когда функции задачи f, φ_i непрерывны и замыкание множества $\{x\colon \varphi_i(x) > 0,\ i = 1, \ldots, m\}$ совпадает с множеством $\Omega = \{x\colon \varphi_i(x) \geqslant 0,\ i = 1, \ldots, m\}$, причем Ω ограничено. Работу такого алгоритма иллюстрирует

Пример 2.1. Рассмотрим задачу

$$\min (x^1 + x^2),$$
$$\varphi_1(x) = -(x^1)^2 + x^2 \geqslant 0,$$
$$\varphi_2(x) = x^1 \geqslant 0.$$

Множество ее допустимых точек и линии уровня целевой функции изображены на рис. 2.3, откуда видно, что решением является точка с нулевыми координатами. Посмотрим, как будет происходить движение к этой точке в методе с логарифмической штрафной функцией. Применительно к рассматриваемой задаче последняя выглядит так:

$$F(x, r) = x^1 + x^2 - r(\ln(-(x^1)^2 + x^2) + \ln x^1).$$

В точке $x(r)$ безусловного минимума по x функции $F(x, r)$ ее частные производные по x^1, x^2 должны быть равны

нулю:

$$1+r\frac{2x^1}{-(x^1)^2+x^2}-r\frac{1}{x^1}=0,$$

$$1-r\frac{1}{-(x^1)^2+x^2}=0.$$

Неотрицательными решениями этих уравнений будут

$$x^1(r)=\frac{(-1+\sqrt{1+8r})}{4},$$

$$x^2(r)=\frac{(-1+\sqrt{1+8r})^2}{16+r}.$$

Видно, что $x^1(r)\to 0$ и $x^2(r)\to 0$ при $r\to 0$, т. е. сходимость к оптимальной точке исходной задачи есть. Заме-

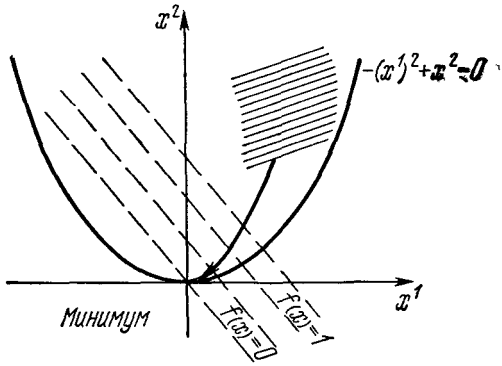


Рис 2 3

тим, кстати, что допустимое множество этой задачи неограничено и, тем не менее, алгоритм сойдется. В этом нет ничего странного, поскольку требование ограниченности Ω при доказательстве теоремы 2.1 было нужно только для того, чтобы гарантировать существование и ограниченность последовательности $\{x_k\}$. Если это обеспечивается иными свойствами задачи, данное требование становится лишним.

4. Методы внешних штрафных функций. В методах внешних штрафных функций штрафы $\delta_k(x\mid\Omega)$, сходящиеся при $k\to\infty$ к индикаторной функции, строят так, чтобы при всех k было

$$\delta_k(x\mid\Omega)=0 \quad \text{для} \quad x\in\Omega$$

и

$$\delta_k(x \mid \Omega) > 0 \quad \text{для} \quad x \notin \Omega.$$

Обычно, как и в методах внутренних штрафных функций, полагают

$$\delta_k(x \mid \Omega) = r_k \Phi(x),$$

только теперь $r_k \to +\infty$ при $k \to \infty$ и $\Phi(x)$ есть функция, определенная на всем пространстве значений x, равная нулю на множестве Ω и положительная за его пределами. Для задач с ограничениями вида

$$\begin{aligned} &\varphi_i(x) \leqslant 0, \quad i = 1, \ldots, m, \\ &\varphi_i(x) = 0, \quad i = m+1, \ldots, l, \end{aligned} \tag{2.6}$$

наиболее распространены две функции $\Phi(x)$:

$$\Phi(x) = \sum_{i=1}^{m} (\varphi_i^-(x))^2 + \sum_{i=m+1}^{l} (\varphi_i(x))^2, \tag{2.7}$$

$$\Phi(x) = \sum_{i=1}^{m} \varphi_i^+(x) + \sum_{i=m+1}^{l} |\varphi_i(x)|. \tag{2.8}$$

Здесь $\varphi_i^+(x)$ — «срезка» функции $\varphi_i(x)$, равная нулю, если $\varphi_i(x) \leqslant 0$, и равная $\varphi_i(x)$, если $\varphi_i(x) > 0$, т. е.

$$\varphi_i^+(x) = \max\{0, \varphi_i(x)\}.$$

Достоинство функции (2.7) по сравнению с (2.8) в том, что если $\varphi_i(x)$, $i = 1, \ldots, l$, непрерывно дифференцируемы, она также будет обладать этим свойством (рис. 2.4). Соответственно, при реализации использующего ее алгоритма для поиска минимумов по x функций

$$F(x, r_k) = f(x) + r_k \Phi(x)$$

можно применять градиентные методы. В то же время негладкая функция (2.8) хороша тем, что уже при конечном значении r_k обеспечивает совпадение точки безусловного минимума суммы $f(x) + r_k \Phi(x)$ с решением исходной задачи (рис. 2.5). Правда, достаточно эффективных алгоритмов минимизации негладких штрафных функций пока нет, и поэтому чаще все же используют гладкий квадратичный штраф. По этой причине и мы в дальнейшем ограничимся рассмотрением алгоритма с гладкой внешней штрафной функцией. Он состоит в следующем:

а) выбираются произвольное начальное приближение x_0 и монотонно возрастающая последовательность чисел $r_k \to \infty$;

б) при $k = 1, 2, \ldots$, начиная с x_{k-1}, решается задача безусловной минимизации по x функции

$$F_k(x) = F(x, r_k) = f(x) + r_k \Phi(x),$$

в результате чего определяется очередное приближение x_k к решению исходной задачи.

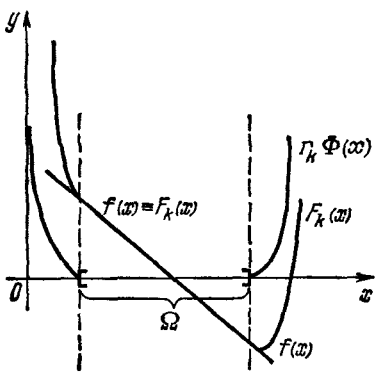


Рис. 2.4.

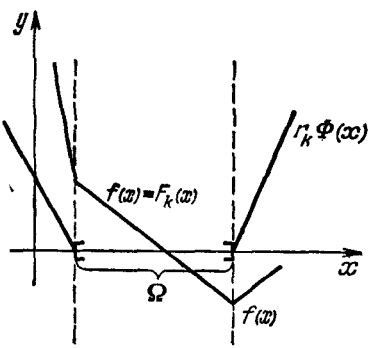


Рис. 2.5.

Прежде, чем дать строгое доказательство сходимости предлагаемого алгоритма, покажем на примере, как он работает

Пример 2.2. Рассмотрим скалярную задачу (рис. 2.6)

$$\max(-x^2 + 4x),$$
$$\varphi(x) = 1 - x \geqslant 0.$$

Ее решение, как легко видеть, есть $\bar{x} = 1$. Квадратичная штрафная функция, которую в данном случае будем максимизировать, имеет вид

$$F(x, r) = -x^2 + 4x - r(\max\{0, x - 1\})^2.$$

Ее производная по x вычисляется по формуле

$$F_x(x, r) = -2x + 4 - 2r \max\{0, x - 1\}.$$

Легко убедиться, что она обращается в нуль в единственной точке

$$x(r) = \frac{4+2r}{2(1+r)},$$

которая и будет точкой максимума $F(x, r)$. Когда $r \to +\infty$, точки $x(r)$ сходятся к решению задачи, причем легко видеть, что все $x(r)$ не удовлетворяют ее ограничениям, т. е движение к оптимуму происходит вне допустимого множества Так бывает всегда, за исключением вырожденных случаев Это объясняет название «метод внешней точки».

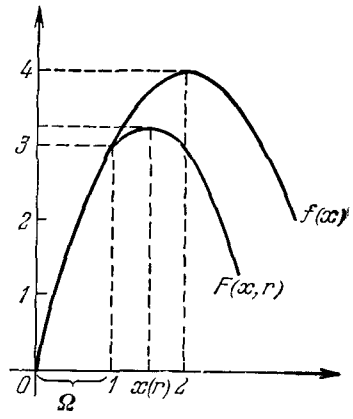


Рис 2 6

5. О сходимости методов внешних штрафных функций. Рассмотрим задачу

$$\min f(x), \quad x \in \Omega, \tag{2.9}$$

где $f(x)$ — непрерывная функция, а Ω — замкнутое множество Допустим, что для нее задана непрерывная функция $\Phi(x)$, равная нулю для любого x из Ω и положительная для всех остальных $x \in E_n$, причем точки $x(r)$ безусловных глобальных минимумов по $x \in E_n$ функций

$$F(x, r) = f(x) + r\Phi(x)$$

существуют и принадлежат при любых $r \geqslant 0$ некоторому ограниченному множеству Y (Это предположение выполнено, например, если найдется точка x' такая, что множество $\{x \mid f(x) \leqslant f(x')\}$ ограничено.) Тогда для любой последовательности чисел $r_k \to \infty$ соответствующая последовательность $\{x(r_k)\}$ ограничена, и мы покажем сейчас, что ее произвольная предельная точка $\bar{x}$ будет решением задачи (2 9) При этом подпоследовательность, пределом которой является $\bar{x}$, далее обозначается через $\{x(r)\}$, а вместо различных пределов по $s \to \infty$, где s — индекс

этой подпоследовательности, фигурируют пределы по $r \to \infty$.

Теорема 2.2. *Точка $\bar{x}$ оптимальна для задачи* (2.9), *причем*

$$f(\bar{x}) = \lim_{r \to \infty} \min_{x \in E_n} F(x, r).$$

Доказательство. Прежде всего, покажем, что

$$\min_{x \in \Omega} f(x) = \min_{x \in E_n} \sup_{r \geqslant 0} F(x, r) = \lim_{r \to \infty} \min_{x \in \Omega} F(x, r) \geqslant$$
$$\geqslant \lim_{r \to \infty} \min_{x \in E_n} F(x, r). \qquad (2.10)$$

Неравенство в этой цепочке сомнений не вызывает, так как при любом r минимум по x функции $F(x, r)$ на всем пространстве E_n не больше ее минимума на содержащемся в E_n множестве Ω. Что же касается равенств, то они следуют из того, что при $x \notin \Omega$ будет $\Phi(x) > 0$, и поэтому

$$\sup_{r \geqslant 0} F(x, r) = \sup_{r \geqslant 0} (f(x) + r\Phi(x)) = +\infty,$$

а при $x \in \Omega$

$$F(x, r) = f(x).$$

Значит,

$$\min_{x \in E_n} \sup_{r \geqslant 0} F(x, r) = \min_{x \in \Omega} \sup_{r \geqslant 0} F(x, r) = \min_{x \in \Omega} f(x) =$$
$$= \lim_{r \to \infty} \min_{x \in \Omega} f(x) = \lim_{r \to \infty} \min_{x \in \Omega} F(x, r).$$

Докажем теперь, что $\bar{x} \in \Omega$. Действительно, в противном случае $\Phi(\bar{x}) > 0$ и в силу непрерывности функции $\Phi(x)$ найдется число $\varepsilon > 0$ такое, что при достаточно больших r будет выполняться неравенство

$$\Phi(x(r)) \geqslant \varepsilon.$$

При этом, в силу ограниченности последовательности $\{f(x(r))\}$,

$$\lim_{r \to \infty} \min_{x \in E_n} F(x, r) \geqslant \lim_{r \to \infty} (f(x(r)) + r \cdot \varepsilon) = +\infty.$$

Но это неравенство противоречит (2.10), следовательно, $\bar{x}$ принадлежит Ω.

Наконец, из неотрицательности $\Phi(x)$ и непрерывности $f(x)$ следует, что

$$\lim_{r\to\infty}\min_{x\in E_n} F(x,\ r)=\lim_{r\to\infty}(f(x(r))+r\Phi(x(r)))\geqslant$$

$$\geqslant\lim_{r\to\infty} f(x(r))=f(\bar{x}).$$

Отсюда и из (2.10) получим

$$f(\bar{x})\leqslant\lim_{r\to\infty}\min_{x\in E_n} F(x,\ r)\leqslant\min_{x\in\Omega} f(x).$$

Так как $\bar{x}$ содержится в Ω, это возможно только, когда

$$f(\bar{x})=\lim_{r\to\infty}\min_{x\in E_n} F(x,\ r)=\min_{x\in\Omega} f(x),$$

что и требовалось доказать.

Мы установили сходимость точек глобальных минимумов внешних штрафных функций к глобальному решению исходной задачи. Если иметь в виду только задачи выпуклого программирования, то этого достаточно, поскольку применяемые на практике внешние штрафные функции для них будут выпуклыми и, соответственно, ни о каких локальных экстремумах говорить не приходится. Что касается невыпуклых задач, то для них при достаточно естественных предположениях можно было бы доказать также сходимость точек локальных минимумов внешних штрафных функций к локальному решению.

6. Сравнительная оценка и общие свойства традиционных методов штрафных функций. Сопоставляя методы внутренних и внешних гладких штрафных функций, в качестве преимущества первых обычно указывают то обстоятельство, что при обращении к ним соблюдение ограничений задачи гарантировано на протяжении всего процесса ее решения. Это важно в случаях, когда целевая функция не определена за пределами допустимого множества и, кроме того, позволяет прервать счет в любой момент времени, получив при этом не какое-то, а допустимое приближение. К недостаткам же внутренних штрафных функций по отношению к внешним следует отнести то, что они имеют смысл только внутри допустимого множества и это обусловливает необходимость использования специальных процедур минимизации, включающих блок проверки соблюдения ограничений, а также их сравнительную сложность. Под сложностью здесь подразумевается

следующее: применив какой-либо из градиентных методов для поиска минимума внутренней штрафной функции, на каждой его итерации придется вычислять производные для всех ограничений задачи, в то время как шаг того же метода при минимизации внешней штрафной функции требует вычисления производных только ялд нарушенных в текущей точке ограничений Наконец, для задачи, поддающейся решению с помощью и внутренних, и внешних штрафов, последние могут оказаться предпочтительнее, потому что не требуют задания допустимой начальной точки, найти которую в общем случае совсем не просто

Помимо указанных качеств, отличающих методы с внутренними штрафами от методов с внешними, есть свойства, общие и для тех, и для других. Мы рассмотрим их на примере штрафной функции

$$F(x, r) = f(x) + \frac{r}{2} \sum_{i=1}^{m} (\varphi_i^+(x))^2,$$

предназначенной для решения задачи

$$\begin{aligned} &\min f(x), \\ &\varphi_i(x) \leqslant 0, \quad i = 1, \ldots, m, \end{aligned} \tag{2.11}$$

причем будем считать, что функции f, φ_i, $i = 1, \ldots, m$, дважды непрерывно дифференцируемы. Можно было бы взять и другую гладкую функцию штрафа — выкладки изменились бы, но основные результаты остались бы прежними.

Итак, пусть $\{r_k\}$ — последовательность монотонно и неограниченно возрастающих положительных чисел, а $x_k = x(r_k)$ — точки минимумов функций $F(x, r_k)$ по $x \in E_n$, сходящиеся к решению x^* задачи (2.11). Тогда, по определению,

$$\frac{\partial F}{\partial x}(x_k, r_k) = f'(x_k) + r_k \sum_{i=1}^{m} \varphi_i^+(x_k) \varphi_i'(x_k) = 0. \tag{2.12}$$

Поскольку

$$\lim_{k \to \infty} x_k = x^*,$$

при достаточно больших k для всех i таких, что $\varphi_i(x^*) < 0$, будет $\varphi_i(x_k) < 0$, т. е. равенство (2.12) примет вид

$$f'(x_k) + r_k \sum_{i \in I} \varphi_i^+(x_k) \varphi_i'(x_k) = 0,$$

где $I=\{i:\ \varphi_i(x^*)=0\}$. Обозначив через λ_i^k произведения $r_k\varphi_i^+(x_k)\geqslant 0$, перепишем последнее равенство так:

$$f'(x_k)+\sum_{i\in I}\lambda_i^k\varphi_i'(x_k)=0. \tag{2.13}$$

Предположим теперь и будем считать в дальнейшем, что градиенты $\varphi_i'(x^*)$, $i\in I$, линейно независимы. Тогда в силу теоремы 3.4 предыдущей главы найдутся множители λ_i^* такие, что

$$f'(x^*)+\sum_{i\in I}\lambda_i^*\varphi_i'(x^*)=0. \tag{2.14}$$

При этом из уравнений (2.13), (2.14) (с учетом сходимости x_k к x^*, линейной независимости векторов $\varphi_i'(x^*)$, $i\in I$, и непрерывности $\varphi_i'(x)$, $i\in I$) видно, что

$$\lim_{k\to\infty}\lambda_i^k=\lambda_i^*,\quad i\in I,$$

или, что то же самое,

$$\lim_{k\to\infty}r_k\varphi_i^+(x_k)=\lambda_i^*,\quad i\in I.$$

Таким образом, произведения параметров штрафа на невязки ограничений исходной задачи в точке минимума по x штрафной функции $F(x, r_k)$ могут служить оценками множителей Лагранжа λ_i^*. Для других гладких штрафных функций связь между невязками ограничений и λ_i^* будет иной, но она всегда есть

Равенство (2.13) дает возможность достаточно точно оценить разность

$$f(x^*)-f(x_k).$$

Чтобы получить такую оценку, перепишем (2.13) так:

$$\frac{\partial L}{\partial x}(x_k,\ \lambda^k)=0, \tag{2.15}$$

где $L(x,\ \lambda)$ — функция Лагранжа вида

$$L(x,\ \lambda)=f(x)+\sum_{i=1}^{m}\lambda_i\varphi_i(x).$$

Поскольку $\varphi_i(x^*)=0$ для $i\in I$ и $\lambda_i^k=0$ для $i\notin I$, ясно, что

$$f(x^*)=L(x^*,\ \lambda^k)$$

и, соответственно,

$$f(x^*) = L(x_k, \lambda^k) + \left(\frac{\partial L}{\partial x}(x_k, \lambda^k),\ x^* - x_k\right) + O(\|x_k - x^*\|^2) =$$
$$= f(x_k) + \sum_{i=1}^{m} \lambda_i^k \varphi_i(x_k) + O(\|x_k - x^*\|^2).$$

Отсюда, вспоминая определение λ_i^k, получим

$$f(x^*) - f(x_k) = r_k \sum_{i=1}^{m} (\varphi_i^+(x_k))^2 + O(\|x_k - x^*\|^2). \quad (2.16)$$

При приближении x_k к точке x^* первое слагаемое равномерно по k имеет порядок нормы $\|x_k - x^*\|$, а второе — порядок квадрата этой нормы. Поэтому на основании (2.16) можно предложить при больших r_k следующую оценку:

$$f(x^*) - f(x_k) \simeq r_k \sum_{i=1}^{m} (\varphi_i^+(x_k))^2.$$

Когда задача (2.11) линейна, это приближенное равенство становится точным. Если же (2.11) — задача выпуклого программирования, функция $L(x, \lambda^k)$ выпукла по x и, следовательно, выполнено неравенство

$$L(x^*, \lambda^k) - L(x_k, \lambda^k) \geqslant \left(\frac{\partial L}{\partial x}(x_k, \lambda^k),\ x_k - x^*\right) = 0,$$

т. е.

$$f(x^*) \geqslant L(x_k, \lambda^k) = f(x_k) + r_k \sum_{i=1}^{m} (\varphi_i^+(x_k))^2.$$

Здесь опять следует отметить, что аналогичные оценки разности $f(x^*) - f(x_k)$ могут быть получены и для других гладких штрафных функций.

Рассмотрим теперь характер поведения функций $F(x, r_k)$ вблизи точек x_k, сохранив предположение о линейной независимости векторов $\varphi_i'(x^*)$, $i \in I$, и считая дополнительно, что $\lambda_i^* > 0$, $i \in I$, и в наборе I меньше, чем n (n — размерность вектора x) индексов (последнее характерно для нелинейных задач). При достаточно больших k градиент функции $F(x, r_k)$ в окрестности x_k вычисляется по формуле

$$\frac{\partial F}{\partial x}(x, r_k) = f'(x) + r_k \sum_{i \in I} \varphi_i^+(x) \varphi_i'(x),$$

а матрица вторых производных

$$H_k = \left\| \frac{\partial^2 F}{\partial x^i \partial x^j}(x_k,\ r_k) \right\|,$$

которые существуют, так как в силу сделанных предположений $\varphi_i'(x_k) > 0$, $i \in I$, равна

$$H_k = f''(x_k) + \sum_{i \in I} \lambda_i^k \varphi_i''(x_k) + r_k A_k,$$

где

$$A_k = \sum_{i \in I} \varphi_i'(x_k)\, \varphi_i'^{\,T}(x_k).$$

Как будет выглядеть функция $F(x,\ r_k)$ вблизи x_k, зависит от обусловленности этой матрицы, т. е. от отношения ее максимального и минимального собственных чисел. Чем больше это отношение, тем более овражной будет функция $F(x,\ r_k)$ и тем труднее искать ее минимум.

Обозначим максимальное и минимальное собственные числа матрицы H_k через M и m. Тогда для любого $y \in E_n$ выполнены неравенства

$$m\|y\|^2 \leqslant (y,\ H_k y) \leqslant M\|y\|^2, \tag{2.17}$$

причем $m \geqslant 0$, так как x_k — точка минимума $F(x,\ r_k)$ по x. Эти неравенства, в частности, справедливы для вектора y_1 такого, что $\|y_1\| = 1$ и

$$(\varphi_i'(x_k),\ y_1) = 0, \quad i \in I. \tag{2.18}$$

(Последний существует, поскольку в списке I меньше n индексов, т. е. система уравнений (2.18) недоопределена.) С другой стороны, для выбранного y_1

$$(y_1,\ H_k y_1) = (y_1,\ W(x_k,\ \lambda^k)\, y_1) + r_k (y_1,\ A_k y_1) = (y_1,\ W(x_k,\ \lambda^k)\, y_1), \tag{2.19}$$

где

$$W(x_k,\ \lambda^k) = f''(x_k) + \sum_{i \in I} \lambda_i^k \varphi_i''(x_k),$$

и, поскольку матрица $W(x_k,\ \lambda^k)$ при $k \to \infty$ сходится к некоторому пределу, из (2.19) следует оценка

$$(y_1,\ H_k y_1) \leqslant c_1. \tag{2.20}$$

Здесь c_1 — не зависящая от k положительная константа. Сравнив (2.20) и (2.17) (с учетом равенства $\|y_1\| = 1$),

получим, что

$$m \leqslant c_1. \tag{2.21}$$

Пусть теперь y_2 — решение системы

$$(\varphi_i'(x_k),\ y_2) = 1, \quad i \in I,$$

причем мы вправе считать, что $\|y_2\|^2 \leqslant c_2$, где c_2 — константа, не связанная с номером k. Тогда

$$(y_2,\ H_k y_2) = (y_2,\ W(x_k,\ \lambda^k)\, y_2) + r_k t \geqslant$$

$$\geqslant c_3 \|y_2\|^2 + \frac{r_k t}{c_2} \|y_2\|^2. \tag{2.22}$$

Здесь t — количество индексов в наборе I, а c_3 — некоторое число, не зависящее от k. Сопоставление (2.22) и правого неравенства в (2.17) показывает, что

$$M \geqslant c_3 + \frac{r_k t}{c_2}.$$

В свою очередь, отсюда и из (2.21) видно, что

$$\frac{M}{m} \geqslant \frac{c_3 + r_k t/c_2}{c_1},$$

т. е., вообще говоря, матрица H_k обусловлена тем хуже, чем больше величина r_k.

Таким образом, с увеличением параметра штрафа r_k задача безусловной минимизации функции $F(x,\ r_k)$ по x усложняется из-за того, что последняя приобретает все более выраженную овражную структуру. Кроме того, при больших r_k сильно возрастает роль, которую играют в вычислительном процессе поиска минимума $F(x,\ r_k)$ по x ошибки округления машины: близкие к нулю величины $\varphi_i(x)$ обычно подсчитываются с относительно низкой точностью и умножение на большое число r_k соответствующих ошибок может привести к тому, что в вычисленном вблизи x_k градиенте функции $F(x,\ r_k)$ не будет ни одного верного знака. В силу указанного обстоятельства найти минимум по x функции $F(x,\ r_k)$ при больших r_k с высокой точностью оказывается практически невозможным. Соответственно, получить очень точное решение задачи с ограничениями методом с квадратичным штрафом тоже нельзя. Сказанное относится и к остальным традиционным методам гладких штрафных функций. Все они пригодны только для поиска весьма приближенных решений. Поэтому раз-

рабатывались методы со штрафными функциями иного типа. Два из них представлены ниже.

7. Метод с оценкой критерия. Рассмотрим задачу

$$\begin{aligned} &\min f(x), \\ &\varphi_i(x) \leqslant 0, \quad i = 1, \ldots, m, \end{aligned} \tag{2.23}$$

и пусть x^* — ее решение. Ясно, что при этом система неравенств

$$\begin{aligned} &f(x) \leqslant v, \\ &\varphi_i(x) \leqslant 0, \quad i = 1, \ldots, m, \end{aligned}$$

будет неразрешимой, какова бы ни была величина $v < f(x^*)$, и, следовательно, для всех $v < f(x^*)$ минимум (если он существует) функции

$$\Psi(x, v) = ((f(x) - v)^+)^2 + \sum_{i=1}^{m} (\varphi_i^+(x))^2$$

по $x \in E_n$ положителен. Если же взять $v \geqslant f(x^*)$, этот минимум окажется равным нулю. Значит, задачу (2.23) можно сформулировать как задачу поиска наименьшего корня $v^* = f(x^*)$ уравнения

$$h(v) = \min_{x \in E_n} \Psi(x, v) = 0$$

и точки x^*, доставляющей безусловный минимум по x функции $\Psi(x, v^*)$.

Реализацию предлагаемой схемы построить совсем несложно. В самом деле, пусть $v_k < v^*$ — очередное приближение искомого корня. Тогда

$$\Psi(x_k, v_k) = \min_{x \in E_n} \Psi(x, v_k) \leqslant \Psi(x^*, v_k) =$$

$$= ((f(x^*) - v_k)^+)^2 + \sum_{i=1}^{m} (\varphi_i^+(x^*))^2 = (v^* - v_k)^2,$$

т. е.

$$v^* - v_k \geqslant \sqrt{\Psi(x_k, v_k)},$$

и в качестве нового приближения v_{k+1} можно взять

$$\begin{aligned} &v_{k+1} = v_k + \sqrt{\Psi(x_k, v_k)} > v_k, \\ &v_{k+1} \leqslant v^*. \end{aligned} \tag{2.24}$$

Получающаяся в результате монотонно возрастающая последовательность $\{v_k\}$ ограничена сверху. Поэтому

$$\lim_{k\to\infty} \Psi(x_k, v_k) = 0,$$

т. е.

$$\lim_{k\to\infty} \varphi_i^+(x_k) = 0, \quad i = 1, \ldots, m,$$

$$\lim_{k\to\infty} (f(x_k) - v_k)^+ = 0$$

и, кроме того,

$$\lim_{k\to\infty} v_k \leqslant f(x^*).$$

Из этих соотношений при весьма слабых предположениях относительно характера функций $f(x)$, $\varphi_i(x)$, $i = 1, \ldots, m$, следует сходимость v_k к v^*, а x_k — к x^*.

Таким образом, мы получили алгоритм решения задачи (2.23), в котором:

а) выбирается начальная заниженная оценка величины $f(x^*)$, т. е. число $v_0 < f(x^*)$;

б) при $k = 0, 1, 2, \ldots$ решается задача безусловной минимизации по x функции $\Psi(x, v_k)$, в результате чего определяется очередное приближение x_k точки x^* и очередная оценка $v_{k+1} = v_k + \sqrt{\Psi(x_k, v_k)}$ величины $f(x^*)$.

Данный алгоритм значительно меньше подвержен влиянию ошибок округления, чем традиционные методы штрафных функций, но, в отличие от них, не обеспечивает сходимости к локальному решению в случае, когда на каждой итерации определяются локальные безусловные минимумы. Что же касается задач выпуклого программирования, для которых функция $\Psi(x, v)$ выпукла по x и, следовательно, имеет только глобальные минимумы, то их можно решать, используя более эффективную формулу пересчета v_k. Эту формулу мы сейчас выведем.

Предположим, что функции $f(x)$, $\varphi_i(x)$, $i = 1, \ldots, m$, в задаче (2.23) выпуклы и дифференцируемы. Как и прежде, обозначим через x^* решение этой задачи, и пусть v_k — очередная заниженная оценка величины $f(x^*)$, т. е. $v_k < f(x^*)$. Тогда, если x_k — точка безусловного минимума по $x \in E_n$

функции $\Psi(x, v_k)$, должно быть

$$\frac{\partial \Psi}{\partial x}(x_k, v_k) = 2(f(x_k) - v_k)^+ f'(x_k) + \\ + 2\sum_{i=1}^{m} \varphi_i^+(x_k)\varphi_i'(x_k) = 0, \qquad (2.25)$$

$$\Psi(x_k, v_k) > 0.$$

При этом обеспечено неравенство $f(x_k) > v_k$, поскольку противное в силу (2.25) означало бы, что x_k — точка минимума суммы $\sum_{i=1}^{m} (\varphi_i^+(x))^2$ и этот минимум положителен, т. е. у исходной задачи нет решения. Значит, равенство (2.25) можно поделить на $(f(x_k) - v_k)$, что дает

$$f'(x_k) + \sum_{i=1}^{m} \lambda_i^k \varphi_i'(x_k) = 0, \qquad (2.26)$$

где

$$\lambda_i^k = \varphi_i^+(x_k)/(f(x_k) - v_k) \geqslant 0.$$

В свою очередь, (2.26) означает, что выпуклая функция Лагранжа

$$L(x, \lambda^k) = f(x) + \sum_{i=1}^{m} \lambda_i^k \varphi_i(x)$$

достигает в точке x_k своего минимума. Поэтому

$$L(x_k, \lambda^k) \leqslant L(x^*, \lambda^k),$$

а так как все λ_i^k неотрицательны и все $\varphi_i(x^*)$ неположительны, кроме того, имеем

$$L(x^*, \lambda^k) \leqslant f(x^*).$$

Следовательно,

$$L(x_k, \lambda^k) \leqslant f(x^*),$$

и мы можем взять

$$v_{k+1} = L(x_k, \lambda^k)$$

в качестве очередной незавышенной оценки величины $f(x^*)$.

причем

$$v_{k+1}=f(x_k)+\sum_{i=1}^{m}\lambda_i^k\varphi_i(x_k)=f(x_k)+\frac{\sum_{i=1}^{m}(\varphi_i^+(x_k))^2}{f(x_k)-v_k}=$$

$$=v_k+\frac{\Psi(x_k,\ v_k)}{f(x_k)-v_k}>v_k,$$

$$v_{k+1}-v_k>\sqrt{\Psi(x_k,\ v_k)}.$$

Итак, для задач выпуклого программирования более эффективен, чем рассмотренный ранее, алгоритм, в котором

а) выбирается начальная оценка v_0 величины $f(x^*)$;

б) на k-й итерации решается задача безусловной минимизации по x функции $\Psi(x,\ v_k)$, в результате чего определяется очередное приближение x_k точки x^*;

в) если $\Psi(x_k,\ v_k)>0$, вычисляется очередная оценка $v_{k+1}=v_k+\Psi(x_k,\ v_k)/(f(x_k)-v_k)$ величины $f(x^*)$, после чего выполняется следующая итерация.

Если функции $f(x)$, $\varphi_i(x)$, $i=1,\ \ldots,\ m$, линейны, этот алгоритм дает оптимальную точку x^* за конечное число шагов. В противном случае искомое решение x^* будет получено в пределе, причем при естественных предположениях скорость сходимости x_k к x^* сверхлинейна. Алгоритм мало чувствителен к ошибкам округления, но, к сожалению, не годится для решения невыпуклых задач. К тому же функции $\Psi(x,\ v_k)$ при v_k, близких к $f(x^*)$, овражны в окрестности точек своих минимумов, так что один из двух основных недостатков традиционных штрафных функций здесь сохраняется. Метод, который лишен этого недостатка и пригоден для решения невыпуклых задач, описан в следующем пункте.

8. Метод с модифицированной функцией Лагранжа. Рассмотрим задачу

$$\min f(x),\qquad \varphi_i(x)\leqslant 0,\quad i=1,\ \ldots,\ m, \tag{2.27}$$

где $f(x)$, $\varphi_i(x)$ — дифференцируемые функции, причем для начала будем считать их выпуклыми. Введем для этой задачи так называемую модифицированную функцию Лагранжа

$$M(x,\ \lambda)=f(x)+\frac{1}{2}\sum_{i=1}^{m}((\varphi_i(x)+\lambda_i)^+)^2,$$

и пусть $x(\lambda)$ — точка ее минимума по $x \in E_n$ при некоторых фиксированных $\lambda_i \geqslant 0$, $i = 1, \ldots, m$. Тогда

$$\frac{\partial M}{\partial x}(x(\lambda), \lambda) = f'(x(\lambda)) + \sum_{i=1}^{m} (\varphi_i(x(\lambda)) + \lambda_i)^+ \varphi_i'(x(\lambda)) = 0.$$

Обозначив через $\bar{\lambda}$ вектор с координатами

$$\bar{\lambda}_i = (\varphi_i(x(\lambda)) + \lambda_i)^+ \geqslant 0,$$

это равенство можно записать так:

$$\frac{\partial L}{\partial x}(x(\lambda), \bar{\lambda}) = 0, \tag{2.28}$$

где

$$L(x, \bar{\lambda}) = f(x) + \sum_{i=1}^{m} \bar{\lambda}_i \varphi_i(x).$$

Поскольку функция $L(x, \bar{\lambda})$ выпукла по x, (2.28) означает, что $x(\lambda)$ — точка ее безусловного минимума Кроме того, ясно, что $x(\lambda)$ — решение «возмущенной» задачи

$$\begin{aligned} &\min f(x), \\ &\varphi_i(x) \leqslant 0, \qquad i \in I_1 = \{i: \bar{\lambda}_i = 0\}, \\ &\varphi_i(x) \leqslant \varphi_i(x(\lambda)), \quad i \in I_2 = \{i: \bar{\lambda}_i > 0\}. \end{aligned} \tag{2.29}$$

Действительно, как легко видеть, $x(\lambda)$ — допустимая точка этой задачи, причем

а) ее функция Лагранжа

$$L_M(x, \bar{\lambda}) = f(x) + \sum_{i \in I_1} \lambda_i \varphi_i(x) + \sum_{i \in I_2} \bar{\lambda}_i (\varphi_i(x) - \varphi_i(x(\lambda)))$$

отличается от $L(x, \lambda)$ слагаемым, не зависящим от x, и, следовательно, достигает в $x(\lambda)$ минимума по x;

б) по определению, все $\bar{\lambda}_i$ неотрицательны и

$$L_M(x(\lambda), \bar{\lambda}) = f(x(\lambda)).$$

Но этих двух условий достаточно, чтобы гарантировать оптимальность $x(\lambda)$ для задачи (2.27).

Теперь понятно, как в принципе можно решить задачу (2.27), используя функцию $M(x, \lambda)$: надо подобрать $\lambda_i \geqslant 0$, $i = 1, \ldots, m$, так, чтобы, отыскав безусловный

минимум $M(x, \lambda)$ по x, получить точку $x(\lambda)$, в которой $\varphi_i(x(\lambda)) = 0$ для всех i таких, что $\varphi_i(x(\lambda)) + \lambda_i > 0$, а попросту говоря — для всех i, при которых $\lambda_i > 0$. Тогда в силу сказанного ранее $x(\lambda)$ будет искомым решением. Соответствующие λ_i существуют — это множители Лагранжа, отвечающие оптимуму задачи (2.27). Конечно, пока этот оптимум не найден, неоткуда взять и связанные с ним множители Лагранжа. Поэтому в действительности однократной безусловной минимизацией $M(x, \lambda)$ по x не обойтись. Нетрудно, однако, построить алгоритм с последовательной безусловной минимизацией функции $M(x, \lambda)$, который даст решение задачи (2.27) в пределе.

Сохраняя все прежние обозначения, рассмотрим функцию $M(x, \lambda)$. Если $x(\lambda)$ — точка ее минимума по $x \in E_n$, должно выполняться неравенство

$$M(x(\lambda), \bar{\lambda}) \leqslant M(x(\lambda), \bar{\lambda}). \tag{2.30}$$

При этом

$$M(x(\lambda), \bar{\lambda}) = f(x(\lambda)) + \frac{1}{2}\sum_{i=1}^{m}((\varphi_i(x(\lambda)) + \bar{\lambda}_i)^+)^2 =$$

$$= f(x(\lambda)) + \frac{1}{2}\sum_{i\in I_1}((\varphi_i(x(\lambda)) + \lambda_i)^+)^2 + \frac{1}{2}\sum_{i\in I_2}((\varphi_i(x(\lambda)) + \lambda_i)^+)^2.$$

Поскольку $(\varphi_i(x(\lambda)) + \lambda_i)^+ = \bar{\lambda}_i = 0$ и, значит, $\varphi_i(x(\lambda)) \leqslant -\lambda_i \leqslant 0$ для $i \in I_1$, первая сумма в правой части этого равенства равна нулю, т. е.

$$M(x(\lambda), \lambda) = f(x(\lambda)) + \frac{1}{2}\sum_{i\in I_2}((\varphi_i(x(\lambda)) + \bar{\lambda}_i)^+)^2 \leqslant$$

$$\leqslant f(x(\lambda)) + \frac{1}{2}\sum_{i\in I_2}(\varphi_i(x(\lambda)) + \bar{\lambda}_i)^2 =$$

$$= f(x(\lambda)) + \sum_{i\in I_2}\bar{\lambda}_i\varphi_i(x(\lambda)) + \frac{1}{2}\sum_{i\in I_2}\varphi_i^2(x(\lambda)) + \frac{1}{2}\sum_{i\in I_2}\bar{\lambda}_i^2 =$$

$$= L(x(\lambda), \lambda) + \frac{1}{2}\sum_{i\in I_2}\varphi_i^2(x(\lambda)) + \frac{1}{2}\sum_{i\in I_2}\lambda_i^2. \tag{2.31}$$

Далее,

$$M(x(\bar\lambda),\ \bar\lambda)=$$

$$=f(x(\bar\lambda))+\frac{1}{2}\sum_{i\in I_1}((\varphi_i(x(\bar\lambda))+\lambda_i)^+)^2+\frac{1}{2}\sum_{i\in I_2}((\varphi_i(x(\bar\lambda))+\bar\lambda_i)^+)^2=$$

$$=f(x(\bar\lambda))+\frac{1}{2}\sum_{i\in I_1}(\varphi_i^+(x(\lambda)))^2+\frac{1}{2}\sum_{\substack{i\in I_2\\ \varphi_i(x(\bar\lambda))+\bar\lambda_i>0}}(\varphi_i(x(\bar\lambda))+\lambda_i)^2+$$

$$+\frac{1}{2}\sum_{\substack{i\in I_2\\ \varphi_i(x(\bar\lambda))+\bar\lambda_i\leqslant 0}}(\lambda_i\varphi_i(x(\bar\lambda))-\bar\lambda_i\varphi_i(x(\bar\lambda)))=$$

$$=L(x(\bar\lambda),\ \bar\lambda)+\frac{1}{2}\sum_{\varphi_i(x(\bar\lambda))+\bar\lambda_i>0}\varphi_i^2(x(\bar\lambda))+$$

$$+\sum_{\substack{i\in I_2\\ \varphi_i(x(\bar\lambda))+\bar\lambda_i>0}}\bar\lambda_i^2-\sum_{\substack{i\in I_2\\ \varphi_i(x(\bar\lambda))+\bar\lambda_i\leqslant 0}}\lambda_i\varphi_i(x(\bar\lambda))\geqslant$$

$$\geqslant L(x(\bar\lambda),\ \bar\lambda)+\frac{1}{2}\sum_{\varphi_i(x(\bar\lambda))+\bar\lambda_i>0}\varphi_i^2(x(\bar\lambda))+\frac{1}{2}\sum_{i\in I_2}\lambda_i^2. \qquad (2.32)$$

Сопоставляя (2.30), (2.31), (2.32), получим

$$L(x(\lambda),\ \lambda)+\frac{1}{2}\sum_{\varphi_i(x(\bar\lambda))+\bar\lambda_i>0}\varphi_i^2(x(\lambda))\leqslant$$

$$\leqslant L(x(\lambda),\ \lambda)+\frac{1}{2}\sum_{i\in I_2}\varphi_i^2(x(\lambda))$$

или, что то же самое,

$$\sum_{\varphi_i(x(\lambda))+\bar\lambda_i>0}\varphi_i^2(x(\bar\lambda))\leqslant$$

$$\leqslant 2(L(x(\lambda),\ \lambda)-L(x(\bar\lambda),\ \bar\lambda))+\sum_{\varphi_i(x(\lambda))+\lambda_i>0}\varphi_i^2(x(\lambda)).$$

Поскольку $x(\lambda)$ — точка минимума функции $L(x,\ \bar\lambda)$ по x, отсюда следует, что

$$\sum_{\varphi_i(x(\bar\lambda))+\lambda_i>0}\varphi_i^2(x(\lambda))\leqslant\sum_{\varphi_i(x(\lambda))+\lambda_i>0}\varphi_i^2(x(\lambda)).$$

Проводя выкладки более аккуратно, мы могли бы получить здесь и строгое неравенство, а это означает, что точка $x(\lambda)$ является решением «возмущенной» задачи типа (2.29), в которой сумма квадратов «возмущений» меньше, чем в самой задаче (2.29) с решением $x(\lambda)$. В этом смысле $x(\lambda)$ «ближе» к решению x^* исходной задачи (2.27), чем $x(\lambda)$. Таким образом, можно предложить следующий алгоритм поиска x^*.

а) выбирается вектор $\lambda^0 \geqslant 0$ и находится точка x_0 безусловного минимума функции $M(x, \lambda^0)$ по x;

б) для $k = 1, 2, \ldots$ вычисляются $\lambda_i^k = (\varphi_i(x_{k-1}) + \lambda_i^{k-1})^+$ и определяется точка x_k минимума функции $M(x, \lambda^k)$ по $x \in E_n$.

Если исходная задача линейна, решение будет найдено за конечное число шагов. В нелинейном случае решение (в естественных предположениях относительно свойств задачи) получается как предел точек x_k.

Все сказанное относилось к задачам выпуклого программирования Если же задача (2.27) невыпукла, ее можно решать с применением функции

$$M(x, \lambda, r) = f(x) + \frac{r}{2} \sum_{i=1}^{m} \left(\left(\varphi_i(x) + \frac{\lambda_i}{r} \right)^+ \right)^2$$

и алгоритм будет выглядеть так.

а) выбираются вектор $\lambda^0 \geqslant 0$ и число $r_0 > 0$, определяется точка x_0 минимума (вообще говоря, локального) функции $M(x, \lambda^0, r_0)$ по $x \in E_n$, r_1 полагается равным r_0 и вычисляются $\lambda_i^1 = (r_0 \varphi_i(x_0) + \lambda_i^0)^+$;

б) определяется точка x_k минимума функции

$$M(x, \lambda^k, r_k) \quad \text{по} \quad x \in E_n;$$

в) проверяется соблюдение неравенства

$$\sum_{\varphi_i(x_k) + \frac{\lambda_i^k}{r_k} > 0} \varphi_i^2(x_k) < \sum_{\varphi_i(x_{k-1}) + \frac{\lambda_i^{k-1}}{r_{k-1}} > 0} \varphi_i^2(x_{k-1});$$

если оно выполнено, r_{k+1} полагается равным r_k и $\lambda_i^{k+1} = (r_k \varphi_i(x_k) + \lambda_i^k)^+$, а в противном случае $\lambda^{k+1} = \lambda^k$ и $r_{k+1} = q r_k$, где $q > 1$ — константа, после этого повторяется п. б) и так далее.

Данный алгоритм обладает теми же свойствами сходимости, что и метод с квадратичной функцией штрафа При этом, хотя параметр r_k может расти, он крайне редко достигает очень больших значений. Поэтому проблем, связанных с овражностью, здесь не возникает. Вообще, следует сказать, что представленный алгоритм считается одним из наиболее эффективных среди универсальных методов решения нелинейных задач На нем мы и закончим изучение таких методов В следующей главе речь пойдет о некоторых специальных алгоритмах.

Глава VI

МЕТОДЫ ОПТИМИЗАЦИИ, ОСНОВАННЫЕ НА ПОСЛЕДОВАТЕЛЬНОМ АНАЛИЗЕ ВАРИАНТОВ

Введение

Вспомнив рассмотренные ранее методы решения задач с ограничениями, читатель наверняка согласится с утверждением о том, что чем больше ограничений наложено на область изменения аргумента минимизируемой функции, тем сложнее становится задача оптимизации — тем труднее найти ее решение с помощью этих методов. С другой стороны (и это интуитивно кажется почти очевидным), задача отыскания значения x_*, доставляющего минимум функции $f(x)$, должна быть тем проще, чем уже допустимая область изменения аргумента x. Если ограничения настолько жесткие, что это множество состоит из нескольких точек и эти точки легко отыскать, задача сводится к простому перебору нескольких чисел. Можно привести и другой пример: допустим, что требуется найти кратчайший путь между двумя точками, соединенными узким коридором — допустимой областью движения. Тогда собственно оптимизационная задача снова тривиальна: просто надо следовать заданным коридором — любой из путей в нем будет практически оптимален. Однако если мы захотим для поиска решений данных задач использовать методы, изложенные выше, то либо вообще ничего не получится (в случае, когда допустимое множество состоит из нескольких точек), либо процедура поиска будет чрезвычайно громоздкой. Поэтому решение задач со сложными ограничениями требует создания качественно иных методов.

В настоящей главе мы рассмотрим алгоритмы, построенные по схеме последовательного анализа вариантов — с использованием процедур, имеющих своей целью на основании косвенных оценок отбросить все те допустимые решения, среди которых не может быть оптимального (или «трудно ожидать», что оно там содержится). По мере выполнения этих процедур происходит постепенное сжатие множества конкурентоспособных вариантов. В конце кон-

цов остается один или несколько, которые уже непосредственно сравниваются между собой.

Принцип последовательного исключения вариантов, отбора среди них наиболее предпочтительных отвечает тому естественному ходу человеческой мысли, который был выработан эволюцией. Но превратить этот общий подход в систему формальных процедур, в математические теории, позволяющие строить эффективные алгоритмы, очень трудно. Тем не менее, многое в этом направлении уже сделано.

В математике первые идеи подобного рода были высказаны еще А. А. Марковым. В послевоенные годы проблематика последовательного анализа вариантов подробно обсуждалась в работах американского математика Вальда. В США эти исследования были продолжены Р. Айзексом и Р. Беллманом. В результате последний пришел к созданию динамического программирования. В СССР идеи А. А. Маркова и Вальда развивались В. С. Михалевичем и его учениками, создавшими общий формализм последовательного анализа вариантов.

Методы оптимизации, основанные на идее последовательного анализа вариантов, в большой степени используют природу изучаемых задач. Поэтому, в отличие от предыдущих глав, мы здесь не будем излагать общую теорию вопроса (и, в частности, схему формализации В. С. Михалевича), а ограничимся анализом некоторых конкретных задач, достаточно полно иллюстрирующих принципы и способы конструирования вычислительных алгоритмов.

Сначала мы рассмотрим так называемые аддитивные задачи, затем будет изложен общий метод динамического программирования. В последнем параграфе этой главы мы рассмотрим ряд задач, для которых метод динамического программирования неприменим, но для решения которых, тем не менее, с помощью последовательного анализа вариантов удается построить удовлетворительные численные алгоритмы.

§ 1. Аддитивные задачи

1. Один тривиальный пример. Методы последовательного анализа вариантов не представляют собой каких-либо стандартных процедур. Содержание этих методов состоит в построении системы правил отбраковки тех множеств

вариантов, среди которых либо заведомо, либо «предположительно» не могут содержаться оптимальные решения. Разумеется, эти правила всегда существенным образом используют природу изучаемых задач. Рассмотрим сначала один пример, демонстрирующий характер рассуждений. Пусть требуется определить минимум функции $f(x)$ вида

$$f(x) = \sum_{i=1}^{N} f_i(x^i), \tag{1.1}$$

где x^i — скалярные компоненты вектора x, причем выбор этого вектора стеснен ограничениями

$$x^i \in G_i, \quad i = 1, \ldots, N. \tag{1.2}$$

Очевидно, что эта задача оптимизации функции N переменных сводится к решению N задач оптимизации функций одного аргумента: компоненты любого из векторов $\tilde{x}$, реализующих минимум функции (1.1) при ограничениях (1.2), будут решениями задач

$$\min_{x^i \in G_i} f_i(x^i).$$

Полученный результат совершенно тривиален. Тем не менее, докажем его.

Обозначим через Ω множество всех допустимых векторов (допустимое множество вариантов) задачи (1.1) — (1.2). Пусть ω_1 — множество всех тех векторов x из Ω, у которых первая компонента не доставляет минимума функции $f_1(x^1)$ на множестве G_1. Очевидно, что искомый «вариант» не может содержаться в ω_1. В самом деле, какой бы «вариант»

$$x_* = \{x_*^1, \ldots, x_*^N\}^T \in \omega_1, \quad f_1(x_*^1) \neq f_1(\tilde{x}^1)$$

мы ни взяли, его сразу можно улучшить. Значение $f(x_0)$, где $x_0 = \{\tilde{x}^1, x_*^2, \ldots, x_*^N\}^T$, будет меньше, чем $f(x_*)$.

Таким образом, оптимальный «вариант» содержится среди векторов, принадлежащих множеству

$$\Omega_1 = \Omega \setminus \omega_1.$$

Мы произвели сужение множества вариантов — от мно-

жества Ω перешли к множеству Ω_1. Повторяя рассуждения, в конце концов получим множество

$$\Omega_N = \Omega \setminus \omega_1 \setminus \omega_2 \setminus \ldots \setminus \omega_N,$$

содержащее векторы

$$\tilde{x} = \{\tilde{x}^1, \ldots, \tilde{x}^N\},$$

каждый из которых является решением задачи.

Изложенную процедуру анализа удобно представить в виде графической схемы (рис. 1.1). На каждом шаге

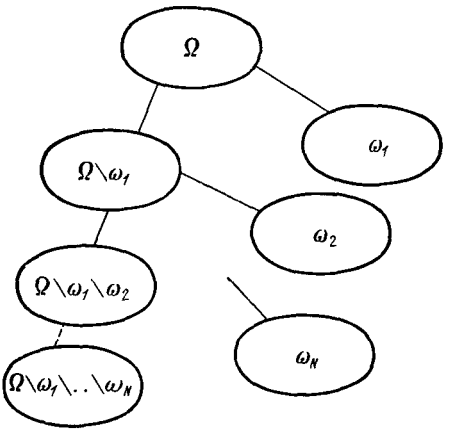


Рис. 1.1.

мы анализировали два множества: ω_i и $\Omega \setminus \omega_1 \setminus \ldots \setminus \omega_i$, выбирая затем одно из них, для чего использовали некоторое «правило сравнения». Последнее целиком определяется содержанием задачи и в рассматриваемом случае тривиально.

Примечания. 1) Структура ω_i — множества вариантов, которое отбрасывалось, и характер рассуждений существенным образом определялись не только видом функции $f(x)$, но и ограничениями (1.2). В самом деле, заменим, например, их следующими:

$$\{x^i,\ x^{i+1}\}^T \in G_i,\quad i = 1, \ldots, N-1. \qquad (1.3)$$

Последнее означает, что проекция допустимой точки на

плоскость координатных осей с номерами i и $i+1$ должна принадлежать заданному множеству G_i. Очевидно, что в этом случае полученный выше результат перестает быть справедливым. Правда, для отыскания минимума функции (1.1) при ограничениях (1.3) тоже можно предложить некоторую процедуру выделения множеств ω_l неконкурентоспособных вариантов. Но эти множества будут совершенно иной природы.

2) Описанный процесс можно интерпретировать как построение некоторого дерева, ветви которого последовательно отсекаются.

2. Аддитивные функции и задачи.

Определение 1.1. Функцию $f(x_0, \ldots, x_N)$ назовем *аддитивной*, если она имеет вид

$$f(x_0, \ldots, x_N) = \sum_{i=0}^{N-1} f_i(x_i, x_{i+1}). \tag{1.4}$$

Здесь $x_0, x_1, \ldots, x_N$ — векторы размерности $n_0, n_1, \ldots, n_N$, соответственно.

Функция, которая рассматривалась в предыдущем пункте, была частным случаем аддитивной.

Если ограничения имеют вид

$$x_i \in G_i, \quad i=0, \ldots, N, \tag{1.5}$$

то задача отыскания минимума функции (1 4) также называется *аддитивной*.

Аддитивные задачи встречаются во многих областях практической деятельности человека. Они допускают наглядную геометрическую интерпретацию, которая будет использоваться в течение всего последующего изложения. Состоит эта интерпретация в следующем. Предполагая, что размерности всех векторов x_i равны n, введем в рассмотрение $(n+1)$-мерное пространство пар $\{x, t\}$, в котором построим гиперплоскости Σ, отвечающие значениям $t=i$, $i=0, 1, \ldots, N$. Тогда любую совокупность векторов $\tilde{x}_0, \ldots, \tilde{x}_N$ мы можем отождествить с ломаной, проходящей в нашем $(n+1)$-мерном пространстве через точки $\{\tilde{x}_0, 0\} \in \Sigma_0, \ldots, \{\tilde{x}_N, N\} \in \Sigma_N$ (рис. 1.2). Будем говорить, что она допустима, если $\tilde{x}_i \in G_i$ при всех $i=0, \ldots, N$. (На рис. 1.2 область, в которой могут располагаться допустимые ломаные, не заштрихована.) Ее общую длину

определим как сумму длин ее звеньев, а длину звена, связывающего точки $\{\tilde{x}_i, i\}$, $\{\tilde{x}_{i+1}, i+1\}$, будем измерять величиной $f_i(\tilde{x}_i, \tilde{x}_{i+1})$. В этом случае длина всей ломаной есть $f(\tilde{x}_0, \ldots, \tilde{x}_N)$.

Теперь исходную задачу можно сформулировать так: среди всех допустимых ломаных, соединяющих гиперплоскости Σ_0 и Σ_N, найти ту, длина которой минимальна.

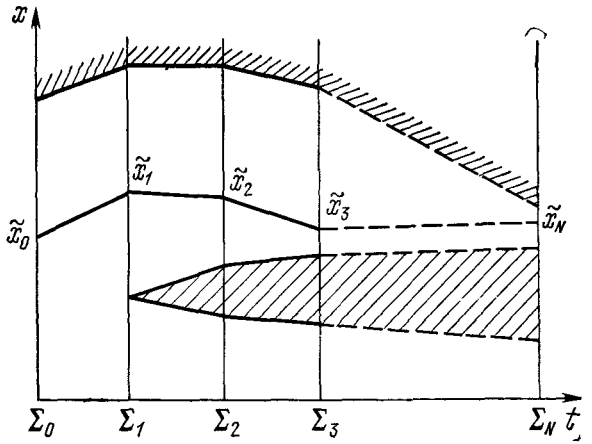


Рис 1.2.

В дальнейшем векторы x_i будем иногда отождествлять с точками $\{x_i, i\} \in \Sigma_i$. Именно в этом смысле следует понимать выражения: «x_i принадлежит Σ_i» или «ломаная, проходящая через точку x_i».

Примечание. Важный частный случай рассматриваемой задачи — тот, в котором концы ломаной фиксированы Формально это означает, что в множествах G_0 и G_N есть только по одной точке.

3. Общая схема последовательного анализа для аддитивных задач. Итак, среди всевозможных ломаных, соединяющих гиперплоскости Σ_0 и Σ_N, мы будем разыскивать ломаную наименьшей длины. Множество всех допустимых, т. е удовлетворяющих ограничению (1.5) ломаных, обозначим через Ω.

Рассмотрим теперь произвольную точку $x_1 \in \Sigma_1$. Кратчайший из отрезков, соединяющих эту точку с гиперплоскостью Σ_0, имеет длину

$$l(x_1) = \min_{x_0 \in G_0} f_0(x_0, x_1). \tag{1.6}$$

Операция (1.6) каждой точке $x_1 \in G_1$ ставит в соответствие число $l(x_1)$.

Рассмотрим теперь функцию $f(x_0, x_1, x_2, \ldots, x_N)$. Так как

$$\min_{x_0 \in G_0} f(x_0, x_1, \ldots, x_N) = l(x_1) + \sum_{i=1}^{N-1} f_i(x_i, x_{i+1}),$$

то любая ломаная, не содержащая отрезка $l(x_1)$, не может быть претендентом на то, чтобы считаться решением задачи минимизации. Эти ломаные образуют множество ω_0, которое мы отбрасываем на первом шаге. В результате мы получим множество

$$\Omega_1 = \Omega \setminus \omega_0.$$

Произведем теперь сужение оставшегося множества Ω_1. Для этого рассмотрим какую-либо точку $x_2 \in \Sigma_2$. Обозначим через $l(x_2)$ длину наиболее короткой ломаной, соединяющей точку x_2 с гиперплоскостью Σ_0. Очевидно, что

$$l(x_2) = \min_{x_1 \in G_1} (l(x_1) + f_1(x_1, x_2)). \tag{1.7}$$

При помощи (1.7) каждой точке x_2 мы поставили в соответствие число $l(x_2)$, т. е. определили на G_2 функцию $l(x_2)$.

Среди всех ломаных из множества Ω_1 мы можем, очевидно, отбросить те ломаные, которые не содержат отрезка ломаной $l(x_2)$. Это множество мы обозначим через ω_1. Итак, мы провели новое сужение множества допустимых вариантов. Наиболее короткая ломаная, соединяющая гиперплоскости Σ_0 и Σ_N, находится среди ломаных, принадлежащих множеству Ω_2:

$$\Omega_2 = \Omega \setminus \omega_0 \setminus \omega_1.$$

Продолжим теперь рассуждения по индукции. Пусть каждую из точек $x_i \in \Sigma_i$ мы соединили с гиперплоскостью Σ_0 ломаной наименьшей длины $l(x_i)$. Тогда длина наиболее короткой ломаной, соединяющей точку x_{i+1},

с гиперплоскостью Σ_0, будет определяться при помощи соотношения

$$l(x_{i+1}) = \min_{x_i \in G_i} (l(x_i) + f_i(x_i, x_{i+1})). \qquad (1.8)$$

На этом шаге все варианты, не содержащие ломаной $l(x_{i+1})$ и образующие множество ω_i, мы отбросим и перейдем к задаче отыскания вариантов, принадлежащих множеству Ω_{i+1}:

$$\Omega_{i+1} = \Omega \setminus \omega_0 \setminus \omega_1 \setminus \ldots \setminus \omega_i.$$

На последнем шаге каждой точке $x_N \in \Sigma_N$ поставим в соответствие число $l(x_N)$ — длину наиболее короткой ломаной, соединяющей точку x_N с гиперплоскостью Σ_0. Для того чтобы выбрать тот вариант, который нам нужен, — кратчайшую ломаную, соединяющую гиперплоскости Σ_0 и Σ_N, нам осталось совершить еще одну процедуру минимизации

$$l = \min_{x_N \in G_N} l(x_N).$$

На этой операции заканчивается процесс решения задачи. Изложенный метод дает возможность отыскать глобальный экстремум и принципиально может быть использован для отыскания экстремума аддитивных функций весьма общего вида

Формула (1.8) — это общее рекуррентное соотношение, описывающее многошаговый процесс отыскания решения, который сводит задачу отыскания минимума функции n переменных к последовательному отысканию минимума n функций одной переменной. В несколько ином виде схема расчета, выраженная формулой (1.8), была предложена в середине 50-х годов и получила название «киевского веника».

4. Численная реализация алгоритма «киевский веник». В предыдущем пункте мы описали процедуру отыскания минимума аддитивной функции, основанную на последовательном «отметании» неконкурентоспособных вариантов (отсюда и название — «веник»). Эта процедура сводится к последовательному решению функциональных уравнений (1.8). Только в исключительных случаях удается провести аналитическое исследование этой задачи, поэтому большое

значение приобретает проблема численного решения таких уравнений.

Для построения численных схем решения аддитивной задачи используют ее конечномерную аппроксимацию. С этой целью в пространстве $\{x, t\}$ строят сетку. Шаг по аргументу t задан: он равен единице. Зададим еще шаг по переменной x — вектор Δx. Узлы сетки обозначим через $P_k(i)$. Индекс i означает номер гиперплоскости Σ_i, а индекс k — номер узла в гиперплоскости Σ_i. Каждые два узла, лежащие на смежных гиперплоскостях, $P_k(i)$ и $P_j(i+1)$, мы можем соединить отрезками. Длину этих отрезков мы будем обозначать через $l_{kj}(i) = f_i(P_k(i), P_j(i+1))$. В результате этой операции мы получим некоторый граф специального вида (рис. 1.3), в котором роль вершин играют узлы $P_k(i)$. Вместо исходной мы будем рассматривать задачу отыскания на этом графе кратчайшего пути, соединяющего гиперплоскости Σ_0 и Σ_N.

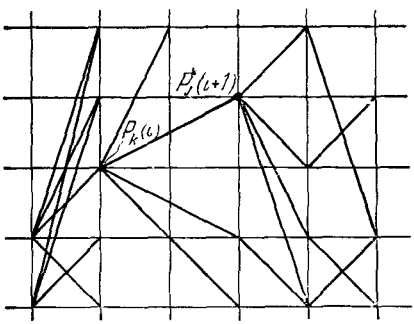


Рис. 1.3

Итак, вместо задачи отыскания траектории, принадлежащей множеству Ω, которое имеет мощность континуума, мы разыскиваем ломаную, принадлежащую конечному множеству $\Omega' \in \Omega$ всех тех ломаных из Ω, которые проходят через узлы сетки.

Обозначим через $l_k(i)$ ломаную кратчайшей длины из Ω', соединяющую узел $P_k(i)$ с гиперплоскостью Σ_0. Тогда, повторяя рассуждения предыдущего пункта, мы снова придем к рекуррентному соотношению (1.8), которое теперь будет выглядеть так:

$$l_s(i+1) = \min_{k \in M_i} \{l_k(i) + l_{ks}(i)\}. \qquad (1.9)$$

Минимум в (1.9) берется по тем k, для которых узлы лежат в допустимой области G_t и принадлежат гиперпло-

скости Σ_i. Число таких узлов обозначено через M_i:

$$P_k(i) \in G_i, \quad k=1, 2, \ldots, M_i.$$

Таким образом, на шаге номера $i+1$ для каждого узла $P_s(i+1)$, $s=1, \ldots, M_{i+1}$, мы осуществляем перебор M_i вариантов путей, соединяющих узел $P_s(i+1)$ с гиперплоскостью Σ_0, выбираем из этих путей кратчайший и запоминаем его. Всего на шаге номера $i+1$ мы должны запомнить M_{i+1} чисел $l_s(i+1)$, $s=1, 2, \ldots, M_{i+1}$.

Определение же величины $l_s(i+1)$ требует вычисления M_i функций

$$l_{ks}(i) = f_i(P_k(i), P_s(i+1)),$$

суммирования их с величиной $l_k(i)$, хранящейся в памяти машины, и сравнения между собой полученных величин. Пусть на это расходуется $M_i r$ машинных операций. Тогда общее число машинных операций, необходимое для реализации алгоритма, равно

$$Q = \sum_{i=0}^{N-1} M_i M_{i+1} r \leqslant M^2 r N, \tag{1.10}$$

где $M = \max\limits_i M_i$.

Число узлов M, очевидно, зависит от размерности задачи. Обозначим через d_{ki} число точек разбиения оси x^k гиперплоскостями Σ_i, и пусть $d = \max\limits_{i,\,k} d_{ki}$. Тогда

$$M \leqslant (d)^n, \tag{1.11}$$

где n — размерность вектора x.

Оценим теперь объем машинной памяти, необходимый для реализации алгоритма «киевский веник». На шаге $i+1$ необходимо помнить траектории, приходящие во все узлы гиперплоскости Σ_i. Всего таких траекторий M_i (по числу узлов). Каждая состоит из i точек в n-мерном пространстве. Отсюда легко получить оценку количества машинных ячеек, необходимых для запоминания траектории

$$R \leqslant (d)^n nN. \tag{1.12}$$

Мы видим, что число операций (1.10) и объем памяти (1.12) катастрофически растут с ростом n. Поэтому центральной трудностью, с которой приходится сталкиваться при реализации описанной процедуры, является размерность задачи.

5. **Метод «блуждающей трубки».** Алгоритм «киевский веник» дает возможность отыскать глобальный экстремум, причем для функций $f_i(x_i, x_{i+1})$ произвольного вида (например, не делалось никаких предположений об их выпуклости и пр.). Однако его реализация требует большой затраты машинного времени и, что может быть еще более важно, большой оперативной памяти машины Поэтому, естественно, что усилия исследователей, занимающихся вычислительными проблемами, были направлены на отыскание способов сокращения числа операций и объема необходимой памяти. Этого удалось достичь ценой отказа от поиска глобального экстремума. Соответствующие алгоритмы развивались в 60-е годы и один из них получил название «блуждающей трубки» (см. [7]). Этот алгоритм имеет характер метода последовательных приближений.

Пусть дано некоторое начальное приближение — ломаная Γ_0, которая задана последовательностью узлов $P_{k_0}(i)$ (рис. 1.4). Задавая шаг Δx, построим сетку S_0, причем

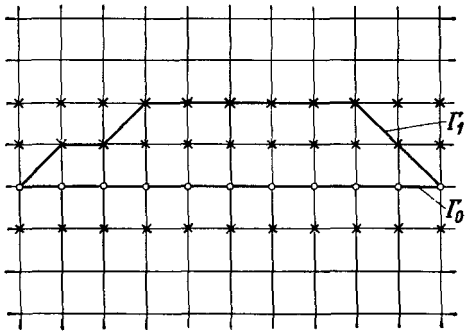


Рис. 1.4

в каждой из плоскостей Σ_i мы включаем в S_0 только по m узлов (на рис 1.4 эти узлы отмечены звездочками). На сетке S_0 реализуем вычислительную схему алгоритма «киевский веник», рекуррентное соотношение которого в этом случае имеет вид

$$l_s(i+1) = \min_{P_k(i) \in S_0} \{l_k(i) + l_{ks}(i)\}. \qquad (1.13)$$

Определив при помощи соотношения (1.13) новую ломаную Γ_1, мы повторим процедуру и т. д. Таким образом, на каждом шаге мы разыскиваем ломаную на некотором подграфе S_j.

Оценим число операций, необходимых для отыскания минимума (в общем случае локального) при помощи алгоритма «блуждающей трубки».

Если ломаная Γ_j известна, то для отыскания следующего приближения — ломаной Γ_{j+1} — нам необходимо произвести Q_j операций, где $Q_j \leqslant rm^2N$ Обозначим через k общее число итераций. Тогда количество машинных операций, необходимых для окончания процесса, будет $Q \leqslant \leqslant krm^2N$.

Чем больше число узлов на подграфах S_j, т. е. чем больше число m, тем меньшее число итераций необходимо для достижения минимума с заданной точностью. Следовательно, число k зависит от отношения M/m. Принимая $k \leqslant k_1 M/m$ для общего числа операций Q, получим оценку

$$Q \leqslant k_1 rMmN. \tag{1.14}$$

Таким образом, в отличие от «киевского веника», число итераций в методе «блуждающей трубки» растет линейно с увеличением числа узлов M.

Отметим в заключение, что хотя метод «блуждающей трубки» экономичнее «киевского веника», он так же, как и последний, бесперспективен для использования в задачах высокой размерности, так как число m узлов в трубке связано с размерностью n вектора x соотношением типа (1.11).

6. Метод локальных вариаций. Итак, мы установили, что чем меньше объем «блуждающей трубки», тем меньшее число операций требует реализация процедуры поиска решения. Это наводит на мысль о необходимости на каждом шаге итерационного процесса использовать трубку (подграф), содержащую наименьшее число узлов. Наводящие соображения такого рода лежат в основе метода локальных вариаций (см [7])

Пусть снова имеется некоторое начальное приближение Γ_0 (рис. 1.5). Наименьшим подграфом S_0, содержащим Γ_0, будет, очевидно, тот, который помимо узлов $P_0(i) \in \Gamma_0$ содержит всего лишь один узел $P_1(i)$. Длина звеньев ломаной Γ_0, соединяющей точку $P_0(i-1)$ с точкой

$P_0(i+1)$, равна

$$\alpha_0 = f_{i-1}(P_0(i-1),\ P_0(i)) + f_i(P_0(i),\ P_0(i+1)).$$

На графе S_0 существует еще одна ломаная, соединяющая точки $P_0(i-1)$ и $P_0(i+1)$, проходящая через узел $P_1(i)$. Ее длина будет такой:

$$\alpha_1 = f_{i-1}(P_0(i-1),\ P_1(i)) + f_i(P_1(i),\ P_0(i+1)).$$

Сравнивая величины α_0 и α_1, мы выбираем из них наименьшую.

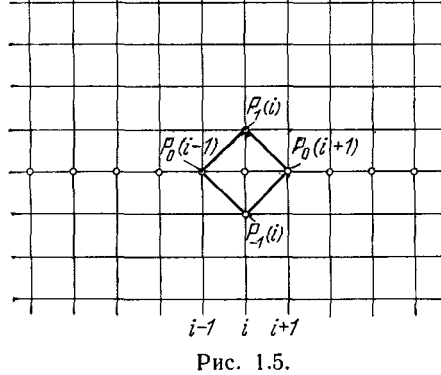


Рис. 1.5.

Структура сетки позволяет ввести узел $P_{-1}(i)$ — узел, симметричный $P_1(i)$ относительно Γ_0 (см. рис. 1.5). Предположим теперь, что имеет место неравенство $\alpha_0 > \alpha_1$. Тогда в качестве нового приближения (ломаной Γ_1) мы выбираем ломаную, проходящую через узел $P_1(i)$. Если же $\alpha_0 < \alpha_1$, проверяем ломаную, проходящую через узел $P_{-1}(i)$.

Вычисление величины α_1 носит название локального варьирования. Процесс последовательных приближений, использующий локальное варьирование, сводится, таким образом, к последовательному «улучшению» положения узлов, через которые проходит ломаная Γ_i.

Метод локальных вариаций можно рассматривать одновременно как метод покоординатного спуска (см. § 1 главы II) с фиксированным шагом на фиксированной сетке, заданной в области, определенной ограничениями.

В настоящее время метод локальных вариаций широко используется, что связано с его простотой в программировании и весьма скромными требованиями к объему оперативной памяти. Что касается объема необходимых вычислений, то он не очень отличается от того количества операций, которое необходимо для реализации метода «блуждающей трубки». Для метода локальных вариаций также справедлива оценка типа (1.14). Тем не менее, есть целый ряд ситуаций, в которых метод «блуждающей трубки» оказывается предпочтительнее метода локальных вариаций.

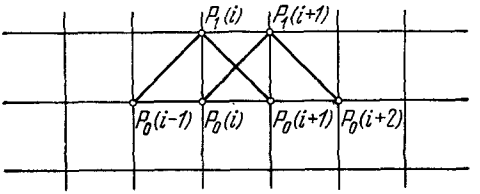


Рис. 1.6.

Последний, оказывается, очень чувствителен к локальным экстремумам, которые к тому же часто оказываются следствием неточностей процесса вычислений. Для иллюстрации сказанного приведем пример (рис. 1.6). Предположим, что исходное приближение (траектория Γ_0) проходит через точки $P_0(i-1)$, $P_0(i)$, $P_0(i+1)$, $P_0(i+2)$. Если мы начнем ее улучшать методом локальных вариаций, то мы должны сравнить траекторию Γ_0 с траекторией, проходящей через точки $P_0(i-1)$, $P_1(i)$, $P_0(i+1)$, с траекторией, проходящей через точки $P_0(i)$, $P_1(i+1)$, $P_0(i+2)$, и т. д. Предположим, что при этом окажутся справедливыми следующие неравенства:

$$f_{i-1}(P_0(i-1),\ P_0(i)) + f_i(P_0(i),\ P_0(i+1)) \leqslant$$
$$\leqslant f_{i-1}(P_0(i-1),\ P_1(i)) + f_i(P_1(i),\ P_0(i+1)),$$
$$f_i(P_0(i),\ P_0(i+1)) + f_{i+1}(P_0(i+1),\ P_0(i+2)) \leqslant$$
$$\leqslant f_i(P_0(i),\ P_1(i+1)) + f_{i+1}(P_1(i+1),\ P_0(i+2))$$

и т. д. Тогда в результате применения метода локальных вариаций мы должны сделать заключение о том, что ломаная Γ_0 и есть оптимальное решение. В действительности же ломаная наименьшей длины может проходить через узлы $P_1(i)$, $P_1(i+1)$, $P_1(i+2)$, ... Этот факт методом

локальных вариаций (на данной сетке) не выявить. В то же время он легко обнаруживается методом «блуждающей трубки».

Существуют различные модификации метода локальных вариаций, позволяющие обходить указанную трудность В качестве одной из них можно предложить следующую процедуру: на шаге номера i допустим к сравнению траектории, проходящие через узлы $P_0(i-1)$, $P_0(i)$, $P_1(i)$, $P_0(i+1)$, $P_1(i+1)$ и $P_0(i+2)$. Предположим, что отобранной оказывается траектория, проходящая через точки $P_0(i-1)$, $P_1(i)$, $P_1(i+1)$, $P_0(i+2)$. Тогда на следующем шаге мы рассматриваем совокупность траекторий, проходящих через точки $P_1(i)$, $P_1(i+1)$, $P_2(i+1)$, $P_1(i+2)$, $P_0(i+2)$, $P_0(i+3)$ и т. д. Однако все подобные приемы резко усложняют программирование и увеличивают необходимое время счета.

7. Стратегия поиска глобального экстремума. Алгоритм «киевский веник» дает возможность отыскать глобальный экстремум аддитивной функции, однако получение подобного решения требует значительной затраты машинного времени и возможно только при условии, что в нашем распоряжении имеется машина с большой оперативной памятью. Методы «блуждающей трубки» и локальных вариаций значительно более экономны, однако они пригодны для отыскания только локальных экстремумов. Поэтому, если априори известно, что исследуемая функция имеет единственный экстремум (например, если она выпукла), то следует применять один из этих методов.

В общем случае используется следующая схема расчетов. Сначала с большим шагом Δx_0 строится грубая сетка и на ней с помощью «киевского веника» ищется ломаная наименьшей длины — Γ_0. Затем делается «правдоподобное» предположение о том, что ломаная, являющаяся решением задачи, находится в окрестности Γ_0, где и строится новая сетка S_1 с меньшим шагом Δx_1. При этом шаг Δx_1 выбирается так, чтобы узлы сетки S_0 были включены в S_1. Затем методом «блуждающей трубки» на сетке S_1 находим ломаную Γ_1, снова дробим шаг, выбираем его равным Δx_2, строим новую сетку S_2 и т. д.

8. Замечание о градиентных методах. Структура аддитивных задач удобна и для применения градиентного спуска. В самом деле, пусть функция $f(x)$ имеет вид (1.4);

тогда

$$\frac{\partial f}{\partial x_t} = \frac{\partial f_t}{\partial x_t} + \frac{\partial f_{t-1}}{\partial x_t},$$

т. е. для того, чтобы вычислить производную по переменной номера t, достаточно продифференцировать лишь два слагаемых в (1.4). Заметим еще, что один шаг градиентного метода требует числа операций, пропорционального n (размерности векторов x_t), а не 2^n, как в методе локальных вариаций и в методе «блуждающей трубки», содержащей наименьшее возможное число узлов (при $m=2$). Однако применение градиентных методов предполагает дифференцируемость функции $f(x_1, \ldots, x_N)$. Методы же, излагавшиеся в этом параграфе, не требуют существования градиента $f'(x)$ и, как уже отмечалось выше, могут работать, когда ограничения (множества G_i) весьма сложны. Если размерность задачи невелика, а структура ограничений G_i сложна, схемы последовательного анализа будут более экономичными, нежели обычные методы нелинейного программирования (см. предыдущую главу). Если же размерность задачи очень велика, то методы последовательного анализа вариантов вообще неприменимы (по крайней мере, в том виде, как они были изложены), и более предпочтительными являются методы нелинейного программирования.

9. Принцип оптимальности. В теории случайных процессов известны так называемые *марковские процессы* или процессы без предыстории. Этим термином называют процесс, развитие которого при $t > t^*$ не зависит от характера его протекания при $t < t^*$. В настоящем параграфе уже рассматривались некоторые марковские процессы, проходящие в дискретном времени. В самом деле, при выводе основного рекуррентного соотношения (1.9) мы предполагали, что выбор участка траектории, приходящей слева в точку x_t (рис. 1.7), может быть сделан независимо от выбора его продолжения. Для процессов марковского типа Р. Беллман сформулировал так называемый *принцип оптимальности*. Введем понятие оптимальной траектории, соединяющей две заданные точки x_i и x_k. Этим термином будем называть связывающую x_i с x_k допустимую траекторию наименьшей длины $L(x_i, x_k)$ (под длиной, как и прежде, подразумевается часть аддитивного

критерия задачи, отвечающая отрезку «дискретного времени» от i до k).

Будем говорить, что процесс удовлетворяет принципу оптимальности, если любой участок оптимальной траектории является оптимальной траекторией. Например (рис. 1. 8), если наш процесс удовлетворяет принципу оптимальности и сплошной линией показана его единственная оптимальная траектория, то штриховая линия, соединяющая точки x_i и x_k и не совпадающая с участком оптимальной траектории процесса, не может быть оптимальной траекторией, соединяющей эти две точки.

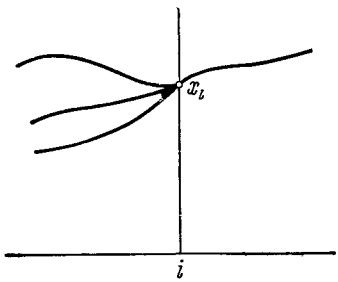


Рис. 1 7.

Очевидно, что процессы, описываемые аддитивными задачами, удовлетворяют принципу оптимальности. В

Рис 1 8.

самом деле, пусть оптимальная траектория проходит через точки x_i и x_k. Тогда

$$l(x_k) = \min_{x_0, x_1, x_2, \ldots, x_{k-1}} \left(\sum_{s=0}^{i-1} f_s(x_s, x_{s+1}) + \sum_{s=i}^{k-1} f_s(x_s, x_{s+1}) \right) =$$

$$= \min_{x_i, \ldots, x_{k-1}} \left(l(x_i) + \sum_{s=i}^{k-1} f_s(x_s, x_{s+1}) \right),$$

и так как точка x_i задана, то

$$l(x_k) = l(x_i) + \min_{x_{i+1}, \ldots, x_{k-1}} \sum_{s=i}^{k-1} f_s(x_s, x_{s+1}) = l(x_i) + L(x_i, x_k),$$

т. е. отрезок оптимальной траектории процесса на участке $(x_i x_k)$ совпадает с оптимальной траекторией, соединяющей точки x_i и x_k.

10. Квазиаддитивные задачи. Рассуждения, которые мы провели, показывают глубокую связь между марковскими процессами и процессами, удовлетворяющими принципу оптимальности Беллмана. Уравнение (1.9) является выражением этого принципа для аддитивных функций. Однако метод, использовавшийся для вывода этого уравнения, может быть применен и к немарковским процессам — тем, которые в своей исходной постановке не удовлетворяют принципу оптимальности. В качестве примера рассмотрим задачу отыскания минимума функции

$$f(x) = \sum_{i=1}^{N-1} f_i(x_{i-1}, x_i, x_{i+1}) \tag{1.15}$$

при ограничениях

$$x_i \in G_i, \quad i = 0, 1, \ldots, N.$$

Характер оптимальной траектории в данной задаче при $i > i^*$ зависит не только от номера i^*, но и от $i^* - 1$. Введем обозначение

$$l_j(x_j, x_{j+1}) = \min_{x_0, x_1, \ldots, x_{j-1}} \sum_{i=1}^{j} f_i(x_{i-1}, x_i, x_{i+1}).$$

Для отыскания минимума этой функции может быть использован алгоритм, который описывается рекуррентным соотношением

$$l_s(x_s, x_{s+1}) = \min_{x_{s-1}} \{l_{s-1}(x_{s-1}, x_s) + f_s(x_{s-1}, x_s, x_{s+1})\}. \tag{1.16}$$

Разумеется, этот алгоритм требует значительно большего объема вычислений и машинной памяти, чем (1.9). Легко проверить, что количество операций в этом случае пропорционально M^4N (M — число узлов), тогда как в аддитивных задачах оценка была M^2N. Отметим, что переход к рекуррентному соотношению (1.16) означает сведение квазиаддитивной задачи (1.15) к аддитивной задаче более высокой размерности.

§ 2. Дискретные управляемые системы

1. Постановки задачи. Рассмотрим динамический процесс, протекающий в дискретном времени. Последнее означает, что лишь в некоторые дискретные моменты времени возможно регистрировать его характеристики и принимать

решения об изменении управляющих воздействий. Обозначив состояние процесса в момент t_i через $x(t_i)$, где x есть n-мерный вектор, предположим, что он описывается уравнениями

$$x(t_{i+1}) = F_i(x(t_i),\ u(t_i)),\quad i = 0,\ 1,\ \ldots,\ N-1. \qquad (2.1)$$

Вектор $x(t_i)$ размерности n назовем фазовым вектором, m-мерный вектор $u(t_i)$ — управлением. Размерности этих векторов, вообще говоря, различны. Начальное состояние процесса считаем заданным:

$$x(t_0) = x_0. \qquad (2.2)$$

Подчиним фазовые векторы и управление следующим ограничениям:

$$x_i \equiv x(t_i) \in X_i,\quad i = 0,\ 1,\ \ldots,\ N, \qquad (2.3)$$

$$u_i \equiv u(t_i) \in U_i,\quad i = 0,\ 1,\ \ldots,\ N-1. \qquad (2.4)$$

Задача состоит в том, чтобы найти минимум функции вида

$$J = J(x_1,\ \ldots,\ x_N,\ u_1,\ \ldots,\ u_N) \qquad (2.5)$$

при ограничениях (2.1)—(2.4). В § 4 главы IV мы уже имели дело с дискретными управляемыми системами и получили для них необходимые условия оптимальности. В этом параграфе мы рассмотрим их с другой точки зрения. Прежде чем переходить к описанию конкретных задач, заметим, что ограничения на фазовые координаты и управления могут иметь более сложную структуру, чем (2.3), (2.4). На практике часто встречаются так называемые задачи со *смешанными ограничениями*. Последние имеют вид

$$\{x,\ u\} \in G.$$

Это выражение означает, что точки, определяющиеся парой векторов x, u, принадлежат некоторому множеству G из пространства E_{n+m}.

2. Задача о брахистохроне. Исследование дискретных систем начнем с классической задачи о брахистохроне, которая изучалась еще в конце XVII века И. Бернулли и была одной из задач, положивших начало вариационному исчислению.

Напомним ее постановку (рис. 2.1). Определить траекторию материальной точки, которая, двигаясь только под

действием силы тяжести, переместится из точки $O=\{0,\ 0\}^T$ в точку $A=\{b,\ -a\}^T$ за минимальное время. В начальный момент материальная точка находится в состоянии покоя. Эту классическую постановку задачи мы усложним дополнительным условием: *искомая кривая не должна пересекать заштрихованной области* (см. рис. 2.1).

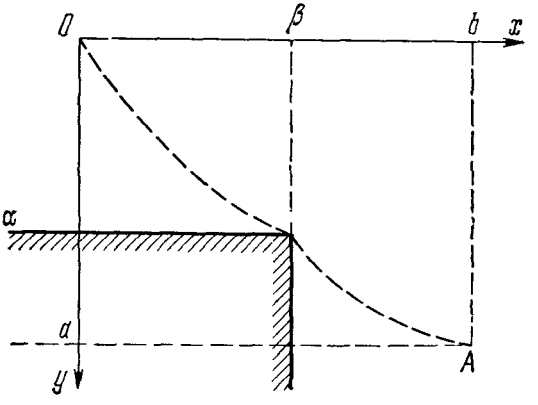


Рис. 2.1.

Обозначим через $ds=\sqrt{dx^2+dy^2}$ элемент дуги искомой гладкой кривой. Тогда скорость вдоль кривой определяется по формуле

$$v=\frac{ds}{dt}=\frac{dx}{dt}\sqrt{1+\left(\frac{dy}{dx}\right)^2}.$$

С другой стороны, $v=\sqrt{2gy}$; тогда

$$\frac{dx}{dt}=\sqrt{\frac{2gy}{1+(dy/dx)^2}}.$$

Из этого соотношения мы можем определить время, которое будет затрачено материальной точкой на перемещение из состояния O в состояние A:

$$T=\int_a^b\sqrt{\frac{1+(dy/dx)^2}{2gy}}\,dx.$$

Меняя очевидным образом обозначения, придем к следующей задаче: определить функции $u(t)$, $x(t)$, связанные

условием

$$\frac{dx}{dt} = u(t) \tag{2.6}$$

при начальном состоянии

$$x(0) = 0 \tag{2.7}$$

и доставляющие минимум интегралу

$$J = \int_a^b \sqrt{\frac{1+u^2}{2gx}}\, dt \tag{2.8}$$

при ограничениях

$$x \notin X, \tag{2.9}$$

где X — заштрихованная область плоскости x, y. Управление $u(t)$ предполагается неограниченным.

3. **Метод Эйлера.** Для решения подобных задач Л. Эйлером в середине XVIII века был предложен метод, который мы и рассмотрим применительно к задаче о брахистохроне.

Зададимся шагом $\Delta t = \tau$ и будем анализировать только управления типа $u(t) = u_i(t_i) = \text{const}$ при $t \in [t_i, t_{i+1}]$, где $t_i = \tau \cdot i$, тогда уравнение (2.6) заменится разностным уравнением

$$x_{i+1} = x_i + \tau u_i, \tag{2.10}$$

а интеграл (2.8) — суммой

$$J = \tau \sum_{i=0}^{N-1} f_i^*(x_i,\ u_i). \tag{2.11}$$

Здесь

$$f_i^* = \sqrt{\frac{1+u_i^2}{2gx_i}}, \quad x_i = x(t_i).$$

Тем самым задача о брахистохроне сводится к оптимизационной задаче для дискретной системы.

Условие (2.10) дает возможность исключить управление u_i и привести (2.11) к виду

$$J = \sum_{i=0}^{N-1} f_i(x_i,\ x_{i+1}), \tag{2.12}$$

где

$$f_i = \tau f^*\left(x_i,\ \frac{x_{i+1}-x_i}{\tau}\right).$$

Таким образом, метод Эйлера позволил преобразовать задачу о брахистохроне к аддитивной задаче оптимизации. Заметим, что ограничения на фазовые переменные (2.9), вообще говоря, уменьшают количество узлов на сетке в пространстве переменных (x, t), которые придется анализировать при решении задачи.

4. Элементарная операция. Для того чтобы задача отыскания фазовой траектории x_i и управления u_i, реализующих минимум функционала (2.5), сводилась к аддитивной, дискретная система должна обладать целым рядом свойств и, прежде всего, функционал должен иметь «аддитивную структуру», т. е. представляться в виде

$$J = \sum_{i=0}^{N-1} f_i(x_i,\ x_{i+1},\ u_i). \tag{2.13}$$

Кроме того, система должна допускать существование так называемой *элементарной операции*. Последняя определена, если паре точек x_i, x_{i+1} можно поставить в соответствие управление u_i, переводящее систему за один такт из состояния x_i в состояние x_{i+1}. Этот факт запишем в виде

$$u_i = B(x_i,\ x_{i+1}). \tag{2.14}$$

Тогда функция (2.13) примет вид

$$J = \sum_{i=0}^{N-1} \tilde{f}_i(x_i,\ x_{i+1}),$$

где

$$\tilde{f}_i = f_i(x_i,\ x_{i+1},\ B(x_i,\ x_{i+1})).$$

Для того чтобы элементарная операция (2.14) могла быть построена, необходимо, чтобы система (2.1) разрешалась относительно u_i. Для этого размерность вектора u_i должна быть не меньше размерности x_i. Заметим, что в задаче о брахистохроне элементарная операция была тривиальна: из (2.10) мы сразу получили

$$u_i = \frac{x_{i+1} - x_i}{\tau}. \tag{2.15}$$

Таким образом, если управление удается исключить и критерий качества имеет вид (2.13), мы приходим к аддитивной задаче, рассмотренной в предыдущем параграфе, и для ее решения можно использовать развитые там методы.

5. Функции с последовательным включением переменных. Другой путь исследования дискретных управляемых систем связан с исключением фазовых переменных. Рассмотрим для определенности задачу с целевой функцией

$$J = \sum_{i=1}^{N} F_i(x_i, u_{i-1}), \qquad F_N = F_N(x_N), \tag{2.16}$$

и конечно-разностными уравнениями

$$x_{i+1} = f_i(x_i, u_i), \quad i = 0, \ldots, N-1. \tag{2.17}$$

Пусть еще векторы x_i, u_i стеснены ограничениями вида (2.2) — (2.4). Поскольку начальное состояние x_0 задано, можно считать, что $f_0 = f_0(u_0)$. Уравнения (2.17) позволяют исключить векторы x_i из функционала (2.16):

$$J = F_1(u_0, f_0(u_0)) + F_2(u_1, f_1(u_1, f_0(u_0))) + \ldots + F_N(f_{N-1}(u_{N-1}, f_{N-2}(u_{N-2}, f_{N-3} \ldots)) \ldots). \tag{2.18}$$

Функции вида (2.18) будем называть *функциями с последовательным включением переменных.*

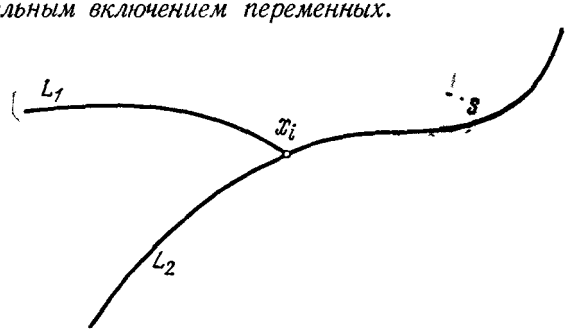


Рис 2.2.

Для любой последовательности управляющих векторов $u_0, u_1, \ldots, u_{N-1}$ можно вычислить соответствующие им фазовые переменные $x_1, \ldots, x_N$. Соединив эти точки отрезками, получим ломаную — фазовую траекторию системы (2.17). Таким образом, задачу можно снова сформулировать в терминах предыдущего параграфа: среди всех ломаных, соединяющих точку x_0 с множеством X_N, *найти* ту, длина которой минимальна, подразумевая под длиной

значение функции (2.18). Однако, хотя задача сформулирована так же, как для аддитивных функций, изложенные в предыдущем параграфе методы здесь не могут быть использованы. В самом деле, эти методы основывались на следующем правиле отбраковки: если две траектории, проходящие через точку x_i, имеют общее продолжение S (рис. 2.2), то мы выбирали ту из них, у которой ломаная, соединяющая точку x_0 с точкой x_i, имеет меньшую длину. Теперь мы так поступить не можем, хотя бы потому, что длина части траектории (L_1, L_2 на рис. 2.2) не идентифицируется ни с каким отрезком ряда (2.18).

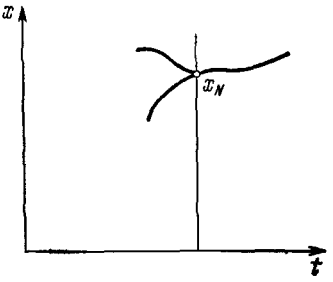


Рис. 2 3

Это обстоятельство наиболее наглядно можно продемонстрировать на простом частном случае, когда все $F_i = 0$, если $i < N$, а $F_N = F_N(x_N)$. Тогда значение целевой функции зависит только от x_N и не зависит от того, какова была траектория: любые траектории, входящие в точку x_N, эквивалентны с точки зрения критерия F_N (рис. 2.3). Таким образом, схема «киевский веник» для подобных задач неприменима. С другой стороны, совершенно очевидно, что процесс, который описывается системой (2.17), является марковским. Характер его течения при $t > t_i$ определяется только значением фазовой переменной при $t = t_i$ и не зависит от значений фазовой переменной при $t < t_i$, а структура функционала (2.16) такова, что для данной задачи справедлив принцип оптимальности Беллмана, который и лежал, как мы видели, в основе метода, изложенного в предыдущем параграфе. Поэтому возникает вопрос: как нужно изменить схему последовательного анализа, которая нас привела к алгоритму «киевский веник», чтобы охватить и рассматриваемый класс задач? Ответ на этот вопрос приведет нас к общей схеме динамического программирования.

6. Общая схема динамического программирования. Рассмотрим последний, N-й шаг процесса, описываемого системой уравнений (2.17), и предположим, что наша система

находится при $t = N - 1$ в некотором фиксированном состоянии x_{N-1}. Обозначим

$$J_k = \sum_{i=1}^{k} F_i(x_i, u_{i-1}). \tag{2.19}$$

Тогда

$$J = J_{N\ 1} + F_N(x_N).$$

Используя (2.17), представим J в следующем виде:

$$J = J_{N-1} + F_N(f_{N-1}(x_{N-1}, u_{N\ 1})).$$

Если система уже находится в состоянии x_{N-1}, то единственная возможность, которая есть в нашем распоряжении, — это так выбрать u_{N-1} — управление на последнем шаге, — чтобы минимизировать функцию

$$F_N(f_{N-1}(x_{N-1}, u_{N-1})).$$

Обозначим

$$S_{N-1}(x_{N-1}) = \min_{u_{N-1} \in U_{N-1}} F_N(f_{N\ 1}(x_{N-1}, u_{N-1})). \tag{2.20}$$

Равенство (2.20) определяет некоторую функцию положения x_{N-1}. Величина

$$J = J_{N-1} + S_{N-1}(x_{N-1})$$

определяет то минимальное значение функции J, которое может быть достигнуто, если траектория системы прошла через точки $x_0, x_1, \ldots, x_{N-1}$. В процессе вычисления величины $S_{N-1}(x_{N-1})$ мы находим управление u_{N-1}, которое доставляет наименьшее значение заданной функции

$$F_N(f_{N-1}(x_{N-1}, u_{N-1})).$$

Итак, одновременно со скалярной функцией $S_{N-1}(x_{N-1})$ мы определяем и вектор-функцию

$$u_{N-1} = \varphi_{N-1}(x_{N\ 1}), \tag{2.21}$$

которая ставит в соответствие каждому фазовому состоянию x_{N-1} вектор управления u_{N-1}. Если система в результате своей эволюции оказалась в состоянии x_{N-1}, то на последнем шаге управление должно быть выбрано согласно формуле (2.21).

По существу мы уже сформулировали правило отбрасывания неконкурентоспособных вариантов. В самом деле, обозначим через Ω множество всех последовательностей

$\{u_0, u_1, \ldots, u_{N-1}\}$, удовлетворяющих условиям $u_i \in U_i$. Каждая из таких последовательностей определяет фазовую траекторию $\{x_0, x_1, \ldots, x_N\}$. Поэтому той же буквой Ω мы будем обозначать и множество траекторий $\{x_0, x_1(u_0), x_2(u_0, u_1), \ldots, x_N(x_0, \ldots, u_{N-1})\}$. Через ω_1 мы обозначим подмножество Ω — совокупность всех тех последовательностей, у которых

$$u_{N-1} \neq \varphi_{N-1}(x_{N-1}).$$

Очевидно, что оптимальный вариант управления не может содержаться среди ω_1. Итак, на первом шаге вычислительной процедуры мы отбрасываем множество ω_1 и продолжаем поиск наилучшего варианта на множестве $\Omega_1 = \Omega \setminus \omega_1$.

Рассмотрим теперь второй шаг. Функцию J мы можем теперь переписать так:

$$\begin{aligned} J = J_{N-2} + F_{N-1}(x_{N-1}, u_{N-2}) + S_{N-1}(x_{N-1}) = \\ = J_{N-2} + F_{N-1}(f_{N-2}(x_{N-2}, u_{N-2})) + \\ + S_{N-1}(f_{N-2}(x_{N-2}, u_{N-2})). \end{aligned} \tag{2.22}$$

Следовательно, если зафиксировать точку x_{N-2}, наименьшее значение функции J определяется только управлением u_{N-2}. Но u_{N-2} входит лишь в два последних слагаемых в (2.22). Значит, если найти функцию

$$\begin{aligned} S_{N-2}(x_{N-2}) = \\ = \min_{u_{N-2} \in U_{N-2}} \{F_{N-1}(f_{N-2}(x_{N-2}, u_{N-2})) + S_{N-1}(f_{N-2}(x_{N-2}, u_{N-2}))\}, \end{aligned}$$

то величина

$$J = J_{N-2} + S_{N-2}(x_{N-2})$$

дает то минимальное значение функции J, которое может быть достигнуто, если траектория системы прошла через точки $x_0, x_1, \ldots, x_{N-2}$. Одновременно с $S_{N-2}(x_{N-2})$ мы определим также управление

$$u_{N-2} = \varphi_{N-2}(x_{N-2}).$$

Таким образом, если нам известно, что в «момент времени» $t = N-2$ система находится в состоянии x_{N-2}, то для того чтобы получить наименьшее значение функции J, управления u_{N-2} и u_{N-1} на двух последних шагах следует

выбрать, согласно формулам

$$u_{N-2} = \varphi_{N-2}(x_{N-2}),$$
$$u_{N-1} = \varphi_{N-1}(x_{N-1}) = \varphi_{N-1}(f_{N-2}(x_{N-2}, \varphi_{N-2}(x_{N-2}))).$$

На этом шаге мы исключаем из оставшегося множества траекторий Ω_1 множество ω_2 всех тех траекторий, для которых

$$u_{N-2} \neq \varphi_{N-2}(x_{N-2}).$$

Продолжая этот процесс, мы на каждом шаге определяем функцию

$$S_{l-1}(x_{l-1}) = \\ = \min_{u_{l-1} \in U_{l-1}} \{F_l(f_{l-1}(x_{l-1}, u_{l-1})) + S_l(f_{l-1}(x_{l-1}, u_{l-1}))\}, \quad (2.23)$$

которая состоянию системы x_{l-1} по формуле

$$J = J_{l-1} + S_{l-1}(x_{l-1})$$

ставит в соответствие то минимальное значение функции J, которое может быть достигнуто, если траектория системы прошла через точки $x_0, x_1, \ldots, x_{l-1}$. Одновременно будет получена функция

$$u_{l-1} = \varphi_{l-1}(x_{l-1}). \quad (2.24)$$

Она задает то значение управляющего вектора, с которым должен развиваться процесс при переходе от состояния x_{l-1} в состояние x_l вдоль оптимальной траектории. Все те варианты процесса, для которых вектор u_{l-1} отличен от вектора (2.24), отбрасываются.

Предположим, наконец, что определены функции $S_1(x_1)$ и $u_1 = \varphi_1(x_1)$; тогда

$$J = J_0 + S_1(x_1) = F_0(u_0) + S_1(x_1),$$

и нам осталось только определить величину u_0 и минимальное значение S_0 при заданном начальном состоянии x_0:

$$S_0 = \min_{u_0 \in U_0} \{F_0(u_0) + S_1(f_0(x_0, u_0))\}.$$

При этом мы находим и

$$u_0 = \varphi_0(x_0).$$

Задача решена: число S_0 — это минимальное значение функции J на множестве

$$U = U_0 \times U_1 \times \ldots \times U_{N-1}.$$

Для реализации этого значения S_0 мы должны построить последовательность векторов

$$u_0 = \varphi_0(x_0),$$
$$u_1 = \varphi_1(x_1) = \varphi_1(f_0(x_0,\ \varphi_0(u_0))),$$
$$u_2 = \varphi_2(x_2) = \varphi_2(f_1(x_1,\ u_1)) =$$
$$= \varphi_2(f_1(f_0(x_0,\ \varphi_0(x_0))),\ \varphi_1(f_0(x_0,\ \varphi_0(x_0))))$$

и т. д.

Уравнение (2.23), определяющее управление u_{i-1}, мы будем называть *уравнением Беллмана.*

7. Численная реализация описанной процедуры. Аналитическое решение уравнения (2.23), как правило, невозможно. Здесь нужны численные процедуры.

Для построения численной схемы в пространстве (x, t) снова построим сетку с некоторым шагом Δx. Узлы сетки будем обозначать через $P_k(i)$. Напомним, что i — это номер гиперплоскости Σ_i, а k — номер узла в гиперплоскости Σ_i. На первом шаге процесса нам нужно найти функцию

$$S_{N-1}(x_{N-1}) = \min_{u_{N-1} \in U_{N-1}} F_{N-1}(f_{N-1}(x_{N-1},\ u_{N-1})).$$

Для этого используем шкалу управлений. Этим термином называют таблицу значений вектора $u_i \in U_i$, построенную с некоторым шагом Δu. Элементы шкалы обозначим через $u_i(j)$, где j — номер элемента в U_i. Теперь задание функции S_{N-1} состоит в построении таблицы ее значений, зависящих от x_{N-1}, каждое из которых определяется перебором величин

$$S_{N-1}(P_k(N-1)) = \min_j F_N(f_{N-1}(P_k(N-1),\ u_{N-1}(j))).$$

Эта операция определит также функцию

$$u_{N-1}(k) = \varphi_{N-1}(P_k(N-1)).$$

Эту таблицу следует хранить в памяти ЭВМ.

Рассмотрим следующий шаг процесса. Здесь сразу возникает новая трудность. На этом шаге мы должны построить таблицу для функции

$$S_{N-2}(P_k(N-2)) =$$
$$= \min_j \{F_{N-1}(f_{N-2}(P_k(N-2),\ u_{N-2}(j))) +$$
$$+ S_{N-1}(f_{N-2}(P_k(N-2),\ u_{N-2}(j)))\}.$$

Но функция S_{N-1} нам задана таблично на сетке узлов $P_l(N-2)$, и среди этих узлов просто может не оказаться такого, чтобы его координаты определялись компонентами вектора $f_{N-2}(P_k(N-2), u_{N-2}(j))$. Другими словами, задав какое-либо значение $u_{N-2}(j)$ из шкалы управлений, в общем случае мы получим точку x_{N-1}, которая не совпадает ни с одним из узлов в гиперплоскости Σ_{N-1} и, следовательно, ее нет в таблице значений функции S_{N-1}.

Переход из заданного состояния x_{N-2} в заданное состояние x_{N-1} возможен лишь в том случае, когда существует управление u_{N-2}, которое является корнем векторного уравнения

$$x_{N-1} = f_{N-2}(x_{N-2}, u_{N-2}), \tag{2.25}$$

а для этого необходимо, по крайней мере, чтобы размерность вектора u была не меньше размерности вектора x. Предположим сначала, что этот факт имеет место *).

При фиксированном x_{N-2} формула (2.25) дает некоторое отображение U_{N-2} на Σ_{N-1}. Этот образ обозначим через $Q(U_{N-2}, x_{N-2})$. Если некоторый узел $P_j(N-1) \in Q(U_{N-2}, x_{N-2})$, то мы говорим, что он достижим из точки x_{N-2}. В противном случае мы говорим, что он не достижим из точки x_{N-2}, а множество $Q(U_{N-2}, x_{N-2})$ мы называем множеством достижимости из точки x_{N-2}.

Элементарной операцией (применительно к данному случаю) мы называем процедуру отыскания действительного корня уравнения (2.25), т. е. функции u_{N-2}, зависящей от x_{N-1} и x_{N-2}:

$$u_{N-2} = \varphi^*_{N-2}(x_{N-1}, x_{N-2}).$$

Используя элементарную операцию, мы можем представить функцию S_{N-2} в следующем виде:

$$S_{N-2}(P_k(N-2)) = \min_l \{F_{N-1}(f_{N-2}(P_k(N-2), \varphi^*_{N-2}(P_l(N-1), P_k(N-2)))) + S_{N-1}(f_{N-2}(P_k(N-2), \varphi^*_{N-2}(P_l(N-1), P_k(N-2))))\}.$$

Для того чтобы построить теперь таблицу функции $S_{N-2}(P_k(N-2))$, нам достаточно перебрать значения узлов $P_j(N-1)$.

*) При этом подходе шкала управлений не используется.

Заметим, что достаточно заполнить лишь таблицу управления $u_{N-2}(P_k(N-2))$.

Если размерность вектора u меньше размерности вектора x, то ситуация значительно усложняется, так как размерность множества достижимости в общем случае оказывается меньшей, нежели размерность пространства Σ_{N-1}.

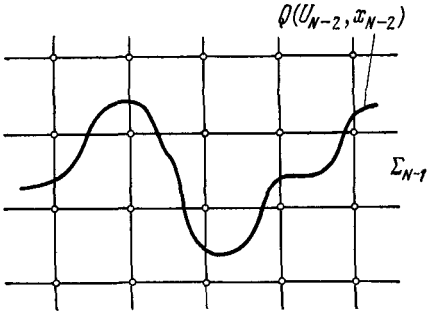


Рис. 2.4.

Поясним ситуацию, которая здесь складывается, на простом примере (рис. 2.4). Пусть размерность вектора x равна двум, а u — скаляр. Тогда векторное уравнение (2.17) будет эквивалентно двум скалярным уравнениям

$$\begin{aligned} x^1_{N-1} &= f^1_{N-2}(x^1_{N-2},\ x^2_{N-2},\ u_{N-2}), \\ x^2_{N-1} &= f^2_{N-2}(x^1_{N-2},\ x^2_{N-2},\ u_{N-2}). \end{aligned} \tag{2.26}$$

При фиксированном x_{N-2} уравнения (2.26) можно рассматривать как параметрическое задание некоторой кривой в гиперплоскости Σ_{N-1}. Эта кривая в общем случае не проходит ни через один из узлов. Следовательно, если формально использовать приведенные выше рассуждения, то мы должны сказать, что ни один из узлов в гиперплоскости Σ_{N-1} не достижим из точки x_{N-2} и, следовательно, все траектории, входящие в точку x_{N-2}, должны обрываться. Вот почему в этом случае все построения должны быть существенным образом модифицированы.

Каждый из узлов $P_j(i)$ мы окружим некоторым множеством $R_j(i)$. Например, мы говорим, что $x_l \in R_j(i)$, если точка x_l лежит внутри параллелепипеда с центром в точке $P_j(i)$ и длиной ребра, равной 2ε, где ε — некоторое заданное число.

Будем теперь относить к числу узлов, достижимых из точки x_{i-1}, все те узлы, окрестности которых $R_j(i)$ имеют с множеством достижимости общие точки. Рассмотрим более подробно, к чему приведет подобное расширение множества допустимых узлов на примере перехода из состояния x_{i-1} в состояние x_i.

Итак, пусть функция $S_i(x_i)$ построена. Это значит, что в нашем распоряжении есть таблица значений $S_i(P_j(i))$ и управление $u_i(x_i)$. Запишем функции $S_i(x_i)$ и $F_i(x_i, u_i)$ в виде

$$S_i(x_i) = S_i(f_{i-1}(x_{i-1}, u_{i-1})),$$
$$F_i(x_i, u_i) = F_i(f_{i-1}(x_{i-1}, u_{i-1})).$$

Фиксируем точку x_{i-1} — рассматриваем один из узлов $P_j(i-1)$.

Снова введем *шкалу управлений*: заменим множество U_{i-1} некоторым конечным множеством, состоящим из точек $u_{i-1}(0), \ldots, u_{i-1}(L)$. Вычисляем последовательно $f_{i-1}(x_{i-1}, u_{i-1}(0))$, $f_{i-1}(x_{i-1}, u_{i-1}(1))$ и т. д. Вычислив, например, $f_{i-1}(x_{i-1}, u_{i-1}(j))$, мы должны выяснить, будет ли этот вектор принадлежать к окрестности одного из узлов в гиперплоскости Σ_i. Если такой узел $P_k(i)$, в окрестности которого окажется точка $f_{i-1}(x_{i-1}, u_{i-1}(j))$, — существует, то эту точку мы идентифицируем с узлом $P_k(i)$ и запоминаем соответствующее ему управление $u_{i-1}(j_k)$. Теперь мы определяем функции

$$\begin{aligned} S_i(P_k(i)) &= S_i(f_{i-1}(x_{i-1}, u_{i-1}(j_k))), \\ F_i(P_k(i)) &= F_i(f_{i-1}(x_{i-1}, u_{i-1}(j_k))). \end{aligned} \tag{2.27}$$

Выражения (2.27) — это некоторые таблицы. Имея эти таблицы в своем распоряжении, мы легко строим таблицу и для функции

$$S_{i-1}(x_{i-1}) = \min_k \{F_i(f_{i-1}(x_{i-1}, u_{i-1}(j_k))) + S_i(f_{i-1}(x_{i-1}, u_{i-1}(j_k)))\}. \tag{2.28}$$

Операция (2.28) каждому узлу $P_r(i-1)$, лежащему в плоскости Σ_{i-1}, ставит в соответствие единственное значение управления $u_{i-1}(j_r)$. Продолжая этот процесс, мы найдем некоторую последовательность управлений $u_0, u_1, \ldots$ u_{N-1}, и с их помощью, используя формулы (2.17), мы можем построить фазовую траекторию $x_0, x_1, \ldots, x_N$

и, следовательно, вычислить минимальное значение J. Изложенная процедура тем точнее дает возможность вычислить минимальное значение функции J, чем на более мелкой сетке мы проводим вычисления. Но увеличение числа узлов приводит к быстрому увеличению необходимой памяти машины и затрат машинного времени. Поэтому при решении подобных задач мы всегда вынуждены использовать какие-либо итеративные методы. В предыдущих разделах этого параграфа мы уже рассматривали некоторые из подобных методов, например, метод «блуждающей трубки». Нетрудно убедиться в том, что этот метод полностью переносится и на общий случай задачи динамического программирования.

§ 3. Задача о коммивояжере и ее обобщения

1. Постановка задачи о коммивояжере. До сих пор в этой главе мы рассматривали задачи, удовлетворяющие принципу оптимальности Беллмана. Для их решения может быть использован один из простейших способов последовательного анализа вариантов — метод динамического программирования. Сейчас мы переходим к изучению некоторых задач, которые уже не удовлетворяют принципу оптимальности, и, следовательно, для этих задач метод динамического программирования непосредственно использован быть не может. Их решение требует развития специальных способов последовательного анализа вариантов. Изучение подобных задач мы начнем с анализа классической задачи о коммивояжере (бродячем торговце).

Предположим, что бродячий торговец должен, покинув город, которому мы присвоим номер 1 (рис. 3.1), объехать еще $N-1$ городов и вернуться снова в город номер 1. В его распоряжении есть дороги, соединяющие эти города Он должен выбрать свой маршрут — порядок посещения городов так, чтобы путь, который ему придется пройти, был как можно короче. Основное условие этой задачи состоит в том, что коммивояжер не имеет права возвращаться снова в тот город, в котором он однажды уже побывал. Это условие будем называть условием (α). Мы считаем, что расстояние между двумя городами — функция $f(x_i, x_j)$ — определено. Разумеется, функция $f(x_i, x_j)$ может

означать не только расстояние, но, например, время или издержки в пути и т. д. Поэтому в общем случае

$$f(x_i, x_j) \neq f(x_j, x_i),$$

а функции $f(x_i, x_i)$ естественно приписать значение ∞. Длина l пути S определяется формулой

$$l = \sum_{i,j} f(x_i, x_j). \tag{3.1}$$

Сумма в выражении (3.1) распространена по всем индексам i и j, удовлетворяющим условию (α), т. е. условию,

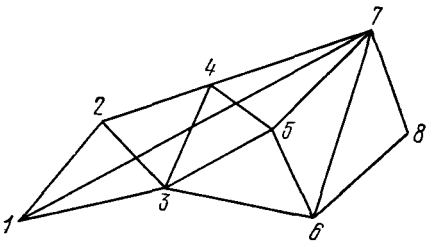


Рис. 3.1.

что каждый из индексов i и j входит в выражение (3.1) один и только один раз. Функция $l = l(x_1, \ldots, x_N)$ является, таким образом, аддитивной — она представима в виде суммы слагаемых, однако сама задача — задача отыскания минимума l — в силу ограничения (α) не является аддитивной и не удовлетворяет принципу оптимальности.

Рассмотрим снова плоскость t, x, где t — дискретный аргумент, принимающий значения 0, 1, 2, ..., N, соответствующие этапам путешествия бродячего торговца. Значение $t = 0$ соответствует его начальному положению в городе номер 1, $t = 1$ — переходу из города номер 1 в город, который он выбрал первым для посещения, и т. д., $t = N$ означает последний этап его путешествия — возвращение в город номер 1. Аргумент x теперь также принимает дискретные значения 1, 2, ..., N (рис. 3.2). Соединим точку (0,1) с точками (1,1), (1,2) ... (1, N) и длинам отрезков, соединяющих эти точки, припишем значения $f(x_1, x_j)$. Далее точки (1, s) — узлы, лежащие на первой вертикали, мы соединим со всеми узлами второй верти-

кали, длинам отрезков мы припишем значения $f(x_s, x_k)$ и т. д. Точки $(N-1, s)$ соединим с точкой $(N, 1)$.

В результате мы построили некоторый граф, каждая ломаная которого, соединяющая точку (0, 1) с точкой $(N, 1)$, описывает путь коммивояжера. Нашу задачу мы можем теперь сформулировать следующим образом. Среди всех ломаных, принадлежащих этому графу и соединяющих точки (0, 1) и $(N, 1)$ и удовлетворяющих условию (α),

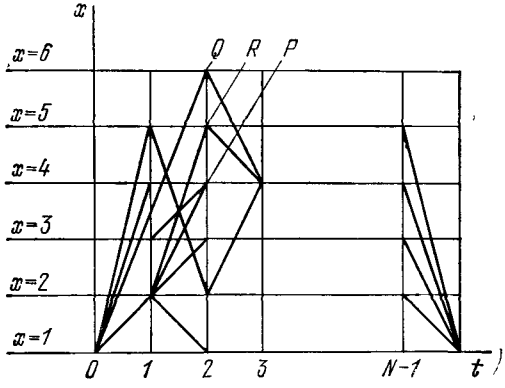


Рис. 3.2.

найти ломаную кратчайшей длины. Условие (α) состоит теперь в том, что искомая ломаная пересекает (в узле) каждую из прямых $x=i$ один и только один раз. Таким образом, мы смогли задачу о коммивояжере сформулировать на том же языке, на котором формулировали ранее задачи, рассмотренные в предыдущих параграфах. Формулировки кажутся почти идентичными. Существенное различие состоит только в существовании условия (α). Однако именно это условие делает задачу коммивояжера качественно отличной от задач, рассмотренных в предыдущих параграфах этой главы.

Рассмотрим узел P, лежащий на третьей вертикали (см. рис. 3.2). Если бы условие (α) отсутствовало, то выбор траектории, которая соединяет точку P с точкой $(N, 1)$, не зависел бы от того пути, который привел нас в точку P. В данном случае ситуация иная, и если два коммивояжера находятся в точке P, но один из них пришел в это

состояние, двигаясь вдоль траектории, проходящей через точку Q, а второй через точку R, то их состояния существенно отличаются друг от друга. Коммивояжер, который двигался по второй траектории, уже побывал в городах номер 2 и номер 5 и в будущем он уже не имеет права снова заезжать в эти города. Что касается коммивояжера, который двигался вдоль первой траектории, то он побывал в городах номер 3 и номер 6; он не имеет права возвращаться в эти города, но зато он еще обязан посетить города номер 2 и номер 5 и т. д.

j \ i	1	2		n
1	∞	c_{12}		c_{1n}
2	c_{21}	∞		c_{2n}
n	c_{n1}	c_{n2}		∞

Рис. 3.3.

Поскольку функция $f(x_i, x_j)$ определена на конечном множестве точек, то и функция $l(x_1, \ldots, x_N)$ также определена на конечном множестве точек. Следовательно, задача определения минимума функции l сводится к перебору некоторого конечного множества значений этой функции, и проблема носит чисто вычислительный характер. Однако именно вычислительные трудности здесь огромны. Легко подсчитать, что число возможных вариантов (число значений функции l) равно $(N-1)!$. Таким образом, непосредственно перебрать и сравнить между собой все возможные пути, по которым может следовать бродячий торговец, для достаточно большого количества городов практически невозможно. Возникает проблема построения такого

метода последовательного анализа вариантов, который выделял бы по возможности большое количество неперспективных вариантов и сводил задачу к перебору относительно небольшого количества «подозрительных» вариантов.

2. Метод ветвей и границ. Основа этого, ныне широко распространенного метода состоит в построении нижних оценок решения, которые затем используются для отбраковки неконкурентоспособных вариантов.

Функция $f(x_i, x_j)$ принимает конечное число значений c_{ij}, которые мы можем представить в виде таблицы (рис. 3.3). Предположим, что мы выбрали некоторый путь S_s. Его длина будет равна

$$l_s = \sum_{i,j} c_{ij}, \tag{3.2}$$

причем сумма (3.2) распространена по i, j так, что каждый из индексов встречается в ней один и только один раз. Величины c_{ij} с двумя одинаковыми индексами мы приняли равными ∞.

Так как в каждый из вариантов s входит только один элемент из каждой строки и столбца, то мы можем проделать следующую операцию, которая здесь называется приведением матрицы. Обозначим через h_i наименьший элемент из строки номера i и построим новую матрицу $C^{(1)}$ с элементами

$$c_{ij}^{(1)} = c_{ij} - h_i.$$

Матрица $C^{(1)}$ определяет новую задачу коммивояжера, которая, однако, в качестве оптимальной будет иметь ту же последовательность городов. Между величинами l_s и $l_s^{(1)}$ будет существовать, очевидно, следующая связь:

$$l_s = l_s^{(1)} + \sum_{i=1}^{N} h_i.$$

Заметим, что в каждой из строк матрицы $C^{(1)}$ будет теперь, по крайней мере, один нулевой элемент. Далее обозначим через g_j наименьший элемент матрицы $C^{(1)}$, лежащий в столбце номера j, и построим новую матрицу $C^{(2)}$ с элементами

$$c_{ij}^{(2)} = c_{ij}^{(1)} - g_j.$$

Величины h_i и g_j называются константами приведения. Оптимальная последовательность городов для задачи

коммивояжера с матрицей $C^{(2)}$ будет, очевидно, такой же, как и для исходной задачи, а длины пути для варианта номера s в обоих задачах будут связаны между собой равенством

$$l_s = l_s^2 + d_0, \tag{3.3}$$

где

$$d_0 = \sum_{i=1}^{N} h_i + \sum_{j=1}^{N} g_j, \tag{3.4}$$

т. е. d_0 равна сумме констант приведения.

Обозначим через l^* решение задачи коммивояжера, т. е.

$$l^* = \min l_s,$$

где минимум берется по всем вариантам s, удовлетворяющим условию (α) Тогда величина d_0 будет простейшей нижней оценкой решения:

$$l^* \geqslant d_0. \tag{3.5}$$

Будем рассматривать теперь задачу коммивояжера с матрицей $C^{(2)}$, которую мы будем называть приведенной матрицей.

Рассмотрим путь, содержащий непосредственный переход из города номера i в город номера j, тогда для пути s, содержащего этот переход, мы будем иметь, очевидно, следующую нижнюю оценку:

$$l_s \geqslant d_0 + c_{ij}^{(2)}.$$

Следовательно, для тех переходов, для которых $c_{ij}^{(2)} = 0$, мы будем иметь снова оценку (3.5). Естественно ожидать, что кратчайший путь содержит один из таких переходов — примем это соображение в качестве рабочей гипотезы. Рассмотрим один из переходов, для которого $c_{ij}^2 = 0$, и обозначим через $(\overline{ij})$ множество всех тех путей, которые не содержат перехода из i в j. Так как из города i мы должны куда-то выйти, то множество $(\overline{ij})$ содержит один из переходов $i \to k$, где $k \neq j$; так как в город номера j мы должны прийти, то множество $(\overline{ij})$ содержит переход $m \to j$, где $m \neq i$. Следовательно, некоторый путь l_s из множества $(\overline{ij})$, содержащий переходы $i \to k$ и $m \to j$, будет иметь следующую нижнюю оценку:

$$l_s \geqslant d_0 + c_{ik}^2 + c_{mj}^2.$$

Обозначим через

$$\theta_{ij} = \min_{k \neq j} c^{2}_{ik} + \min_{m \neq i} c^{(2)}_{mj}.$$

Тогда очевидно, что для любого l_s из множества путей $(\overline{ij})$ мы будем иметь оценку

$$l_s \geqslant d_0 + \theta_{ij}. \tag{3.6}$$

Мы предполагаем исключить некоторое множество вариантов $(\overline{ij})$, поэтому мы заинтересованы выбрать такой переход $i \to j$, для которого оценка (3.6) была бы самой высокой. Другими словами, среди нулевых элементов матрицы $C^{(2)}$ выберем тот, для которого θ_{ij} максимально. Это число обозначим через $\hat{\theta}_2$. Таким образом, все множество возможных вариантов мы разбили на два множества I_1 и I_2. Для путей из множества I_1 мы имеем оценку (3.5). Для путей из множества I_2 оценка будет следующей:

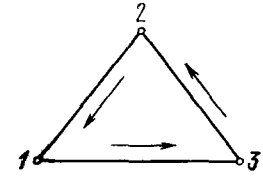
$$l_s \geqslant d_0 + \hat{\theta}_2 = d_2. \tag{3.7}$$

Рассмотрим теперь множество I_1 и матрицу $C^{(2)}$. Так как все пути, принадлежащие этому множеству, содержат переход $i \to j$, то для его исследования нам достаточно рассмотреть задачу коммивояжера, в которой города номеров i и j совпадают. Размерность этой задачи будет уже равна $N-1$, а ее матрица получится из матрицы $C^{(2)}$ вычеркиванием столбца номера j и строки номера i.

Поскольку переход $j \to i$ невозможен, то элемент $c^{(2)}_{ji}$ принимаем равным бесконечности.

	1	2	3	
1	∞	C_{12}	C_{13}	
2	C_{21}	∞	C_{23}	
3	C_{31}	C_{32}	∞	

а)



б)

	1	3	
1	∞	C_{13}	
2	C_{21}	∞	

в)

Рис 3 4.

Рассмотрим случай $N=3$ (рис. 3.4, *а*) и предположим, что мы рассматриваем тот вариант, который содержит переход $3 \to 2$. Тогда задача коммивояжера после

вычеркивания третьей строки и второго столбца вырождается в тривиальную. Ее матрица изображена на рис. 3.4, *в*. В этом случае мы имеем единственный путь, и его длина будет, очевидно, равна сумме

$$l = c_{13} + c_{21}.$$

Итак, если в результате вычеркивания строки номера i и столбца номера j мы получим матрицу второго порядка, то задачу можно считать решенной.

Пусть теперь $N > 3$. После вычеркивания мы получим матрицу порядка $N - 1 > 2$.

С этой матрицей $(N-1)$-го порядка совершим процедуру приведения. Матрицу, которую таким образом получим, обозначим через $C^{(3)}$, а через $d^{(1)}$ — сумму ее констант приведения. Тогда для $l_s \in \mathrm{I}_1$, мы будем иметь оценку

$$l_s \geqslant d_0 + d^{(1)} = d_1. \tag{3.8}$$

На этом первый шаг алгоритма закончен. В результате одного шага мы разбили множество всех возможных вариантов на два множества I_1 и I_2 и для путей, принадлежащих этим множествам, мы получили оценки (3.8) и (3.7) (рис. 3.5).

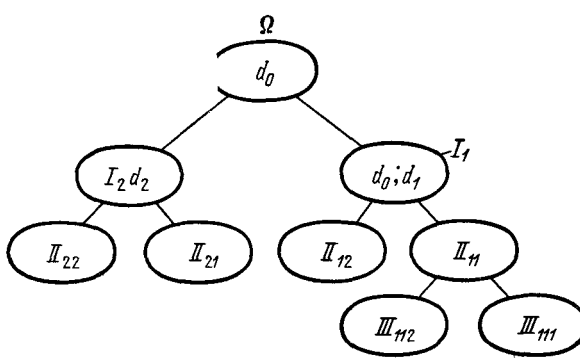


Рис. 3.5.

Введем понятие стандартной операции, которую мы будем обозначать символом $\Omega((sl), d_r, d_k)$. Этим термином мы назовем процедуру разбиения произвольного множества вариантов Ω с приведенной матрицей N — n-го порядка $C^{(n+2)}$

и оценкой d_ω на два множества. Одно из этих множеств состоит из всех тех путей, которые содержат переход из города номер s в город номер l и имеют нижнюю оценку d. Другое множество состоит из всех путей, не содержащих этого перехода и имеющих в качестве нижней оценки число d_k. Стандартную операцию можно представить в форме следующей блок-схемы (см. рис. 3.6).

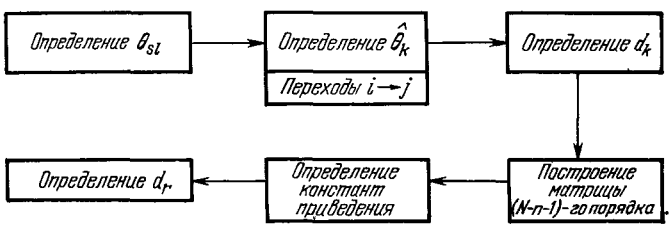


Рис. 3.6.

Итак, первый шаг метода ветвей и границ состоит в проведении стандартной операции над исходным множеством Ω. На следующем шаге мы продолжаем развивать дерево возможных вариантов. Сначала мы сравниваем две оценки d_1 и d_2 и для последующего анализа выбираем то из множеств I_1 или I_2, для которого соответствующая оценка меньше.

Предположим, что

$$d_1 < d_2;$$

тогда над множеством I_1 с матрицей $C^{(3)}$ мы совершим стандартную операцию. В результате мы разобьем множество возможных вариантов I_1 на два подмножества II_{11} и II_{12}, первое из которых содержит некоторый переход $i_1 \to j_1$, а другое содержит все пути, не имеющие непосредственного перехода из города i_1 в город j_1. Еще раз повторим рассмотренную выше процедуру: для каждого из нулевых элементов матрицы $C^{(3)}$ построим число

$$\theta_{i_1 j_1} = \min_{k \neq i_1} c^{(3)}_{i_1 k} + \min_{m \neq j_1} c^{(3)}_{m j_1},$$

определим значение

$$\hat{\theta}_{12} = \max_{i_1,\ j_1} \theta_{i_1 j_1}$$

и элемент матрицы $C^{(3)}$, для которого достигается это значение. Если $l_s \in \Pi_{12}$, то

$$l_s \geqslant d_1 + \hat{\theta}_{12} = d_{12}. \tag{3.9}$$

Затем в матрице $C^{(3)}$ вычеркиваем строку номера i_1 и столбец номера j_1, полагаем $c_{j_1 i_1} = \infty$ и над полученной матрицей совершаем операцию приведения. В результате мы найдем новые константы приведения. Их сумму обозначим через $d^{(11)}$ и в заключение находим оценку d_{11} для элементов множества Π_{11}.

Если $l_s \in \Pi_{11}$, то

$$l_s \geqslant d_1 + d^{(11)} = d_{11}. \tag{3.10}$$

На этом второй шаг алгоритма ветвей и границ закончен. Мы разбили множество вариантов I_1 на два множества, Π_{11} и Π_{12}, и для элементов этих множеств получили нижние оценки (3.10) и (3.9), соответственно.

Теперь мы должны сравнить оценку (3.10) с оценкой (3.7) для элементов множества I_2, которое мы исключили из рассмотрения на предыдущем шаге. Если окажется, что

$$d_2 > d_{11},$$

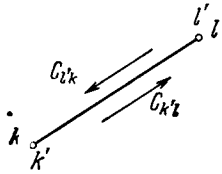
то мы переходим к третьему шагу, который состоит в применении стандартной операции к множеству Π_{11}. (Если размерность матрицы при этом равна двум, то, как мы видели выше, процесс заканчивается.)

Если окажется, что $d_{11} > d_2$, то множеством вариантов с оптимальной нижней оценкой будет множество I_2. Другими словами, теперь будем предполагать, что наиболее короткий путь содержится среди элементов множества I_2 — множества всех вариантов, не содержащих перехода $i \to j$. Следовательно, матрица, характеризующая это множество, получается из матрицы $C^{(2)}$ заменой величины $c_{ji}^{2)}$ на ∞. Над этим множеством мы производим стандартную операцию и разбиваем его на два множества Π_{21} и Π_{22} с оценками d_{21} и d_{22}, соответственно. Одновременно мы выделяем переход $k \to l$, который содержит все варианты множества Π_{21}. Затем мы снова сравниваем все оценки d_{11}, d_{12}, d_{21} и d_{22} и выбираем то из множеств, для которого оценка будет наименьшей. Над выбранным множеством совершаем стандартную операцию и т. д. Так мы продолжаем до тех пор, пока очередная матрица не будет иметь

порядок (2×2). В этом случае, как мы видели, расчет заканчивается — мы получаем задачу коммивояжера для двух городов (рис. 3.7), и длина единственного маршрута будет

$$l_s = c_{k'l} + c_{l'k}.$$

Итак, мы получили некоторую цепочку (ветвь) переходов, длину которой мы вычислили. Сам порядок построения этой цепочки показывает, что ее длина — наименьшая среди всех ветвей дерева, изображенного на рис. 3.5.



а)

	k	l	
k'	∞	$C_{k'l}$	
l'	$C_{l'k}$	∞	

б)

Рис. 3.7.

Примечание. Для фактического построения цепочки переходов мы должны, начав с конца ветви, вспомнить все переходы, которые содержали выбранные нами множества.

3. Использование верхних оценок. Существует целый ряд модификаций метода ветвей и границ. Некоторые из них заметно ускоряют процедуру счета. Другие оказываются весьма полезными при построении приближенных решений. К числу последних относится использование верхних оценок. Рассмотрим один из способов построения этих оценок.

Вернемся снова к дереву вариантов, изображенному на рис. 3.5.

На первом шаге мы разбили все множества вариантов на два подмножества I_1 и I_2, нижние оценки для которых соответственно d_1 и d_2; примем $d_1 < d_2$. На втором шаге мы рассматриваем множество I_1 и разбиваем его на два множества II_{11} и II_{12} с оценками d_{11} и d_{12}. Пусть $d_{11} < d_{12}$. Далее, следуя общей схеме метода ветвей и границ, мы должны были бы вернуться к рассмотрению множества I_2.

Теперь мы введем в процедуру данного шага еще операцию построения верхних оценок. Для этого, исключая множество I_2, перейдем сразу к рассмотрению множества II_{11}, разобьем его снова на два множества, III_{111} и III_{112}, отбросим множество II_{12} и т. д. Другими словами, построим дерево вариантов, изображенное на рис. 3.8. Эта

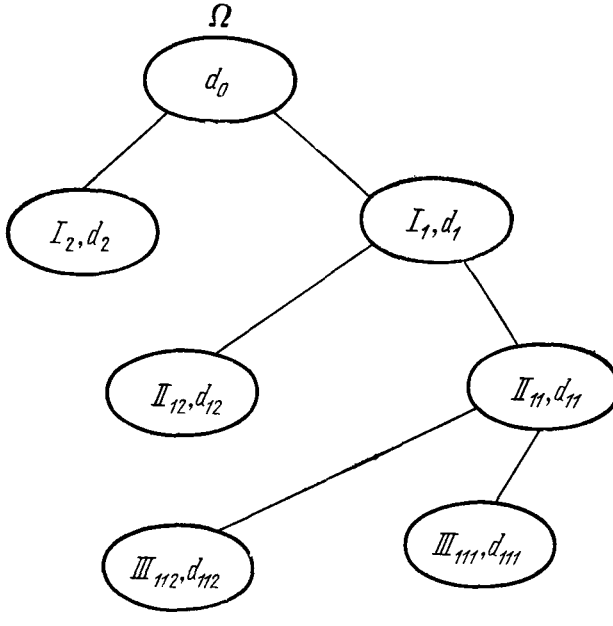


Рис 3 8

процедура нам выделит некоторое решение и некоторый путь s_1, длина которого будет l_{s_1}. Поскольку этот путь может и не быть решением задачи коммивояжера, то

$$l^* \leqslant l_{s_1}.$$

Решение l_{s_1} естественно назвать квазиоптимальным.

Итак, на этом шаге мы получили двустороннюю оценку

$$d_1 \leqslant l^* \leqslant l_{s_1}. \qquad (3.11)$$

Построением оценки (3.11) заканчивается первый шаг процесса.

На следующем шаге метода ветвей и границ мы снова находим некоторое множество путей $\Pi_{i_1 i_2}$, $i_1 = 1, 2$, $i_2 = 1, 2$, среди которых мы ожидаем найти наикратчайший, и оценку $d_{i_1 i_2}$. Среди путей, принадлежащих этому множеству, строим квазиоптимальный. Пусть его длина будет l_{s_2}. Теперь мы получим еще одну двустороннюю оценку

$$d_{i_1 i_2} \leqslant l^* \leqslant l^*_{s_2}, \tag{3.12}$$

где $l^*_{s_2} = \min \{l_{s_1}, l_{s_2}\}$.

Поскольку последовательность d_1, $d_{i_1 i_2}$, ... сходится снизу к l^*, а последовательность l_{s_1}, $l^*_{s_2}$, . . сходится к l^* сверху, то на каждом шаге мы получаем все более и более узкий диапазон для l^*. Если нам задана точность — число ε — величина интервала, внутри которого должно находиться решение, то как только разность $l^*_{s_k} - d_{i_1 \ldots i_k}$ начнет удовлетворять неравенству

$$|l^*_{s_k} - d_{i_1 \ldots i_k}| < \varepsilon, \tag{3.13}$$

мы прекращаем счет.

Построение верхних оценок требует, разумеется, дополнительной затраты времени, тем не менее эта затрата времени обычно окупается. Прежде всего, эта информация полезна сама по себе, далее, мы можем значительно сократить число итераций, используя условие (3.13). Наконец, если для каких-либо из множеств мы получаем нижнюю оценку, которая превосходит верхнюю, то это множество мы исключаем из рассмотрения раз и навсегда.

Примечание. Описанный способ построения верхних оценок — не единственный. Помимо квазиоптимальных решений можно строить локально оптимальные. Этим термином мы называем путь коммивояжера, построенный по следующему правилу. Предположим, что коммивояжер, оказавшись в городе номера i_r, уже побывал в городах $i_1, i_2, \ldots, i_k$. Тогда в качестве следующего города он выбирает город j_r, $j_r \neq i_1, \ldots, i_k$, который является ближайшим к городу i_r. Оказавшись в городе j_r, он выбирает город j_m, который является ближайшим к городу j_r и отличен от тех городов, которые коммивояжер уже посетил. Однако использование локально оптимальных решений для построения верхних оценок требует некоторых

дополнительных рассуждений и построения дополнительных программ. В то же время способ построения верхних оценок, который был изложен выше, использует практически ту же самую программу метода ветвей и границ и не требует никакого видоизменения общей схемы метода. Тем не менее, локально оптимальные решения играют важную роль в задачах, подобных задачам коммивояжера.

4. **Решение с заданной точностью.** Мы уже заметили, что особенность рассматриваемого метода состоит в том, что на каждом шаге мы производим не только сжатие множества возможных вариантов, не только отбрасываем определенное количество вариантов, но и включаем в последующее рассмотрение определенное число новых вариантов. Причем, чем больше новых вариантов нам приходится включать в рассмотрение, тем медленнее работает метод. Таким образом, если все числа c_{ij} почти равны между собой, то число вновь появившихся вариантов будет примерно тем же, что и отброшенных, и метод ветвей и границ практически будет сводиться к полному перебору всех вариантов, число которых имеет порядок $(N-1)!$. В этом случае метод будет абсолютно не эффективен.

Однако ситуация, о которой только что шла речь, будет обладать одной важной особенностью, позволяющей значительно сократить число рассматриваемых вариантов. Если все числа c_{ij} почти равны между собой, то длина пути l_s почти не зависит от выбранной последовательности городов. В пределе, когда все c_{ij} равны между собой, нам достаточно вычислить только один маршрут. Все остальные будут иметь ту же длину.

Рассмотрим снова схему последовательного анализа вариантов, изображенную на рис. 3.5, и зададимся некоторым числом $\varepsilon \in [0, 1]$.

Рассмотрим второй шаг процесса анализа вариантов и предположим, что мы уже вычислили оценку d_{11}. Согласно методу ветвей и границ, мы должны будем рассмотреть множество вариантов I_2. Если окажется, что $d_2 > d_{11}$, то на этом шаге множество I_2 мы не рассматриваем и переходим к разбиению множества вариантов II_{11}. В противном случае, когда $d_2 < d_{11}$, мы должны будем рассмотреть множество I_2 и провести его разбиение на множества II_{21} и II_{22}. Теперь же мы поступаем следующим образом. Мы будем рассматривать множество I_2 лишь в том случае,

если оценка d_2 удовлетворяет неравенству

$$d_2 \leqslant d_{11}(1-\varepsilon).$$

Точно так же мы поступим и на следующем шаге. Мы будем рассматривать множества I_2 и II_{12} лишь в том случае, когда

$$d_2 \leqslant d_{111}(1-\varepsilon),$$
$$d_{12} \leqslant d_{111}(1-\varepsilon),$$

и т. д. Описанную процедуру мы назовем ε-модернизацией метода ветвей и границ.

В результате такой процедуры мы получим не точное, а приближенное решение. В частном случае, когда $\varepsilon = 1$, оно будет квазиоптимальным. Величина ε характеризует точность решения. В том случае, когда все c_{ij} близки друг другу, ε-модификация требует очень небольшого перебора вариантов и крайне экономна. Если c_{ij} сильно отличаются друг от друга, то ε-модификация практически совпадает с основной схемой метода ветвей и границ, который в этом случае также достаточно экономен.

5. Заключение. Идея последовательного анализа и отбраковки вариантов естественна и плодотворна. Но ее реализация всегда основывается на использовании особенностей природы изучаемых задач и требует предварительного анализа — универсальных рецептов практически нет!

Мы рассмотрели два класса задач. Особенность первого — его «марковость» — позволила сформулировать принцип оптимальности, на базе которого была развита общая схема динамического программирования. Эта схема может быть описана в терминах метода ветвей и границ: на каждом шаге мы отсекали одну из ветвей, для отсеченной ветви получали однозначную оценку — принадлежит ей или нет искомый вариант. И ветвь (множество вариантов), однажды отброшенная, стиралась из памяти машины — мы к ней уже никогда больше не возвращались.

Иное дело — задача о коммивояжере. Мы строили оценки, которые не были столь категоричны. Множество отброшенных вариантов нельзя было стирать из памяти машины. Оно нам могло еще понадобиться.

Язык метода ветвей и границ весьма универсален. С его помощью может быть описан обширный круг задач.

Но это только язык, сам по себе он еще не дает инструмента анализа. Он только указывает на то, какие оценки должны быть получены. Сами же оценки — это каждый раз искусство.

Сегодня методы, использующие идеи последовательного анализа, приобрели весьма широкое распространение, и трудно переоценить их практическую значимость. Особое значение приобрели методы, получившие общее название метода последовательных расчетов, однако их изложение нас вывело бы далеко за рамки этого курса.

Приложение

ДИАЛОГОВАЯ СИСТЕМА ОПТИМИЗАЦИИ

§ 1. Принципы построения диалоговых систем

Изложенные в этой книге численные методы являются эффективным средством решения многих оптимизационных задач, возникающих в научно-исследовательских разработках, в проектно-конструкторских расчетах При использовании численных методов оптимизации для решения практических задач приходится проводить огромный объем вычислений Поэтому широкое внедрение методов оптимизации стало возможным лишь сейчас, благодаря созданию современных мощных быстродействующих электронно-вычислительных машин третьего поколения.

При работе на машинах первого и второго поколений пользователь обычно составлял одну-две стандартные программы оптимизации и пытался с их помощью решить поставленную задачу. Если ему это не удавалось, то он программировал новый метод и делал новую попытку решить ту же задачу. Появление ЭВМ третьего поколения открыло возможность по-новому организовать внешнее математическое обеспечение. Вместо стандартных программ сейчас стало реальным использовать пакеты программ, т. е. комплекс программных средств, включающих в себя библиотеку разнообразных алгоритмов для решения данного класса задач, вспомогательные программы и управляющую программу, организующую на ЭВМ решение задач с учетом их специфики. Опыт работы показывает, что использование пакетов прикладных программ значительно увеличивает возможности ЭВМ.

Создание эффективно действующих и достаточно универсальных пакетов программ оптимизации представляет собой большую научно-техническую задачу. К разработке пакетов привлечены большие коллективы ученых-математиков и специалистов по вычислительной технике. После создания первых вариантов пакетов проводится их опытная эксплуатация и в процессе решения разно-

образных практических задач накапливаются различные замечания, уточнения, которые учитываются при создании последующих версий пакетов.

Сейчас перед учеными стоят проблемы резкого повышения эффективности всего общественного производства, ускорения научно-технического прогресса. Для их успешного решения необходимо научиться создавать системы планирования и проектирования сложных комплексов, решать задачи, связанные с региональным развитием, уметь рассчитывать социальные, экономические и экологические последствия принимаемых решений.

Опыт работы по созданию таких систем показывает, что эти задачи нельзя ставить и решать как единую оптимизационную задачу. Методы оптимизации, имеющие большое значение при решении частных задач управления и проектирования, становятся, как правило, неприемлемыми при расчете управления сложными комплексами и системами. Можно указать две основные причины такого явления.

Во-первых, математические модели (т. е. математические описания) социально-экономических процессов, достаточно адекватно отражающие основные закономерности, чрезвычайно сложны. Трудности их расчета возникают прежде всего из-за высокой размерности, свойственной большинству практически важных задач (например, экономическая задача оптимизации межотраслевого баланса, техническая задача проектирования самолета и т. д.). Поэтому решение единой оптимизационной задачи в рамках этих моделей становится невозможным даже с привлечением самых мощных современных электронно-вычислительных машин.

Вторая, не менее важная причина заключается в возникновении принципиальных трудностей при формализации целей и критериев проектирования. Часто бывает так, что проектировщик, изготовляя новую конструкцию, стремится сделать многоцелевой объект, причем ему необходимо выполнить ряд требований, являющихся «внешними» по отношению к данному, проектируемому объекту. Например, конструкторы, проектирующие пассажирский самолет, стремятся создать прежде всего наиболее экономичный вариант. Вместе с тем, они должны учитывать факторы, связанные с надежностью самолета,

с созданием условий, способствующих простоте его эксплуатации, самолет должен быть приспособлен к тем взлетно-посадочным полосам, где он будет использоваться, и так далее. Как правило, выполнение «внешних» требований приводит к ухудшению основных показателей. Необходимо, вместе с тем, достичь некоторого компромисса. Решить этот вопрос в рамках математических моделей трудно, поэтому обычно в таких случаях привлекают экспертов — специалистов, обладающих содержательными знаниями о решаемой задаче. Используя опыт практической работы, эксперты приходят к некоторому соглашению. Аналогичные ситуации возникают часто при составлении программ освоения новых регионов, при разработке планов крупных строительств.

Все это заставляет в практике управления и планирования сложных социально-экономических, технических систем наряду с классическими методами оптимизации использовать такие приемы, как эвристические подходы, основанные на интуиции и опыте экспертов. Для реализации такого подхода создаются так называемые имитационные системы, в которые входят математическое описание проектируемого объекта, электронно-вычислительная машина и системное обеспечение, позволяющее группе специалистов в режиме непосредственного диалога с ЭВМ рассчитывать возможные последствия принимаемых решений, анализировать результаты и вырабатывать таким образом наилучший вариант проектируемого объекта. Электронно-вычислительные машины третьего поколения создали техническую базу для построения таких систем, позволив легко вводить в ЭВМ данные, необходимые для воспроизведения исследуемого процесса, оперативно получать из ЭВМ информацию о его течении в форме, удобной для дальнейшего анализа, непосредственно управлять исследуемым процессом. По приказу экспертов, ведущих расчеты, осуществляется изменение искомых параметров и функций, изучается реакция на них системы

Использование имитационных систем в практике проектирования сложных комплексов не только не означает отказа от решения точных оптимизационных задач, но опирается на эту технику В имитационную систему целесообразно включить пакет программ, предназначенных для поиска оптимальных решений. С его помощью можно

отыскать решение вспомогательных задач, упрощая работу с имитационной системой.

Например, в имитационную систему, созданную для проектирования самолета, можно включать блок оптимизации для отыскания силовой конструкции самолета, обладающей наименьшим весом и выдерживающей заданные нагрузки. Ясно, что уменьшение веса самолета, не идущее в ущерб его прочностным свойствам, приводит только к его улучшению. Аналогичная ситуация имеет место почти всюду в задачах машиностроения (проектирование автомобилей, кораблей, ракет и т. д.).

По мере развития численных методов теории оптимизации и более глубокого понимания сущности управляемых процессов, роль методов оптимизации будет, естественно, возрастать. Управления, получаемые в результате численного решения оптимизационных задач из вспомогательной информации, помогающей принимать решения, будут все более превращаться в основу для принятия решения.

Возникают вопросы, каким образом создать эффективно действующий пакет, предназначенный для решения широкого класса разнообразных оптимизационных задач? Какие выбрать для него методы?

В настоящее время разработано большое количество численных методов оптимизации. Математик при решении конкретной задачи стремится выбрать наилучший метод, позволяющий за наименьшее время использования электронно-вычислительной машины найти решение с заданной точностью. Но как выбрать наилучший метод, какими взять параметры в выбранном методе?

Качество численного метода характеризуется многими факторами: областью сходимости, скоростью сходимости, временем выполнения одного шага оптимизации, объемом памяти машины, необходимым для реализации метода, классом задач, решаемых методом, и т. д. Сейчас не существует, да, по-видимому, и не будет существовать наилучшего во всех отношениях универсального численного метода. Мы считаем, что не поиск универсального метода, а разумное сочетание разнообразных методов позволит с наибольшей эффективностью решать поставленные задачи. Поэтому для создания пакета оптимизации необходимо прежде всего иметь библиотеку разнообразных программ.

Все программы, входящие в библиотеку, должны быть оформлены в единых стандартах, так как только в этом случае можно будет легко осуществлять переход от одного метода решения задач к другому. Библиотека стандартизованных программ оптимизации и вспомогательных к ним программ образуют простейший пакет.

Поиск оптимального решения можно осуществлять в пакетном режиме, когда пользователь, т. е. специалист, проводящий расчеты, заранее определяет последовательность применяемых методов, задает их параметры и вводит все это в ЭВМ. Такой подход, однако, может привести к значительному объему расчетов, поскольку часто бывает трудно предугадать ход процесса оптимизации. Более целесообразным представляется проводить вычисления в диалоговом режиме, когда вычислитель в процессе расчетов, получая сведения о текущих результатах, изменяет параметры программ, осуществляет целенаправленный переход от одного метода к другому. Такой режим работы позволяет в максимальной степени использовать опыт и интуицию математика — вычислителя. Для успешной работы с диалоговой системой от математика — вычислителя требуется профессиональное знание используемых методов оптимизации, хорошее понимание свойств, специфики применяемых методов, умение правильно ориентироваться в новых ситуациях, верно использовать различные эвристические приемы. Эти гребования существенно сужáют круг пользователей.

В ряде случаев важно иметь автоматизированный пакет, содержащий библиотеку программ оптимизации, вспомогательные и управляющую программы, которые обеспечивают автоматический выбор последовательности используемых алгоритмов для решения каждой конкретной задачи. Такие пакеты необходимы в двух случаях:

если с системой работает пользователь, не являющийся специалистом в области методов оптимизации,

если процесс расчетов должен по каким-либо причинам протекать автономно, без участия человека.

Это может иметь место, например, когда эксперты используют имитационную систему для принятия решений на высоком уровне, а вспомогательные задачи оптимизации должны решаться без их участия.

Возможны и более сложные комбинированные пакеты, в которых по желанию пользователя можно применять режимы диалога и автоматической работы.

Таким образом, можно указать четыре уровня пакетов программ оптимизации:

первый — библиотека стандартизованных программ оптимизации и вспомогательных к ним программ;

второй — диалоговая система оптимизации;

третий — автоматизированный пакет оптимизации;

четвертый — комбинированный пакет оптимизации.

В настоящее время во многих организациях ведутся работы над диалоговыми системами. Это связано со многими причинами, прежде всего с появлением эффективно работающих терминальных устройств и операционных систем, позволяющих осуществить диалоговое взаимодействие пользователя с электронно-вычислительными машинами третьего поколения. Созданные первые, далеко не совершенные диалоговые системы показывают, что их использование существенно упрощает и ускоряет расчеты по сравнению с традиционными методами эксплуатации ЭВМ. Организация диалога существенно снижает непроизводительные затраты времени пользователя, дает возможность в максимальной степени использовать его опыт численных расчетов и интуицию. Наличие в системе библиотеки алгоритмов оптимизации освобождает пользователя от трудоемкой работы по программированию численных методов оптимизации. Диалоговый сервис позволяет быстро переходить от одного метода к другому.

Особенно важно, что с помощью диалоговой системы можно уточнять постановку решаемой задачи. Процесс использования системы при этом происходит следующим образом. Коллектив экспертов ставит задачу и грубо очерчивает границы области, в которой ищется решение, допуская, что в дальнейшем возможно уточнение этой области. Затем математик — вычислитель, используя диалоговый режим, отыскивает на ЭВМ решение оптимизационной задачи и передает его экспертам. Анализируя решение, эксперты проверяют, не оказалось ли найденное решение недопустимым по некоторым ограничениям, о которых они не информировали математиков, полагая, что эти ограничения несущественны. Если обнаружены некоторые нарушения, то о них сообщается системе и

производится вторичная оптимизация откорректированной задачи. Возможны и обратные действия, когда эксперты замечают, что целый ряд ограничений оказывается несущественным и их можно опустить, упростив тем самым расчеты. Математик — вычислитель вводит новую информацию в ЭВМ, производит оптимизацию и сообщает новые результаты экспертам, которые, используя подчас неформальные приемы, выбирают наиболее рациональные в каком-то смысле варианты.

Мы кратко остановимся на описании одного из первых пакетов оптимизации, разработанного в 1975 — 1976 годах в Вычислительном Центре АН СССР. Пакет представляет собой диалоговую систему оптимизации (ДИСО), предназначенную для решения задач безусловной минимизации функций многих переменных и задач нелинейного программирования. В процессе работы над этой системой были найдены основные принципы построения, функционирования и эксплуатации таких систем. Опыт практической работы с ДИСО лег в основу создания последующих, более совершенных и универсальных версий системы. В них существенно расширены возможности пользователя, реализованы разнообразные методы решения задач оптимального управления системами, описываемыми обыкновенными дифференциальными уравнениями при наличии смешанных ограничений на вектор управлений и фазовый вектор. Некоторые простейшие методы решения таких задач были затронуты выше, в главе VI.

В созданной в ВЦ АН СССР диалоговой системе оптимизации можно выделить следующие пять компонент:

1) библиотека стандартных программ оптимизации и вспомогательных к ним программ;

2) электронно-вычислительная машина (в данном случае это БЭСМ-6);

3) алфавитно-цифровое терминальное устройство типа «дисплей» с пишущей машинкой для ввода информации (в данном случае — Видеотон-340);

4) математическое обеспечение диалога;

5) пользователь, ведущий расчеты.

В соответствии со стандартными требованиями пользователь оформляет задачу оптимизации, вводит ее в ЭВМ с перфокарт, магнитной ленты, либо непосредственно с буквенно-цифровой клавиатуры. С помощью диалоговых

средств пользователь вызывает из библиотеки программ наиболее подходящие (с его точки зрения) алгоритмы, подбирает их параметры. Программное обеспечение производит склейку программ, загрузку ЭВМ. Результаты расчетов выводятся на экран дисплея (одновременно можно получать результаты и на алфавитно-цифровом печатающем устройстве). Анализируя результаты вычислений, пользователь снова принимает решения о дальнейших расчетах, получая, таким образом, возможность следить за ходом решения задачи, оперативно вмешиваться в процесс расчетов, выбирать методы, корректируя в случае необходимости их параметры и при желании даже модифицируя сами методы. ДИСО является системой с директивным входным языком. Пользователь определяет, как часто и в каком виде должны выдаваться результаты на экран дисплея, и далее, используя заранее определенные директивы, ведет расчеты.

При построении ДИСО авторы стремились выполнить следующие требования:

1) описание задач оптимизации должно быть простым и приемлемым для использования любого алгоритма из библиотеки программ;

2) систему можно легко расширять путем введения новых численных методов, постановок новых задач;

3) в системе можно использовать алгоритмы, написанные на языках высокого уровня — на алголе-60 и фортране;

4) пользователь должен иметь возможность управлять любыми формальными параметрами процедур, просто переходить от одного алгоритма к другому, оперативно менять постановки задач и алгоритмы численных расчетов;

5) для решения задач нелинейного программирования можно использовать, если в этом есть необходимость, любой метод безусловной минимизации;

6) система должна замечать по крайней мере самые простые ошибки пользователя и сообщать ему об этом (например, не определены какие-либо нужные формальные параметры процедур, с терминала введена неправильная директива и т. д.);

7) систему следует строить так, чтобы ее можно было использовать для решения задач в пакетном режиме;

8) система должна быть приспособлена для работы по предписанному сценарию, когда пользователь заранее

составляет программу вычислений в зависимости от получаемых результатов и вводит ее в ЭВМ. Расчеты проходят без непосредственного участия пользователя.

При создании ДИСО возникли противоречивые требования: с одной стороны, было бы желательно, чтобы пользователь мог как можно более существенно влиять на процесс решения задачи. Для этого следовало увеличить библиотеку программ, расширить список параметров процедур, ввести сюда все вспомогательные коэффициенты, предусмотреть как можно большее число возможных вариантов алгоритмов. Но для работы с такой системой требуется хорошая подготовка пользователя, он должен знать все тонкости методов, глубоко разбираться в библиотеке программ. С другой стороны, для того чтобы расширить круг пользователей, следует упростить работу с ДИСО, автоматизировать процесс принятия решений, ограничить набор алгоритмов. Другая трудность связана с детализацией человеко-машинного взаимодействия. Диалог можно оформить так, чтобы система задавала человеку весьма подробные вопросы об организации расчетов и давала комментарии и разъяснения по этим вопросам. Это упростило бы работу начинающего пользователя, но удлиннило бы время расчетов опытного пользователя. В созданном варианте ДИСО авторы пришли к некоторому, временно́му компромиссу.

§ 2. Библиотека программ решения задач безусловной минимизации

В библиотеку ДИСО входят: программы оптимизации, набор стандартных программ вычисления с разными степенями точности градиентов функций и матриц вторых производных, вспомогательные программы вывода информации. Вся библиотека написана на языке алгол-60 и создана на основе модульного принципа. Такая организация допускает взаимозаменяемость модулей, изменение одного из них не приводит к изменению других. Это позволяет, сохраняя в целом систему, производить постепенную переработку отдельных блоков, включать новые и исключать неудачные варианты.

Программы, предназначенные для решения задач оптимизации, имеют стандартную форму записи и унифициро-

ванную структуру. При выборе методов авторы стремились к наилучшему использованию специфики каждого из методов и к созданию набора алгоритмов, наиболее дополняющих друг друга. Многие из этих методов описаны в главах II и V, вместе с тем было использовано несколько новых методов, взятых из различных книг и журнальных статей.

Каждая программа в библиотеке имеет свой заголовок. Программы, реализующие методы безусловной минимизации функций многих переменных, имеют заголовки, начинающиеся с буквы А, далее идет цифра, предписанная для каждого конкретного метода. Аналогично в случае методов решения задач нелинейного программирования в начале пишется буква С, далее указывается некоторая цифра.

Для нахождения безусловного минимума созданы программы, реализующие следующие методы:

А1, А11 — варианты покоординатного спуска (глава II, § 1, п. 7);

А22, А23 — варианты наискорейшего спуска (глава II, § 1, п. 3);

А2, А21 — варианты наискорейшего спуска Полака (см. [8]);

А3, А31, А32 — метод сопряженных градиентов Флетчера — Ривса и его модификации (глава II, § 3, п. 3);

А4 — модифицированный вариант метода Ньютона (глава II, § 2,);

А5 — метод Пауэлла (см. [1]);

А6, А7 — методы случайного поиска;

А8 — метод Хука и Дживса (см. [11]);

А9 — симплекс-метод Нилдера и Мида (см. [11]).

Ниже мы приведем протоколы расчетов, проведенных в диалоговом режиме на ЭВМ с использованием этой библиотеки. Так как все алгоритмы записаны на языке алгол-60, нам придется внести некоторые изменения в формулировки задач, рассмотренных в главах II, V, с тем, чтобы максимально приблизить их к используемым программам и правилам работы с ДИСО.

Задача безусловной минимизации функции многих переменных состоит в отыскании

$$\min_x F(x), \tag{2.1}$$

где $x=[x^1, \ldots, x^n]$ суть n-мерный вектор, $F(x)$ — по крайней мере непрерывная функция x. Численные методы решения (2.1) приводят к некоторому итеративному процессу вида

$$x_{k+1}=x_k+c_k\varphi(x_k, k) \quad k=0, 1, 2, \ldots \qquad (2.2)$$

где c_k — шаг спуска. Задание функции φ и последовательности c_k определяют метод оптимизации.

Программы оформлены в соответствии с требованиями языка алгол-60. Каждая программа представляет собой процедуру общего типа с заголовком

AI(X, Y, G1, INP, RP, F, AGR, AGS, ABB).

Здесь I — номер алгоритма. Смысл формальных параметров процедуры следующий:

X — массив размером [1 : N], при входе в процедуру ему присваиваются координаты начальной точки x_0, после окончания k-го шага процесса (2.2) в этом массиве хранится вектор x_k;

Y — реальное число; всюду $Y=F(x)$; после выполнения k шагов итераций по методу безусловной минимизации $Y=F(x_k)$;

G1 — массив [1 : N]; определяет градиент функции $F(x)$ в текущей точке x;

INP — масеив [1 : 100] целых чисел; является информационным массивом, описание его части, относящейся к задачам безусловной минимизации, дано ниже;

RP — массив [1 : 100]; является информационным массивом, описание его части, относящейся к задачам безусловной минимизации, дано ниже;

F — реальная процедура с одним формальным параметром X, определяющая для каждого вектора X значение минимизируемой функции F(X);

AGR — процедура общего типа с формальными параметрами F, X, Y, H, N, G; процедура предназначена для вычисления градиента G функции F в N-мерной точке X; предполагается, что значение функции F в точке X известно и приписано Y; H — величина, пропорциональная шагу вычисления производной;

AGS — процедура общего типа с формальными параметрами F, X, Y, G, H, N, GS, процедура предназначена для вычисления матрицы размера N×N вторых произ-

водных GS функции F в N-мерной точке X при условии, что известно Y-значение функции F в точке X и ее градиент G в той же точке; H — величина, пропорциональная шагу вычисления вторых производных;

ABB — процедура вывода выходной информации; стандартный вариант такой программы хранится в библиотеке ДИСО и оформлен в виде процедуры ABB1; пользователь может использовать эту процедуру, либо в соответствии со стандартными требованиями написать свою программу, ориентированную на решение конкретной задачи.

Приведем описание лишь той части информационных массивов INP и RP, которая существенна для понимания протоколов расчетов. Использовались следующие идентификаторы для обозначения отдельных элементов указанных массивов:

D: = INP [54], N: = INP [55], HS: = INP [52],
HP: = INP [53], C: = RP [51], E: = RP [52],
H: = RP [53], E1: = RP [54].

Разъясним смысл этих параметров:

C — начальный шаг спуска (в (2.2) это c_0);

D — максимально возможное количество шагов, которое делается по итеративной схеме (2.2);

E — точность расчетов, допускающая более раннее, чем через D шагов, окончание итераций; обычно расчеты прекращаются, если $|F(x_{k-1}) - F(x_k)| < E$; в градиентных методах используется часто другое условие, $\|F_x(x_k)\| < E$; выбор того или иного условия производит составитель метода;

H — величина, пропорциональная шагу, используемому для численного отыскания производных;

HS — номер шага итерации, начиная с которой работает процедура вывода информации ABB;

HP — шаг, с которым включается процедура ABB, начиная с шага HS (если HS = 0, HP = 1, то процедура ABB работает на каждом шаге);

E1 — вспомогательный параметр, играющий разную роль в каждой из программ.

Заголовки всех указанных процедур были составлены с некоторым «запасом»: далеко не во всех программах используется процедура AGR, тем более AGS. Такое рас-

ширение заголовка было сделано, чтобы обеспечить модульную структуру.

Перед началом расчетов пользователь вводит в систему процедуру F, и, если он этого пожелает, свои индивидуальные варианты процедур AGR, AGS, ABB. Пользователь может ввести также начальную точку x_0, сообщив об этом системе.

В заголовках процедур, реализующих методы безусловной минимизации, входными параметрами являются:

X, INP, RP, F, AGR, AGS, ABB.

Выходные параметры следующие:

X, Y, G1,

причем G1 определяется лишь в тех методах, в которых вычисляются производные $F(x)$.

§ 3. Библиотека программ решения задач нелинейного программирования

В библиотеку вошли программы, реализующие следующие методы:

C2, C3 — варианты методов внутренних штрафных функций (глава V, § 2, п. 5);

C4, C41 — варианты внешних штрафных функций (глава V, § 2, п. 4);

C13 — метод возможных направлений Г. Зойтендейка (глава V, § 1, п. 2);

C5 — метод с оценкой критерия (глава V, § 2, п. 7; для регулировки использовалась формула Моррисона (2.24));

C7 — метод с модифицированной функцией Лагранжа (глава V, § 2, п. 8);

C9 — метод простой итерации (см. [1]) *);

*) [1] Голиков А. И., Евтушенко Ю. Г. Об одном классе методов решения задач нелинейного программирования. ДАН СССР, 1978, 239, № 3, 519—522.

[2] Евтушенко Ю. Г. Численные методы нелинейного программирования. ДАН СССР. 1975, 221, № 5, 1016—1019

[3] Евтушенко Ю. Г. Численные методы решения задач нелинейного программирования. Журнал вычислительной математики и математической физики. 1976, 16, № 2, 307—324.

С12 — метод простой итерации с использованием барьерных функций (см. [1]);

С8 — метод Ньютона (см. [2, 3]);

С61, С62 — варианты релаксационного метода, описанного в [4], [5] *).

В дальнейшем будет удобно вместо длинных наименований методов использовать названия программ, реализующих эти методы. В методах С7, С9, С12 итеративный процесс ведется по двойственным переменным, изменение основной переменной происходит в результате решения вспомогательной задачи безусловной минимизации. Поэтому в приведенном ниже каталоге методов нелинейного программирования эти методы названы двойственными.

Задача нелинейного программирования состоит в отыскании минимума функций многих переменных

$$\min f^0(x), \tag{3.1}$$

где $x=[x^1, \ldots, x^n]$ есть n-мерный вектор, при наличии ограничений типа равенств

$$f^i(x)=0, \quad i \in [1:l]$$

и неравенств

$$f^j(x) \leqslant 0, \quad j \in [l+1:m].$$

Дадим краткое описание методов С9, С12, С8, С61, С62, не приведенных в предыдущих главах книги. Изложение начнем с модификации метода Ньютона С8, приспособленной для решения задач нелинейного программирования.

Составим модификацию функции Лагранжа вида

$$N(x, p, w)=f^0(x)+\sum_{i=1}^{l} f^i(x)\, p^i+\sum_{j=l+1}^{m} f^j(x)\,(w^j)^2;$$

здесь введены векторы $p=[p^1, \ldots, p^l]$ и $w=[w^{l+1}, \ldots, w^m]$. Несложно показать, что для приведенной функции N остается справедливой теорема Куна — Таккера, доказанная в главе IV. Поэтому, если существуют векторы x_*,

*) [4] Евтушенко Ю. Г. Два численных метода решения задач нелинейного программирования. ДАН СССР, 1974, 215, № 1, 38—40.

[5] Евтушенко Ю. Г., Жадан В. Г. Релаксационный метод решения задач нелинейного программирования. Журнал вычислительной математики и математической физики, 1977, 17, № 4, 890—904.

p_*, w_*, являющиеся седловыми точками в задаче отыскания безусловного седла

$$\max_p \max_w \min_x N(x, p, w), \tag{3.2}$$

то вектор x_* будет решением исходной задачи (3.1). Необходимые условия седла состоят в выполнении следующих условий стационарности:

$$\frac{dN(x_*, p_*, w_*)}{dx} = 0, \quad \frac{dN(x_*, p_*, w_*)}{dp^i} = f^i(x_*) = 0 \tag{3.3}$$

$$\frac{dN(x_*, p_*, w_*)}{dw^j} = 2w^j_* f^j(x_*) = 0.$$

Здесь i, j — целые, причем $i \in [1 : l]$, $j \in [l+1 : m]$. Если строить функцию Лагранжа в традиционном виде, то

$$L(x, p) = f^0(x) + \sum_{i=1}^{m} f^i(x) p^i.$$

Очевидно, что первые l координат вектора p совпадают с координатами вектора p, введенного при определении функции N. Остальные координаты связаны соотношением

$$p^j = (w^j)^2, \quad j \in [l+1 : m].$$

Здесь тоже можно поставить задачу об отыскании седловой точки

$$\max_{p \in I} \min_x L(x, p),$$

где $I = \{p : p^j \geqslant 0 \text{ для } l < j \leqslant m\}$.

Решение этой задачи существенно сложнее, чем решение (3.2), где отыскивается безусловное седло, сложнее выписываются и необходимые условия минимакса (достаточно сравнить условия (3.3) с условиями (3.17) из главы IV). Для решения (3.2) можно использовать широкий класс численных методов отыскания седловых точек. Воспользуемся, например, методом Ньютона. Объединим векторы x, p, w единым символом z, условия стационарности (3.3) перепишем в компактной форме

$$N_z(z_*) = 0, \quad \text{где} \quad N(z_*) = N(x_*, p_*, w_*).$$

Метод Ньютона будет иметь вид

$$z_{k+1} = z_k - \alpha_k N_{zz}(z_k) N_z(z_k). \tag{3.4}$$

Здесь шаг c_k обычно равен единице Однако в ряде случаев удобно дробить шаг c_k, требуя выполнения условия невозрастания нормы N_z:

$$\| N_z(z_{k+1}) \| \leqslant \| N_z(z_k) \|.$$

Такой прием часто позволяет избежать расходимости метода, что бывает в тех случаях, когда начальная точка z_0 находится вдали от решения.

Схема (3.4) широко используется для решения задач нелинейного программирования, позволяя достичь квадратичной скорости сходимости — необычно высокой для задач этого класса. Ниже даны результаты расчетов, полученные с помощью этого метода, иллюстрирующие его высокую эффективность.

Существенным недостатком метода Ньютона является необходимость вычислять, обращать и хранить в памяти матрицу N_{zz}. При решении многих практических задач размерность этой матрицы часто составляет несколько сотен. Матрица такого размера не помещается в оперативную память многих ЭВМ. Поэтому от этого метода приходится отказываться. Наиболее подходящим инструментом решения таких задач оказались методы С7, С9, С12. Скорость их сходимости выше, чем в методах штрафов и несущественно от них отличающихся вариантов метода с оценкой критериев. Метод С7 был описан выше, в главе V. Поэтому кратко остановимся на методах С9 и С12.

Упростим на время задачу (3.1), отбросив в ней ограничения типа неравенств. Составим обобщенную функцию Лагранжа

$$H(x, p) = f^0(x) + \sum_{i=1}^{l} \varphi(f^i(x), p^i).$$

Здесь введена некоторая, пока не определенная функция двух скалярных аргументов $\varphi(a, b)$, обладающая непрерывными частными производными. Обозначим через

$$\varphi'(a, b) = \frac{d\varphi(a, b)}{da}.$$

Рассмотрим вспомогательную задачу безусловной минимизации обобщенной функции Лагранжа:

$$H(x(p),\ p) = \min_x \left[f^0(x) + \sum_{i=1}^{l} \varphi(f^i(x),\ p^i) \right]. \qquad (3.5)$$

Предполагаем, что эта задача имеет решение $x = x(p)$. Тогда необходимое условие минимума состоит в равенстве нулю первой производной H по x:

$$H_x(x(p),\ p) =$$

$$= f^0_x(x(p)) + \sum_{i=1}^{l} \varphi'(f^i(x(p)),\ p^i) f^i_x(x(p)) = 0. \qquad (3.6)$$

Считаем, что в задаче (3.1) существуют векторы x_*, p_*, для которых выполнены необходимые условия минимума Куна – Таккера:

$$f^0_x(x_*) + \sum_{i=1}^{l} f^i_x(x_*) p^i_* = 0, \quad f^i(x_*) = 0, \quad i \in [1:l]. \qquad (3.7)$$

Сравнение (3.6) и (3 7) наводит на мысль попытаться найти корни уравнений

$$\varphi'(f^i(x(p)),\ p^i) = p^i, \quad i \in [1:l]. \qquad (3.8)$$

Если $\tilde{p} = [\tilde{p}^1, \ldots, \tilde{p}^l]$ – их решения, $x(\tilde{p}) = \tilde{x}$ и точка $\tilde{x}$ – допустимая, то (3.6) совпадает с первым условием Куна – Таккера. Наложим на φ требование, гарантирующее допустимость точки $\tilde{x}$. Оно заключается в следующем:

Условие B1. Функция $\varphi(a, b)$ непрерывно дифференцируема, для любых действительных чисел b имеет место $\varphi'(0, b) = b$ и если $a \neq 0$, то $\varphi'(a, b) \neq b$.

При выполнении этого условия, из того, что $\tilde{p}$ – решение (3.8), автоматически следует, что векторы $\tilde{x}$, $\tilde{p}$ удовлетворяют условиям Куна – Таккера. Решение исходной задачи (3.1) свелось к отысканию действительных корней системы (3.8). Благодаря этому для решения (3.1) можно использовать весь богатый арсенал существующих методов решения систем нелинейных уравнений. Метод простой итерации, в частности, приводит к схеме:

$$p^i_{k+1} = \varphi'(f^i(x(p_k)),\ p^i_k), \quad i \in [1:l]. \qquad (3.9)$$

Возможны другие, самые разнообразные варианты метода.

Метод простой итерации сходится лишь при выполнении определенных условий Перенося их на рассматриваемый случай, придем к еще одному дополнительному требованию, которое следует наложить на функцию φ.

Условие B2. Если x_*, p_* удовлетворяют (3.7), то

$$\frac{d^2\varphi(a^l, p_*^l)}{da^2} > 0, \quad \frac{d^2\varphi(a^l, p_*^l)}{db\,da} = 1;$$

здесь $i \in [1 : l]$, $a^l = f^l(x(p_*))$.

Можно привести большое число функций, удовлетворяющих B1 и B2, например, класс функций вида $\varphi(a, b) = a(b - \alpha'(0)) + \alpha(a)$, где $\alpha(a)$ — строго выпуклая функция от a. В качестве φ, таким образом, можно взять

$$\varphi^1(a, b) = ab + a^2/2,$$
$$\varphi^2(a, b) = ab + \operatorname{ch} a,$$
$$\varphi^3(a, b) = a(b - 1) + \exp a.$$

В формулы (3.5) и (3.9) в качестве $\varphi(a, b)$ подставим $\varphi(\tau a, b)/\tau$, где τ — некоторый положительный параметр. Тогда при выполнении стандартных достаточных условий экстремума для задач нелинейного программирования метод простой итерации порождает последовательность двойственных переменных $\{p_k\}$ такую, что последовательность $\{x(p_k)\}$ локально сходится к решению (3.1), если имеют место B1, B2 и τ достаточно велико.

Рассмотрим задачу (3.1) в случае, когда присутствуют ограничения только типа неравенства. Функцию Лагранжа строим аддитивным образом, положив

$$H(x, p) = f^0(x) + \sum_{j=l+1}^{m} \psi(f^j(x), p^j).$$

Условия Куна — Таккера в данном случае имеют вид

$$f_x^0(x_*) + \sum_{j=l+1}^{m} f_x^j(x_*)\, p_*^j = 0, \quad f^j(x_*) \leqslant 0,$$
$$f^j(x_*)\, p_*^j = 0, \quad p_*^j \geqslant 0, \qquad (3.10)$$

где $j \in [l+1 : m]$.

Аналогом (3.5) будет следующая задача:

$$H(x(p), p) = \min_x \left[f^0(x) + \sum_{j=l+1}^{m} \psi(f^j(x), p^j) \right].$$

Проводя рассуждения, близкие к приведенным, приходим к системе

$$p^j = \psi'(f^j(x(p)),\ p^j),\quad j \in [l+1:m].$$

Обозначим через $P(a)$ множество действительных, неотрицательных решений уравнения $b = \psi'(a,\ b)$. Роль условий B1, B2 будут играть следующие:

Условие B3. При $a > 0$ множество $P(a)$ пусто; при $a < 0$ множество $P(a)$ состоит только из нуля, если $a \geqslant 0$, то $a \in P(0)$; для любых a и любых $b \geqslant 0$ имеет место неравенство

$$\psi'(a,\ b) \geqslant 0. \tag{3.11}$$

Условие B4. Пусть векторы x_*, p_* удовлетворяют (3.10), тогда

если $f^j(x_*) = 0$, то

$$\frac{d^2\psi(f^j(x_*),\ p^j_*)}{da^2} > 0,\quad \frac{d^2\psi(f^j(x_*),\ p^j_*)}{da\,db} = 1;$$

если $f^j(x_*) < 0$, то

$$\frac{d^2\psi(f^j(x_*),\ p^j_*)}{da^2} = 0,\quad \frac{d^2\psi(f^j(x_*),\ p^j_*)}{da\,db} < 1.$$

Требованиям B3 и B4 удовлетворяют, например, следующие две функции:

$$\psi^1(a,\ b) = \gamma(a_+) + be^a,$$

$$\psi^2(a,\ b) = \gamma(a_+) + b\begin{cases} 1 + a + a^2 + a^3, & \text{если } a \geqslant 0, \\ 1/(1-a), & \text{если } a \leqslant 0. \end{cases}$$

Здесь $a_+ = \max[0,\ a]$, $\gamma(a)$ — достаточно гладкая функция такая, что $\gamma(0) = \gamma'(0) = 0$; если $a > 0$, то $\gamma(a) > 0$, $\gamma'(a) > 0$ (например, $\gamma(a) = a^4$).

Метод простой итерации состоит в следующем:

$$p^j_{k+1} = \psi'(f^j(x(p_k)),\ p^j_k),\quad j \in [l+1:m].$$

В качестве $\psi(a,\ b)$ подставим $\psi(\tau a,\ b)/\tau$. Согласно (3.11), если $p_0 > 0$, то в процессе итераций всегда будет $p_k \geqslant 0$ и можно показать, что все предельные точки последовательностей $x(p_k)$ и p_k удовлетворяют условиям Куна — Таккера (3.10).

Очевидно, что описанные методы переносятся на общий случай задачи (3.1), когда одновременно присутствуют

ограничения типа равенств и неравенств В методе С9 ограничения типа равенств учитывались с помощью функции φ^1, ограничения типа неравенств — с помощью ψ^2.

В ряде решенных задач метод С9 дал лучшие результаты, чем С7, благодаря тому, что функция H была более гладкой. Это особенно важно было в тех задачах, в которых, для безусловной минимизации использовались методы, требующие высокой степени гладкости минимизируемой функции, например, метод сопряженных градиентов Флетчера — Ривса.

В некоторых задачах бывает так, что функции, определяющие ограничения типа равенства, не определены, если нарушено условие:

$$|f^i(x)| < g, \quad i \in [1:l],$$

и ограничения типа неравенства могут быть вычислены, только если

$$f^j(x) < g, \quad j \in [l+1:m],$$

причем известна точка x_0, удовлетворяющая указанным условиям. В этом случае можно модифицировать функцию Лагранжа, положив

$$H(x, p) = f^0(x) + \sum_{i=1}^{l} [f^i(x)\, p^i + \tau g^2/(g^2 - (f^i(x))^2)] + \\ + \sum_{j=l+1}^{m} \left[\frac{1}{\tau}\, p^j e^{f^j(x)} + \gamma\,(\tau f^j_+(x))/(g - f^j(x))\right].$$

Несложно показать, что условия В1 — В4 выполняются на этих множествах, причем $H \to \infty$, если x приближается к границе этой области. Таким образом, последовательность точек $x(p)$, полученных из безусловной минимизации H, остается внутри требуемой области. Так модифицированная функция Лагранжа была использована в методе С12.

Перейдем к методам С6, С61, С62. Их описание проще всего проводить на основе непрерывного варианта. Такой прием весьма распространен при исследовании сходимости. Как правило, анализ непрерывных вариантов методов (если таковые существуют) много проще, чем анализ дискретных аналогов, и часто они яснее объясняют суть метода.

Рассмотрим вначале случай, когда в (3 1) присутствуют ограничения только типа равенств. Составим функцию

$$L(x,\ p) = f^0(x) + \sum_{i=1}^{l} f^i(x)\, p^i.$$

Введем следующую систему обыкновенных дифференциальных уравнений.

$$\frac{dx}{dt} = -L_x(x,\ p). \tag{3.12}$$

Функцию $p(t)$ определим таким образом, чтобы вдоль решений $x(t)$ этой системы все функции $f^i(x(t))$ сохраняли постоянное значение:

$$\frac{df^i(x(t))}{dt} = -\sum_{s=1}^{n} \frac{\partial f^i}{\partial x^s}\frac{dx^s}{dt} = 0.$$

Отсюда определим вектор p, подставим его выражение в правую часть системы (3 12), получим

$$\begin{gathered} \frac{dx}{dt} = -[f_x^1(x) + G_x p], \\ Ap + G_x^T f_x^0 = 0, \quad A = G_x^T G_x - D(G). \end{gathered} \tag{3.13}$$

Здесь G_x — матрица $n \times l$, $(i,\ j)$ элемент которой равен $\partial f^j(x)/\partial x^i$, $f_x^0(x)$ — матрица-столбец $n \times 1$, i-й элемент которой равен $\partial f^0(x)/\partial x^i$. Символ $D(G)$ обозначает диагональную матрицу, у которой i-й элемент есть i-я координата вектора G, $G = (f^1(x),\ f^2(x),\ \ldots,\ f^l(x))$.

Несложно показать, что производная функция $f^0(x)$ в силу полученной системы отрицательна. Поэтому, если начальная точка x_0 допустима, то и все точки $x(x_0,\ t)$ решения системы (3.13) также допустимы и вдоль траекторий (3.13) функция $f^0(x(x_0,\ t))$ монотонно убывает. Решение задачи Коши (3.13) при $t \to \infty$ сходится к некоторому локальному решению задачи (3.1).

Метод обобщается и на тот случай, когда в (3.1) существуют ограничения типа равенств и неравенств. В этом случае $G(x)$ считаем m-мерной вектор-функцией, у которой i-я координата есть $f^i(x)$. Для работы метода требуется, чтобы начальная точка x_0 принадлежала множеству X_0:

$$X_0 = \{x : f^1(x) = \ldots = f^l(x) = 0,\quad f^{l+1}(x) < 0,\quad \ldots,\ f^m(x) < 0\}.$$

Тогда вся траектория $x(x_0, t)$ будет принадлежать допустимому множеству, $f^0(x(x_0, t))$ будет монотонно убывающей функцией t. При определенных предположениях решение $x(x_0, t)$ будет сходиться к некоторому локальному решению (3.1).

При численной реализации метода уравнение (3.13) интегрируется по схеме Эйлера с шагом c. Очевидно, что при достаточно малых значениях c дискретный вариант метода будет близок к непрерывному. Метод особенно эффективен в тех случаях, когда ограничения типа равенств линейные, так как в этом случае ограничения типа равенств сохраняют постоянное значение в дискретном варианте при любых c. В общем случае приходится совершать регулировку шага, чтобы не сильно нарушалось это условие, метод сохранял свойство релаксационности и вектор x не выходил из допустимого множества. В программе С6 шаг c постоянный, в С61, С62 предусмотрена регулировка шага.

Программы, предназначенные для решения (3.1), оформлены в виде процедур с заголовками

CI (X, Y, P, INP, RP, F, CGR, CGS, CBB).

Здесь I — номер метода. Разъясним формальные параметры процедуры:

X — массив [1 : N]; совпадает с вектором x в задаче (3.1); в начале расчетов вектор x_0 помещается в X; после проделанных расчетов в массиве хранится результирующий вектор x;

Y — массив [0 : M]; $Y[i] := f^i(x)$ для $i \in [0 : M]$;

P — массив [1 : M]; является вектором двойственных переменных; пользователь задает их перед обращением к процедуре; в этом же массиве содержатся последующие значения вектора двойственных переменных;

INP — массив [1 : 100] целых чисел; является информационным массивом, описание его части, относящейся к НЛП, дано ниже;

RP — массив [1 : 100]; является информационным массивом, описание его части, относящейся к НЛП, будет дано ниже;

F — процедура общего типа с тремя формальными параметрами: F (X, Y, K); процедура определяет описанный

выше весь вектор Y, если $K<0$, в противном случае определяется $Y[K]=f^K(x)$.

CGR — процедура общего типа с формальными параметрами F, X, Y, H, N, GR, K; процедура предназначена для вычисления градиента GR размером [1 : N] функции F в точке X, если известно значение Y, шаг вычисления производной пропорционален H;

CGS — процедура общего типа для вычисления матриц вторых производных функции F;

CBB — процедура вывода информации; стандартный вариант этой процедуры хранится в архиве ДИСО с заголовком CBB1.

Использовались следующие идентификаторы для элементов информационных массивов:

D := INP [34], N := INP [35], L := INP [36],
M := INP [37], HS := INP [32], HP := INP [33],
C := RP [31], E := RP [32], H := RP [33],
A := RP [35], Z := RP [40], R := RP [37].

Смысл большинства из этих идентификаторов тот же, что и для методов безусловной минимизации:

C — начальный шаг спуска в итеративном методе решения (3.1);

D — максимально возможное количество шагов, которое делается по итеративной схеме;

E — точность расчетов, допускающая более ранний, чем через D шагов, выход из программы решения задачи (3.1);

H — величина, пропорциональная шагу, используемому для численного отыскания производных функций;

HS — номер шага итерации, начиная с которой работает процедура вывода информации CBB;

HP — шаг, с которым включается процедура CBB, начиная с шага HS (если HS = 0, HP = 1, то процедура CBB работает на каждом шаге);

L — количество ограничений типа равенства;

M — полное количество ограничений;

A — величина начального коэффициента функции штрафа;

Z — величина, на которую умножается коэффициент штрафа после осуществления каждого шага;

B — величина, на которую уменьшается точность минимизации функции штрафа после каждого шага;

E1 — дополнительный параметр; его смысл разный в каждой из программ;

N — размерность вектора X.

Перед началом счета пользователь вводит в систему процедуры для вычисления функций $f^i(x)$, определяет фактические значения идентификаторов N, M, L. Пользователь может определить начальную точку и указать, какие методы безусловной минимизации он предполагает использовать, если в этом возникнет необходимость. Кроме того, пользователь при желании может задать специальные процедуры вычисления градиентов и матриц вторых производных, а также свою процедуру вывода. Вместо этого можно указать, какие программы из библиотеки будут выполнять эти операции. Если пользователь не делает специальных указаний, то при решении задачи нелинейного программирования информация о решении вспомогательной задачи безусловной минимизации не выводится, необходимые параметры этих методов формируются автоматически на основании тех параметров, которые заданы пользователем для задачи нелинейного программирования.

§ 4. Примеры работы с ДИСО

В качестве программной базы ДИСО взята система Пульт — БЭСМ-6, разработанная в ВЦ АН СССР, а в качестве инструмента для написания ДИСО — метапроцессор системы лорд. С помощью лорда была составлена процедурно-ориентированная система, оперирующая в рамках системы Диспак методами оптимизации, процедурами-функциями и другими модулями, написанными на языке алгол-60.

Диалог был организован по-разному в случае задач безусловной минимизации (БМ) и нелинейного программирования (НЛП). В первом случае ДИСО «ведет» пользователя, задавая ему вопросы, на которые он отвечает, используя определенный словарь. В случае НЛП система запрашивает параметры метода, весь дальнейший диалог ведет пользователь. В первом случае для пользователя могут показаться несколько утомительными многочисленные вопросы, однако опыт численных расчетов показал,

что эта форма организации диалога предпочтительнее, так как она дисциплинирует пользователя, не позволяя ему забывать о весьма существенных директивах. Например, после выполнения расчетов по каждому из методов пользователь должен решать: считать ли успешными сделанные расчеты и взять ли полученную точку в качестве начальной или нет. В случае БМ система задает этот вопрос пользователю, в случае НЛП пользователь должен сам дать команду о том, что полученную точку следует взять за начальную при последующих расчетах. Однако пользователи часто забывают о необходимости такой команды, в результате чего приходится делать повторные вычисления. Опыт работы с ДИСО показывает, что в системе следует предусмотреть дублирующий вывод введенной пользователем информации для контроля ошибок, возникающих иногда из-за сбоя аппаратуры, но чаще из-за небрежности пользователя.

Опыт практических расчетов показал, что целесообразно вводить вектор (массив) масштабирующих коэффициентов $v[0:M]$. При решении задач НЛП вместо вычисления $f^i(x)$ определяется величина

$$f^i(x) \cdot v[i].$$

Пользователь по своему усмотрению может изменять компоненты вектора v, усиливая или ослабляя, таким образом, влияние отдельных ограничений. Это особенно важно в начале расчетов, в тех случаях, когда ограничения плохо «подогнаны». С точки зрения постановки задачи (3.1) не важно, какое взять i-е ограничение — равным $f^i(x)$ или $10^6 \cdot f^i(x)$, однако для вычислительного процесса разница будет весьма существенной. В начале все координаты вектора v полагаются равными единице, пользователь может их изменять в процессе расчетов.

Приведем четыре примера решения задач. Примеры были взяты специально простыми, чтобы не переполнять иллюстрацию работы ДИСО излишним цифровым материалом. В качестве задач БМ использована задача Била о нахождении минимума функции

$$F(x) = \sum_{i=1}^{3} [g_i - x^{(1)}(1 - (x^{(2)})^i)]^2,$$

$$g_1 = 1.5, \quad g_2 = 2.25, \quad g_3 = 2.625,$$

и задачи Розенброка об отыскании минимума

$$F(x) = 100\,[x^{(2)} - (x^{(1)})^2] + (1 - x^{(1)})^2.$$

Эти функции приведены в книге [1]. В обоих случаях наименьшее значение функций равно нулю. Первая функция была введена в ДИСО под названием БИЛ, вторая — с названием РОЗЕНБРОК. Минимум в первом случае достигается в точке $x^{(1)} = 3$, $x^{(2)} = .5$, во втором — в точке $x^{(1)} = x^{(2)} = 1$.

Для НЛП в качестве теста использовалась задача, в которой

$$\mathrm{N} = 3,\ \mathrm{L} = 1,\ \mathrm{M} = 5,$$
$$f^0(x) = [x^{(1)} + 3x^{(2)} + x^{(3)}]^2 + 4\,[x^{(1)} - x^{(2)}]^2 - 1.8310995,$$
$$f^1(x) = x^{(1)} + x^{(2)} + x^{(3)} - 1,\ f^2(x) = -x^{(1)},\ f^3(x) = -x^{(2)},$$
$$f^4(x) = -x^{(3)},\ f^5(x) = 3 - 4x^{(3)} - 6x^{(2)} + [x^{(1)}]^3. \qquad (4.1)$$

Минимальное значение функции $f^0(x)$ на допустимом множестве приближенно равно нулю. Заголовок процедуры, описывающей данную задачу, был Ф1.

В приведенных ниже протоколах диалога использовались две процедуры вывода АВВ1 и СВВ1. Процедура АВВ1 выводила результаты решения задачи безусловной минимизации. Вначале указывалась величина К — номер шага итерационного процесса (2.2), далее следовал текущий вектор X и Y — значение минимизируемой функции в точке X. Процедура СВВ1 использовалась в случае задач нелинейного программирования. Вывод начинался с указания К — номера шага итеративного процесса, далее следовало содержимое вектора Y, текущий вектор X и вектор двойственных переменных P.

В конце расчетов печаталась величина СА — количество обращений к вычислению минимизируемой функции. В случае задач нелинейного программирования СА есть количество обращений к процедуре, определяющей функции $f^i(x)$. Первые и вторые производные функций всюду считались численно. Дефис в начале строки указывает, что содержимое строки — информация, сообщаемая системе пользователем.

Пример 1. Решалась задача Била. Пользователь не воспользовался тем, что для этой задачи сравнительно легко можно получить аналитические формулы и записать

программы для вычисления первых и вторых производных. Вместо этого он вызвал стандартные процедуры из библиотеки программ для вычисления градиента AGR1, матрицы вторых производных AGS1 и использовал стандартную процедуру вывода ABB1. В качестве начальной точки была взята точка с координатами — 2, — 2. Далее ДИСО «вела» пользователя, задавая ему вопросы, необходимые для реализации диалога. Пользователь согласился вывести каталог методов безусловной минимизации и выбрал для расчетов простейший метод Хука и Дживса A8, в котором направление минимизации полностью определяется на основании последовательных вычислений целевой функции. В методе вначале производится покоординатный спуск (осуществляя «исследующий поиск»), затем происходит одновременное движение по всем координатам в том направлении, по которому удалось сдвинуться во время выполнения первого этапа (т. е. осуществляется «поиск по образцу»).

Пользователь определил основные параметры метода, положив

$$C=1,\ E=.00001,\ H=.5,\ D=4,\ HS=0,\ HP=1,\ E1=1.$$

Система повторила принятую цифровую информацию. Заметим, что шаг вычисления производной H = .5 пользователем был задан неудачно, так как погрешности вычисления производных в таком случае будут чрезвычайно большими. Однако в данном методе это было несущественно, так как здесь производные не считались. Система напечатала текст: «Формируется файл оптимизации». В это время производились действия, необходимые для расчетов, вызывались нужные программы, склеивались, производилась трансляция, определялись фактические параметры процедур. После выполнения этих операций система «спросила» пользователя: «Начать счет?». Ответ пользователя был: «Да».

В процессе расчетов на экран терминала выводились результаты, полученные в процессе выполнения четырех шагов итераций по методу A8. Строка, начинающаяся с K = 0, содержит сведения о начальной точке. Значение функции Била в ней было равно 495.70. За проделанные четыре шага это значение было понижено до 0.70312. Пользователь, естественно, счел полученную точку удачной

и решил расчеты продолжать из нее. Количество обращений и вычислению функции составило всего лишь 30. Вместе с тем, анализ полученных результатов показывает, что на втором, третьем и четвертых шагах метод работал неэффективно, почти не изменяя координат вектора x. Поэтому стало целесообразным перейти к использованию других, более быстро сходящихся алгоритмов, полагая, что найденная точка будет приемлемой для их работы. Пользователь выбрал методификацию метода сопряженных градиентов А31. Для него были заданы следующие параметры:

$$C=0.1, \quad E=.00001, \quad H=.0001, \quad D=5,$$
$$HS=0, \quad HP=1, \quad E1=.1.$$

После формирования файла оптимизации система начала счет. Так как $HS=0$, повторилась в начале начальная точка, найденная из предыдущих расчетов и далее выводились последующие результаты. За проделанные 5 шагов произошло 67 обращений к минимизируемой функции, значение функции уменьшилось до 0.0054794, т. е. полученная точка была, безусловно, лучше исходной. Метод А31 однако, на последних четырех шагах несущественно изменил вектор x. Поэтому пользователь решил попытаться сделать еще несколько шагов этим же методом, увеличив, однако, начальный шаг спуска С. Была введена директива $C:=1$, остальные параметры метода не изменялись. Поэтому было сделано 5 шагов, в результате которых значение функции уменьшилось до величины 0.0000935, количество обращений к вычислению функции было 76. На последних шагах значение минимизируемой функции уменьшалось значительно, поэтому пользователь решил продолжить расчеты, не меняя никаких параметров метода. За следующие 5 шагов значение минимизируемой функции уменьшилось до $0.81249_{10}-6$. На этом пользователь решил прекратить расчеты. Приведем протокол этого расчета, выведенный на алфавитно-цифровое печатающее устройство.

Протокол
-ДИСО
ДИАЛОГОВАЯ СИСТЕМА ОПТИМИЗАЦИИ
ОПРЕДЕЛИТЕ ЗАДАЧУ ('НЛП' ИЛИ 'БМ')
-БМ

```
БЕЗУСЛОВНАЯ МИНИМИЗАЦИЯ
ЗАДАЙТЕ ИМЯ ФУНКЦИИ И ВСПОМОГАТЕЛЬНЫЕ ПРОЦЕДУРЫ
-БИЛ, AGR1, AGS1, ABB1
РАЗМЕРНОСТЬ N-2
ОПРЕДЕЛИТЕ НАЧАЛЬНУЮ ТОЧКУ
  —2.  —2.
НА ВОПРОСЫ ОТВЕЧАЙТЕ 'ДА' ИЛИ 'НЕТ'
ХОТИТЕ ПОСМОТРЕТЬ КАТАЛОГ АЛГОРИТМОВ?
-ДА
A1, A11,—МЕТОДЫ ПОКООРДИНАТНОГО СПУСКА
A22, A23—МЕТОДЫ НАИСКОРЕЙШЕГО СПУСКА
A2, A21—НАИСКОРЕЙШИЙ СПУСК ПОЛЛАКА
A3, A31, A32—МЕТОДЫ СОПРЯЖЕННЫХ ГРАДИЕНТОВ
A4—МЕТОД НЬЮТОНА
A5—МЕТОД ПАУЭЛЛА
A6, A7—МЕТОДЫ СЛУЧАЙНОГО ПОИСКА
A8—МЕТОД ХУКА И ДЖИВСА
A9—СИМПЛЕКС-МЕТОД
ЗАДАЙТЕ МЕТОД БЕЗУСЛОВНОЙ МИНИМИЗАЦИИ
-A8
ОПРЕДЕЛИТЕ ДЛЯ МЕТОДА A8
C—ШАГ, E—ТОЧНОСТЬ, H—ШАГ ГРАД, D—ЧИСЛО ШАГОВ,
HP—ШАГ ПЕЧАТИ, HS—ШАГ НАЧАЛА ПЕЧАТИ И ДОПОЛНИТЕЛЬНЫЙ ПАРАМЕТР E1
-1.  0.00001  0.5  4.  1.  0.  1
1.000000
0.000010
0.500000
4.0
1.0
0.0
1.000000
ФОРМИРУЕТСЯ ФАЙЛ ОПТИМИЗАЦИИ
НАЧАТЬ СЧЕТ?
-ДА
K=0  X=—.20000 10+01 —.20000 10+01  Y=.49570 10+03
K=1  X=—.10000 10+01 —.10000 10+01  Y=.38703 10+02
K=2  X=.20000 10+01   .00000 10+00  Y=.70312 10+00
K=3  X=.20000 10+01   .90949 10—12  Y=.70312 10+00
K=4  X=.20000 10+01   .18190 10—11  Y=.70312 10+00
```

```
 CA=30
ПРОДОЛЖАТЬ ПОИСК ИЗ ПОЛУЧЕННОЙ ТОЧКИ?
-ДА
ПОИСК ПРОДОЛЖАЕТСЯ ИЗ ПОЛУЧЕННОЙ ТОЧКИ
ПРОДОЛЖИТЬ ПОИСК ТЕМ ЖЕ МЕТОДОМ?
-НЕТ
ХОТИТЕ ПОСМОТРЕТЬ КАТАЛОГ АЛГОРИТМОВ?
-НЕТ
ЗАДАЙТЕ МЕТОД БЕЗУСЛОВНОЙ МИНИМИЗАЦИИ
-A31
ОПРЕДЕЛИТЕ ДЛЯ МЕТОДА A31
C—ШАГ, E—ТОЧНОСТЬ, H—ШАГ ГРАД, D—ЧИСЛО
ШАГОВ, HP—ШАГ ПЕЧАТИ, HS—ШАГ НАЧАЛА ПЕЧАТИ
                    И ДОПОЛНИТЕЛЬНЫЙ ПАРАМЕТР E1
-.1  0.00001  0.0001  5.  1.  0.  .1
0.100000
0.000010
0.000100
5.0
1.0
0.0
0.100000
ФОРМИРУЕТСЯ ФАЙЛ ОПТИМИЗАЦИИ
НАЧАТЬ СЧЕТ?
-ДА

 K=0   X=.20000₁₀+01  .18190₁₀-11  Y=.70312₁₀+00
 K=1   X=.20876₁₀+01  .23358₁₀+00  Y=.40445₁₀+00
 K=2   X=.32098₁₀+01  .54948₁₀+00  Y=.57291₁₀-02
 K=3   X=.32091₁₀+01  .54676₁₀+00  Y=.56190₁₀-02
 K=4   X=.32076₁₀+01  .54798₁₀+00  Y=.55458₁₀-02
 K=5   X=.32054₁₀+01  .54543₁₀+00  Y=.54794₁₀-02
CA=67

ПРОДОЛЖИТЬ ПОИСК ИЗ ПОЛУЧЕННОЙ ТОЧКИ?
-ДА
ПОИСК ПРОДОЛЖАЕТСЯ ИЗ ПОЛУЧЕННОЙ ТОЧКИ
ПРОДОЛЖИТЬ ПОИСК ТЕМ ЖЕ МЕТОДОМ?
-ДА
НАЧАТЬ СЧЕТ?
-НЕТ
ХОТИТЕ ПЕРЕОПРЕДЕЛИТЬ ПАРАМЕТРЫ?
-ДА
```

ДАЙТЕ ДИРЕКТИВУ 'ИМЯ ПАРАМЕТРА'=
-С=
-1.
НАЧАТЬ СЧЕТ?
-ДА

К=0	X=.32054$_{10}$+01	.54543$_{10}$+00	Y=.54794$_{10}$−02
К=1	X=.32039$_{10}$+01	.54736$_{10}$+00	Y=.53755$_{10}$−02
К=2	X=.32028$_{10}$+01	.54570$_{10}$+00	Y=.53085$_{10}$−02
К=3	X=.30288$_{10}$+01	.51349$_{10}$+00	Y=.11284$_{10}$−02
К=4	X=.29756$_{10}$+01	.49663$_{10}$+00	Y=.26535$_{10}$−03
К=5	X=.29765$_{10}$+01	.49375$_{10}$+00	Y=.93503$_{10}$−04

СА=76
ПРОДОЛЖИТЬ ПОИСК ИЗ ПОЛУЧЕННОЙ ТОЧКИ?
-ДА
ПОИСК ПРОДОЛЖАЕТСЯ ИЗ ПОЛУЧЕННОЙ ТОЧКИ
ПРОДОЛЖИТЬ ПОИСК ТЕМ ЖЕ МЕТОДОМ?
-ДА
НАЧАТЬ СЧЕТ?
-ДА

К=0	X=.29765$_{10}$+01	.49375$_{10}$+00	Y=.93503$_{10}$−04
К=1	X=.29767$_{10}$+01	49428$_{10}$+00	Y=.89811$_{10}$−04
К=2	X=.29771$_{10}$+01	.49403$_{10}$+00	Y=.87480$_{10}$−04
К=3	X=.29943$_{10}$+01	.49749$_{10}$+00	Y=.32514$_{10}$−04
К=4	X=.30022$_{10}$+01	.50071$_{10}$+00	Y=.13862$_{10}$−05
К=5	X=.30022$_{10}$+01	.50054$_{10}$+00	Y=.81249$_{10}$−06

СА=71
ПРОДОЛЖИТЬ ПОИСК ИЗ ПОЛУЧЕННОЙ ТОЧКИ?
-НЕТ
ОСТАВИТЬ ПРЕДЫДУЩЕЕ ПРИБЛИЖЕНИЕ?
-НЕТ
НАЧАТЬ ПОИСК С НОВОЙ ТОЧКИ?
-НЕТ
ЗАКОНЧИТЬ РЕШЕНИЕ?
-ДА
СИСТЕМА ЗАКОНЧИЛА СЧЕТ

Пример 2. Решалась задача о безусловной минимизации функции Розенброка. Для вычисления производных и вывода информации предполагалось использовать стандартные процедуры. В качестве начальной была взята точка с координатами 3, — 3. Значение минимизируемой функции в ней равно 14404. Начальное приближение,

таким образом, определено весьма грубо, поэтому расчеты были начаты с простейшего варианта покоординатного спуска. Пользователь не стал просматривать каталог методов, так как он ознакомился с ним при решении предыдущего примера. Пользователь определил параметры метода После выполненых пяти шагов значение минимизируемой функции стало равным 0.040624. Последние шаги были малоэффективными, поэтому пользователь решил повторить расчеты этим методом, увеличив шаг С, взяв его равным 2. Однако это не привело к каким-либо существенным результатам. После следующих пяти шагов значение минимизируемой функции составило 0.040128. Поэтому пользователь перешел к методу Ньютона. За десять шагов удалось уменьшить значение минимизируемой функции до $0.30252_{10}-7$. На этом расчеты были закончены.

Протокол

```
-ДИСО
ДИАЛОГОВАЯ СИСТЕМА ОПТИМИЗАЦИИ
ОПРЕДЕЛИТЕ ЗАДАЧУ ('НЛП' ИЛИ 'БМ')
-БМ
БЕЗУСЛОВНАЯ МИНИМИЗАЦИЯ
ЗАДАЙТЕ ИМЯ ФУНКЦИИ И ВСПОМОГАТЕЛЬНЫЕ
                                    ПРОЦЕДУРЫ
-РОЗЕНБРОК
РАЗМЕРНОСТЬ N=2
ОПРЕДЕЛИТЕ НАЧАЛЬНУЮ ТОЧКУ
-3   -3
НА ВОПРОСЫ ОТВЕЧАЙТЕ 'ДА' ИЛИ 'НЕТ'
ХОТИТЕ ПОСМОТРЕТЬ КАТАЛОГ АЛГОРИТМОВ?
-НЕТ
ЗАДАЙТЕ МЕТОД БЕЗУСЛОВНОЙ МИНИМИЗАЦИИ
-А1
ОПРЕДЕЛИТЕ ДЛЯ МЕТОДА А1
С—ШАГ, Е—ТОЧНОСТЬ, Н—ШАГ ГРАД, D—ЧИСЛО
ШАГОВ, НР—ШАГ ПЕЧАТИ, HS—ШАГ НАЧАЛА ПЕЧАТИ
                    И ДОПОЛНИТЕЛЬНЫЙ ПАРАМЕТР Е1
-1   0.00001   0.0001   5.   1.   0.   .1
1.000000
0 000010
0.000100
5.0
```

```
1 0
0 0
0.100000
ФОРМИРУЕТСЯ ФАЙЛ ОПТИМИЗАЦИИ
НАЧАТЬ СЧЕТ?
-ДА
  K=0  X= 30000₁₀+01  —.30000₁₀+01  Y= 14404₁₀+05
  K=1  X= 12416₁₀+01   .14351₁₀+01  Y= 11925₁₀+01
  K=2  X= 12025₁₀+01   .14457₁₀+01  Y= 41010₁₀—01
  K=3  X=.12020₁₀+01   .14450₁₀+01  Y= 40823₁₀—01
  K=4  X= 12018₁₀+01   .14444₁₀+01  Y= 40724₁₀—01
  K=5  X=.12016₁₀+01   .14438₁₀+01  Y= 40624₁₀—01
 CA=143
ПРОДОЛЖИТЬ ПОИСК ИЗ ПОЛУЧЕННОЙ ТОЧКИ?
-ДА
ПОИСК ПРОДОЛЖАЕТСЯ ИЗ ПОЛУЧЕННОЙ ТОЧКИ
ПРОДОЛЖИТЬ ПОИСК ТЕМ ЖЕ МЕТОДОМ?
-ДА
НАЧАТЬ СЧЕТ?
-НЕТ
ХОТИТЕ ПЕРЕОПРЕДЕЛИТЬ ПАРАМЕТРЫ?
-ДА
ДАЙТЕ ДИРЕКТИВУ 'ИМЯ ПАРАМЕТРА'=
-C=
-2.
2 000000
НАЧАТЬ СЧЕТ?
-ДА
  K=0  X= 12016₁₀+01   14438₁₀+01  Y= 40624₁₀—01
  K=1  X= 12013₁₀+01   14432₁₀+01  Y= 40524₁₀—01
  K=2  X=.12011₁₀+01  .14427₁₀+01  Y= 40425₁₀—01
  K=3  X= 12008₁₀+01  .14421₁₀+01  Y= 40326₁₀—01
  K=4  X= 12006₁₀+01  .14415₁₀+01  Y= 40227₁₀—01
  K=5  X= 12003₁₀+01  .14409₁₀+01  Y= 40128₁₀—01
 CA=151
ПРОДОЛЖИТЬ ПОИСК ИЗ ПОЛУЧЕННОЙ ТОЧКИ?
 ДА
ПОИСК ПРОДОЛЖАЕТСЯ ИЗ ПОЛУЧЕННОЙ ТОЧКИ
ПРОДОЛЖИТЬ ПОИСК ТЕМ ЖЕ МЕТОДОМ?
-НЕТ
ХОТИТЕ ПОСМОТРЕТЬ КАТАЛОГ АЛГОРИТМОВ?
-НЕТ
```

```
ЗАДАЙТЕ МЕТОД БЕЗУСЛОВНОЙ МИНИМИЗАЦИИ
-A4
ОПРЕДЕЛИТЕ ДЛЯ МЕТОДА A4
C — ШАГ, E — ТОЧНОСТЬ, H — ШАГ ГРАД, D — ЧИСЛО ШАГОВ,
HP — ШАГ ПЕЧАТИ, HS — ШАГ НАЧАЛА ПЕЧАТИ И ДОПОЛ-
                                НИТЕЛЬНЫЙ ПАРАМЕТР E1
—0.1 0.00001 0.001 10. 1. 0. 0.1
0.100000
0.000010
0.001000
10.0
1.0
0.0
0.100000
ФОРМИРУЕТСЯ ФАЙЛ ОПТИМИЗАЦИИ
НАЧАТЬ СЧЕТ?
-ДА
  K=0   X=.12003₁₀+01  .14409₁₀+01  Y=.40128₁₀—01
  K=1   X=.11219₁₀+01  .12527₁₀+01  Y=.18555₁₀—01
  K=2   X=.10301₁₀+01  .10567₁₀+01  Y=.28365₁₀—02
  K=3   X=.10102₁₀+01  .10211₁₀+01  Y=.14433₁₀—03
  K=4   X=.10011₁₀+01  .10022₁₀+01  Y=.16507₁₀—05
  K=5   X=.99985₁₀+00  .99971₁₀+00  Y=.21604₁₀—07
  K=6   X=.99985₁₀+00  .99970₁₀+00  Y=.23688₁₀—07
  K=7   X=.99984₁₀+00  .99968₁₀+00  Y=.25594₁₀—07
  K=8   X=.99984₁₀+00  .99967₁₀+00  Y=.27320₁₀—07
  K=9   X=.99983₁₀+00  .99966₁₀+00  Y=.28870₁₀—07
  K=10  X=.99983₁₀+00  .99965₁₀+00  Y=.30252₁₀—07
 CA=222
ПРОДОЛЖИТЬ ПОИСК ИЗ ПОЛУЧЕННОЙ ТОЧКИ?
-НЕТ
ОСТАВИТЬ ПРЕДЫДУЩЕЕ ПРИБЛИЖЕНИЕ?
-НЕТ
НАЧАТЬ ПОИСК С НОВОЙ ТОЧКИ?
-НЕТ
ОКОНЧИТЬ РЕШЕНИЕ?
-ДА
СИСТЕМА ЗАКОНЧИЛА СЧЕТ
```

Пример 3. Решалась задача нелинейного программирования (4.1). В качестве начальной была взята точка с координатами —2, —2, —2. Для счета был выбран

релаксационный метод С61, определены его параметры:

$$C=.5,\quad E=.00001,\quad H=.0001,\quad D=5,$$
$$HS=0,\quad HP=1.$$

В отличие от режима расчета задач БМ, при решении задач НЛП пользователь «ведет» диалог. Обладая некоторым запасом команд, он сам вводит с терминала указания о дальнейшем процессе расчетов. Единственной командой, которая поступает от системы, является команда определить параметры методов. Она следует после того как пользователь указывает новый метод.

Пользователь дал команду «Счет». На экране был выведен вектор v. Пользователь не давал команд об изменении этого вектора, поэтому все его координаты равны единице. Далее был напечатан текст: «Точка X не является внутренней». Напомним, что для работы метода С61 обязательно надо задавать в качестве начальной точки внутреннюю. Одновременно было выведено содержимое векторов Y, X, P. Информация о недопустимости точки X была верной. Действительно, значение функции, определяющей ограничения типа равенства, оказалось равным $Y[1]=-7$ вместо нуля. Значения $Y[2]=Y[3]=Y[4]=2>0$, $Y[5]=15>0$, в то время как все эти величины должны быть отрицательными. Вектор двойственных переменных не существен для этого метода, он не задавался пользователем. Поэтому в системе всем координатам P были приписаны единицы. Пользователь обратился к вспомогательной процедуре ВНУТР, предназначенной для отыскания внутренней точки.

В теле этой процедуры формируется функция штрафа

$$S(x)=\sum_{i=1}^{l}[f^i(x)]^2+\sum_{i=l+1}^{m}[f_+^i(x)]^2;$$

здесь $f_+^i=\max[0, f^i]$. Далее производится безусловная минимизация этой функции. Система запросила пользователя, какой метод безусловной оптимизации он предлагает использовать для отыскания внутренней точки. Пользователь указал программу А31, определил необходимые параметры. После пяти шагов была найдена новая точка, в которой выполнялись все ограничения типа неравенств. Функция, определяющая ограничения типа равенства, при-

нимала значение 0.00039. Пользователь счел это значение приемлемым, дал команду ОК, означающую, что найденная точка X будет дальше использоваться в качестве начальной и вновь обратился к методу С61. Параметры метода были определены раньше, поэтому сразу давалась директива «Счет». После пяти шагов была получена новая точка, в которой ограничения типа неравенства были удовлетворены, ограничение типа равенства выполнялось примерно с той же точностью, что и в начале расчетов по методу С61. Значение минимизируемой функции при этом уменьшилось с 1.5904 до 0.0074687. Пользователь дал команду ОК и перешел к методу Ньютона. Пользователь решил переопределить только два параметра, положив D = 3, C = 1.

Метод Ньютона «сработал» исключительно эффективно. Уже на первом шаге ошибка в выполнении ограничений типа равенства составила $Y[1] = -.114_{10} - 9$, после дальнейших двух шагов было получено $Y[1] = -.909_{10} - 12$. В точке решения величина Y [5] должна быть равна нулю. Вместо этого получено: $Y[5] = .456_{10} - 10$. Значение минимизируемой функции точно не известно. Согласно приведенным результатам $Y[0] = -.998_{10} - 4$. Расчеты были на этом закончены.

```
Протокол
-ДИСО
ДИАЛОГОВАЯ СИСТЕМА ОПТИМИЗАЦИИ
ОПРЕДЕЛИТЕ ЗАДАЧУ ('НЛП' ИЛИ 'БМ')
-НЛП
НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ
ЗАДАЙТЕ ИМЯ ФУНКЦИИ И ВСПОМОГАТЕЛЬНЫЕ
                                ПРОЦЕДУРЫ
— Ф1, CGR1, CGS1, CBB1
N = 3 M = 5 L = 1
ХОТИТЕ ПОСМОТРЕТЬ КАТАЛОГ АЛГОРИТМОВ?
-ДА
С2, С3, С4, С41 — МОДИФИКАЦИИ МЕТОДА ШТРАФОВ
С5 — МЕТОД НАГРУЖЕННОГО ФУНКЦИОНАЛА
С6, С61, С62 — РЕЛАКСАЦИОННЫЕ МЕТОДЫ
С7, С9, С12 — ДВОЙСТВЕННЫЕ МЕТОДЫ
С8 — МЕТОД НЬЮТОНА
С10 — МЕТОД ЛИНЕАРИЗАЦИИ
```

```
С13 — МЕТОД ВОЗМОЖНЫХ НАПРАВЛЕНИЙ
ОПРЕДЕЛИТЕ НАЧАЛЬНУЮ ТОЧКУ
— —2  —2. —2.
ЗАДАЙТЕ ИМЯ МЕТОДА
— С61
ФОРМИРУЕТСЯ ФАЙЛ ОПТИМИЗАЦИИ
ОПРЕДЕЛИТЕ С — ШАГ, Е — ТОЧНОСТЬ, Н — ШАГ ГРАД,
D — ЧИСЛО ШАГОВ, HP — ШАГ ПЕЧ, HS — ШАГ НАЧАЛА ПЕЧ.
— 0.5 0.00001 0.0001 5. 1. 0.
0 500000
0.000010
0 000100
5.0
1.0
0 0
ДАЛЬШЕ РАБОТАЙТЕ САМОСТОЯТЕЛЬНО
-СЧЕТ
V=+1.0000  +1.0000  +1.0000  +1.0000  +1.0000  +1.0000
ЧКА X НЕ ЯВЛЯЕТСЯ ВНУТРЕННЕЙ
=1  Y=.98169₁₀+02  —.70000₁₀+01  .20000₁₀+01
                   .20000₁₀+01  .20000₁₀+01  .15000₁₀+01
=—.20000₁₀+01  —.20000₁₀+01  —.20000₁₀+01  P=.10000₁₀+01
          .10000₁₀+01  .10000₁₀+01  .10000₁₀+01  .10000₁₀+01
CA=6
-ВНУТР
ОПРЕДЕЛЕНИЕ ДОПУСТИМОЙ ТОЧКИ
ЗАДАЙТЕ МЕТОД БЕЗУСЛОВНОЙ МИНИМИЗАЦИИ
-А31
ОПРЕДЕЛИТЕ ДЛЯ МЕТОДА А31
С — ШАГ, Е — ТОЧНОСТЬ, Н — ШАГ ГРАД, D — ЧИСЛО ШАГОВ,
HP — ШАГ ПЕЧАТИ, HS — ШАГ НАЧАЛА ПЕЧАТИ И ДОПОЛ-
                                НИТЕЛЬНЫЙ ПАРАМЕТР Е1
— 0 1 0 000001 0.0001 5. 1. 0. 0.1
0 100000
0.000001
0.000100
5.0
1.0
0.0
0.100000
ФОРМИРУЕТСЯ ФАЙЛ ОПТИМИЗАЦИИ
-СЧЕТ
```

```
V=+1.0000  +1.0000  +1.0000  +1.0000  +1.0000  +1.0000
Y=+.9817₁₀+02  -.7000₁₀+01  +.2000₁₀+01
                +.2000₁₀+01  +.2000₁₀+01  +.1500₁₀+02
X=-.2000₁₀+01  -.2000₁₀+01  -.2000₁₀+01
K=0  X=-.20000₁₀+01  -.20000₁₀+01  -.20000₁₀+01
     Y=.28600₁₀+03
K=1  X=-.26961₁₀+01  -.15970₁₀+01  -.17191₁₀+01
     Y=.61945₁₀+02
K=2  X=-.58145₁₀+00  .96362₁₀+00  .78898₁₀+00
     Y=.36730₁₀+00
K=3  X=.34044₁₀+00  .47696₁₀+00  .30387₁₀+00
     Y=.14708₁₀-01
K=4  X=.32902₁₀+00  .41814₁₀+00  .24510₁₀+00
     Y=.59943₁₀-04
K=5  X=.33173₁₀+00  .42085₁₀+00  .24781₁₀+00
     Y=.14986₁₀-06
     Y=.1590₁₀+01  .3871₁₀-03  -.3317₁₀+00
               -.4208₁₀+00  -.2478₁₀+00  -.4798₁₀+00
CA=852
ЧТО ДЕЛАТЬ ДАЛЬШЕ?
-M C61
ФОРМИРУЕТСЯ ФАЙЛ ОПТИМИЗАЦИИ
-OK
СЧЕТ
V=+1.0000  +1.0000  +1.0000  +1.0000  +1.0000  +1.0000
K=0  Y=.15904₁₀+01  .39000₁₀-03  -.33173₁₀+00
               -.42085₁₀+00  -.24781₁₀+00  -.47983₁₀+00
     X=.33173₁₀+00  .42085₁₀+00.  24781₁₀+00
     P=.10000₁₀+01  .10000₁₀+01.  10000₁₀+01
                                  .10000₁₀+01  .10000₁₀+01
K=1  Y=.37660₁₀+00  .38998₁₀-03  -.23190₁₀+00
               -.10326₁₀+00  -.66522₁₀+00  -.26799₁₀+00
     X=.23190₁₀+00  .10326₁₀+00  .66522₁₀+00
     P=-.38333₁₀+01  .60185₁₀+00  .15093₁₀+01
                                  -.33688₁₀+01  .84580₁₀+00
K=2  Y=.10915₁₀+00  .39222₁₀-03  -.29580₁₀+00
               -.14395₁₀+00  -.56064₁₀+00  -.80375₁₀-01
     X=.29580₁₀+00  .14395₁₀+00  .56064₁₀+00
     P=.15168₁₀+00  -.55108₁₀+00  -.78812₁₀+00
                                  .31445₁₀+00  .13771₁₀+01
K=3  Y=.43845₁₀-01  .39225₁₀-03  -.28751₁₀+00
               -.10472₁₀+00  -.60816₁₀+00  -.37185₁₀-01
```

```
          X=.28751₁₀+00  .10472₁₀+00  .60816₁₀+00
          P=—.66702₁₀+00  .56053₁₀—01  .54509₁₀+00
                                       —.16953₁₀+00  .10732₁₀+01
   K=4    Y=.17837₁₀—01  .39226₁₀—03  —.29377₁₀+00
                         —.10701₁₀+00  —.59961₁₀+00  —.15140₁₀—01
          X=.29377₁₀+00  .10701₁₀+00  .59961₁₀+00
          P=—.20415₁₀+00  —.43547₁₀—01  —.43757₁₀—01
                                       .28122₁₀—01  .11839₁₀+01
   K=5    Y=.74687₁₀—02  .39226₁₀—03  —.29393₁₀+00
                         —.10298₁₀+00  —.60348₁₀+00  —.64081₁₀—02
          X=.29393₁₀+00  .10298₁₀+00  .60348₁₀+00
          P=—.28220₁₀+00  —.10780₁₀—02  .75297₁₀—01
                                       —.12910₁₀—01  .11535₁₀+01
   CA=216
   -ОК
   -М  С8
   ФОРМИРУЕТСЯ ФАЙЛ ОПТИМИЗАЦИИ
   -Д=
   ПРИЕМ ЧИСЛА ШАГОВ МЕТОДА
   -3.
   -С=
   ПРИЕМ ШАГА МЕТОДА
   -1.
   -СЧЕТ
V=+1.0000  +1.0000  +1.0000  +1.0000  +1.0000  +1.0000
K=0  Y=.74656₁₀—02  .39000₁₀—03  —.29393₁₀+00  —.10298₁₀+00
                                  —.60348₁₀+00  —.64060₁₀—02
X=.29393₁₀+00  .10298₁₀+00  .60348₁₀+00
P=—.28220₁₀+00  .32833₁₀—01  27440₁₀+00  .35930₁₀+00  .10740₁₀+01
K=1  Y=—.98171₁₀—05  —.11369₁₀—09  —.29563₁₀+00  —.10415₁₀+00
                                  —.60022₁₀+00  .56838₁₀—04
X=.29563₁₀+00  .10415₁₀+00  .60022₁₀+00
P=—.28910₁₀+00  —.18993₁₀—03  .31201₁₀—02  .19417₁₀+02
                                                .10831₁₀+01
K=2  Y=—.10206₁₀—03  .18190₁₀—11  —.29428₁₀+00  —.10130₁₀+00
                                  —.60442₁₀+00  .19347₁₀—05
X=.29428₁₀+00  .10130₁₀+00  .60442₁₀+00
P=—.22733₁₀+00  —.86725₁₀—06  —.85332₁₀—04
                                  —.13582₁₀—04  .10793₁₀+01
K=3  Y=—.99802₁₀—04  —.90949₁₀—12  —.29428₁₀+00  —.10130₁₀+00
                                  —.60442₁₀+00  .28137₁₀—11
X=.29428₁₀+00  .10130₁₀+10  .60442₁₀+00
```

P = —.22735$_{10}$+00 .00000$_{10}$+00 .00000$_{10}$+00 .00000$_{10}$+00
.11643$_{10}$+01

ЧТО ДЕЛАТЬ ДАЛЬШЕ?
-КОНЕЦ

Пример 4. Решалась задача (4.1). Начальная точка бралась следующая (2, 2, 2). Расчеты начинались с использования метода внешних штрафов С41. Вспомогательная задача минимизации решалась методом сопряженных градиентов АЗ. Основные параметры метода были

$$C = .02, \quad E = .00001, \quad H = .0001,$$
$$D = 2, \quad HS = 0, \quad HP = 1.$$

Дополнительные параметры

$$A = 5, \quad Z = 10000, \quad B = 1$$

обозначают, что на первом шаге коэффициент штрафа равен 5, на втором шаге он увеличивается в 10000 раз. Точность решения задачи безусловной минимизации не изменяется. Благодаря резкому увеличению штрафа, удалось с высокой степенью точности удовлетворить ограничения типа равенства. В начальной точке Y[1] = 5, после первой итерации Y[1] = —.0231, после второй итерации Y[1] = —.8463$_{10}$ — 5. Расчеты по методу штрафов позволили с хорошей точностью определить двойственные переменные, подготовив, таким образом, начальные условия для применения метода Ньютона. Обратившись к методу Ньютона, пользователь получил то же самое решение, что и в предыдущем примере, и на этом закончил расчеты.

Протокол
-ДИСО
ДИАЛОГОВАЯ СИСТЕМА ОПТИМИЗАЦИИ
ОПРЕДЕЛИТЕ ЗАДАЧУ («НЛП» ИЛИ «БМ»)
-НЛП
НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ
ЗАДАЙТЕ ИМЯ ФУНКЦИИ И ВСПОМОГАТЕЛЬНЫЕ ПРОЦЕДУРЫ
—Ф1, CGR1, CGS1, CBB1
N=3 M=5 L=1
ХОТИТЕ ПОСМОТРЕТЬ КАТАЛОГ АЛГОРИТМОВ?
-НЕТ
ОПРЕДЕЛИТЕ НАЧАЛЬНУЮ ТОЧКУ

—2. 2. 2.
ЗАДАЙТЕ ИМЯ МЕТОДА
-C41
ФОРМИРУЕТСЯ ФАЙЛ ОПТИМИЗАЦИИ
ОПРЕДЕЛИТЕ С — ШАГ, Е — ТОЧНОСТЬ, Н — ШАГ ГРАД, D — ЧИСЛО ШАГОВ, НР — ЧИСЛО ПЕЧ, HS — ШАГ НАЧАЛА ПЕЧ.
-0 02 0.00001 0.0001 2. 1. 0.
0.020000
0 000010
0 000100
2.0
1.0
0 0
ОПРЕДЕЛИТЕ ДОПОЛНИТЕЛЬНЫЕ ПАРАМЕТРЫ МЕТОДА
A, Z, B
-5. 10000. 1.
ДАЛЬШЕ РАБОТАЙТЕ САМОСТОЯТЕЛЬНО
СЧЕТ
V = +1 0000 +1.0000 +1.0000 +1.0000 +1.0000 +1.0000
K = 0 Y = $.98169_{10}+02$ $50000_{10}+01$ $-20000_{10}+01$
$-20000_{10}+01$ $-20000_{10}+01$ $-.90000_{10}+01$
X = $.20000_{10}+01$ $.20000_{10}+01$ $20000_{10}+01$
K = 1 Y = $-.13415_{10}+00$ $-23134_{10}-01$ $-.29134_{10}+00$
$-.84664_{10}-01$ $-.60086_{10}+00$ $.11332_{10}+00$
X = $.29134_{10}+00$ $.84664_{10}-01$ $60086_{10}+00$
K = 2 Y = $-.55469_{10}-05$ $-84629_{10}-05$ $-.29263_{10}+00$
$-.97749_{10}-01$ $-.60961_{10}+00$ $.12535_{10}-03$
X = $.29263_{10}+00$ $.97749_{10}-01$ $60961_{10}+00$
P = $-33860_{10}+00$ $.00000_{10}+00$ $.00000_{10}+00$
$.00000_{10}+00$ $.23477_{10}+01$
CA = 3324
-OK
-M C8
ФОРМИРУЕТСЯ ФАЙЛ ОПТИМИЗАЦИИ
-C=
ПРИЕМ ШАГА МЕТОДА
-1.
-D=
ПРИЕМ ЧИСЛА ШАГОВ МЕТОДА
- 5. 1.
-СЧЕТ

```
V=+1.0000  +1.0000  +1.0000  +1.0000  +1.0000  +1.0000
K=0  Y=-.55469(10)-05  -.84629(10)-05  -.29263(10)+00
                -.97749(10)-01  -.60961(10)+00  .12535(10)-03
X=   X=.29263(10)+00  .97749(10)-01  .60961(10)+00
     P=-.33860(10)+00  .10000(10)-02  .10000(10)-02
                                      .10000(10)-02  .15322(10)+01
K=1  Y=-.13911(10)-03  -.18645(10)-09  -.29427(10)+00
                -.10125(10)+00  -.60448(10)+00  .33793(10)-04
     X=.29427(10)+00  .10125(10)+00  .60448(10)+00
     P=-.22978(10)+00  -.55807(10)-05  -.35867(10)-04
                                      .84162(10)-05  .11461(10)+01
K=2  Y=-.10203(10)-03  -.90949(10)-12  -.29428(10)+00
                -.10130(10)+00  -.60442(10)+00  .19107(10)-05
     X=.29428(10)+00  .10130(10)+00  .60442(10)+00
     P=-.22736(10)+00  .27981(10)-09  .16778(10)-07
                                      .86490(10)-09  .10812(10)+01
K=3  Y=-.99806(10)-04  -.90949(10)-12  -.29428(10)+00
                -.10130(10)+00  -.60442(10)+00  .34580(10)-08
     X=.29428(10)+00  .10130(10)+00  .60442(10)+00
     P=-.22735(10)+00  .23855(10)-15  .69443(10)-13
                                      .24097(10)-15  .10793(10)+01
K=4  Y=-.99802(10)-04  -.90949(10)-12  -.29428(10)+00
                -.10130(10)+00  -.60442(10)+00  .37232(10)-11
     X=.29428(10)+00  .10130(10)+00  .60442(10)+00
     P=-.22735(10)+00  .00000(10)+00  .00000(10)+00
                                      .00000(10)+00  .10793(10)+01
K=5  Y=-.99802(10)-04  -.90949(10)-12  -.29428(10)+00
                -.10130(10)+00  -.60442(10)+00  .28137(10)-11
     X=.29428(10)+00  .10130(10)+00  .60442(10)+00
     P=-.22735(10)+00  .00000(10)+00  .00000(10)+00
                                      .00000(10)+00  .11649(10)+01
CA=411
ЧТО ДЕЛАТЬ ДАЛЬШЕ?
-КОНЕЦ
```

§ 5. Некоторые подходы к проблеме создания управляющих программ

Опыт конструирования и эксплуатации ДИСО показывает, что целесообразно на базе одной и той же библиотеки алгоритмов разработать несколько вариантов программного обеспечения, создавая системы для опытного, начи-

нающего пользователя и пользователя, который либо вовсе не знаком с методами оптимизации, либо в силу ряда причин не может заниматься диалогом (например, он может вести диалог на более высоком уровне и физически не в состоянии одчовременно управлять блоком оптимизации). В последнем случае мы вынуждены создавать пакеты программ, т. е. единый комплекс из библиотеки алгоритмов и управляющей программы, который обеспечивает автоматизированный выбор наиболее подходящей последовательности используемых методов, их параметров, в зависимости от конкретной решаемой задачи. Несложно составить удовлетворительно работающие управляющие программы для решения однотипных задач. Вопрос о создании управляющей программы в общем случае является чрезвычайно сложным по следующим причинам: слишком разнообразными могут быть решаемые задачи и слишком слабо изучены методы их решения.

Первые управляющие программы создаются на основе эвристических рассуждений. Первым шагом для практического создания управляющей программы является разработка специальной информационно-логической системы, которая, используя разнообразную информацию о конкретной задаче, результаты расчетов, отбраковывает наименее подходящие методы и сужает, таким образом, число конкурентоспособных методов. Этот путь использован в [1] *), где авторы строят пакеты программ для решения систем линейных алгебраических уравнений.

Информационно-логическая система должна самостоятельно производить анализ ситуаций, сложившихся в процессе вычисления, используя теоремы из своей памяти, оперируя служебными арифметическими программами. Система будет либо принимать самостоятельные решения, либо давать рекомендации вычислителю. В перспективе желательно создание системы, которая самостоятельно манипулировала бы алгоритмами оптимизации, оценивала ситуации, принимала решения, изменяя параметры, меняя метод и определяя момент окончания расчетов. Сейчас ведется работа по созданию подобных систем.

---

*) [1] Молчанов И. Н., Николенко Л. Д., Кириченко М. П. Об одном пакете программ для решения систем линейных алгебраических уравнений. Кибернетика, 1972, № 1, 127—133.

Рассмотрим в упрощенной постановке задачу о построении оптимальной управляющей программы для решения простейшей задачи — минимизации функции многих переменных $F(x)$. Некоторые подходы к решению этой задачи, по-видимому, впервые обсуждались в [2] *).

Предположим, что существует n методов A1, A2, ..., An безусловной минимизации, причем в каждом методе все вспомогательные параметры вырабатываются автоматически, в зависимости от начальной точки x, с которой начинается счет по выбранному методу. Пользователь задает последовательность работы алгоритмов, количество шагов, сделанных по каждому из них, стремясь за фиксированное время расчетов T получить точку x, в которой минимизируемая функция $F(x)$ принимает возможно меньшее значение. Целочисленная переменная $s=0, 1, 2, \ldots$, обозначает номер шага итеративного процесса, построенного с помощью различных алгоритмов, например, последовательность A1, A1, A1, A8, A8, A31, A9, .. означает, что вначале три шага ($s=1, 2, 3$) было сделано по методу A1, далее — два шага ($s=4, 5$) — по методу A8 и т. д. Последовательности $\{s\}$ соответствует последовательность точек $\{x_s\}$. Пусть на $s-1$ шаге была получена точка $x=x_{s-1}$, сделан s-й шаг по i-му методу и найдена точка $x=x_s$. Определим функции $R_i(x_{s-1})=F(x_{s-1})-F(x_s)$, через $T(x_{s-1})$ будем обозначать время, затраченное ЭВМ на выполнение одного шага i-м методом начиная из точки x_{s-1}. Отношение $w_i(s)=\frac{F(x_{s-1})-F(x_s)}{T(x_{s-1})}$ будем называть эффективностью i-го метода в точке x_{s-1}.

Для всевозможных i определим управляющие функции $\mu_i(k)$, принимающие только два значения: нуль и единицу. Если на k-м шаге используется метод Ai, то положим $\mu_i(k)=1$; если этот метод не используется на k-м шаге, то положим $\mu_i(k)=0$. Тогда задачу об оптимальном управлении процессом минимизации можно сформулировать как задачу о нахождении

$$\max_{\mu_i} \sum_{k=1}^{\infty} \sum_{i=1}^{n} R_i(x_{k-1})\,\mu_i(k)$$

*) [2] Моисеев Н. Н. Методы оптимизации. Гл. I, Изд-во ВЦ АН СССР, 1969.

при условии

$$\sum_{k=1}^{\infty}\sum_{i=1}^{n} T_i(x_{k-1})\,\mu_i(k) \leqslant T, \quad \mu_i(k) = 0,\ 1.$$

Приведенная схема не учитывает адаптирующихся свойств алгоритмов. Поэтому при рассмотрении таких методов (например, метод сопряженных градиентов) следует объединять несколько шагов, рассматривая их как один шаг в последовательности $\{s\}$. Так же можно поступать и в случае нерелаксационных методов. Решение поставленной задачи может оказаться полезной при разработке управляющих программ. Недостаток такого подхода прежде всего в том, что функции $R_i(x)$, $T_i(x)$ заранее не известны. Можно делать адаптирующиеся схемы, используя пробные шаги для выяснения значений функции. Другой подход — изучение этих функций, введение различных аппроксимаций.

Упростим задачу: будем считать, что $T_i(x) \equiv T_i$ и $R_i(x) \equiv R_i$. В этом случае несущественно, в какой последовательности используются алгоритмы, важно лишь, сколько шагов сделал каждый из них. Обозначим $\sum_{k=1}^{\infty} \mu_i(k) = M_i$; тогда задача (4.1) сводится к задаче о рюкзаке:

максимизировать $\sum_{i=1}^{n} R_i M_i$ при ограничениях $\sum_{i=1}^{n} T_i M_i \leqslant$ $\leqslant T$, M_i — целые, $l = \max_i [T/T_i]$.

Методы решения таких задач известны [1]*); заметим, что при больших значениях l в основном будут использоваться алгоритмы с максимальным отношением R_i/T_i, т. е. наиболее эффективные алгоритмы. Описанные упрощенные схемы не учитывают еще одного немаловажного фактора: в процессе расчетов приходится определять не только последовательность методов, но и задавать их параметры. Один выход из этого положения — организовать некоторую сетку на значениях этих параметров, после чего рассматривать метод с каждым конкретным набором параметров

*) [1] Х у Т Целочисл нное программирование и потоки в сетях — М.: Мир, 1974.

(из узлов сетки) как отдельный метод. Задача, таким образом, сводится к рассмотренной выше. Другой вариант — выполнять градиентный спуск по этим параметрам, стремясь к максимизации эффективности метода. Пусть, например, на k-м шаге используется i-й метод. Из одной и той же точки сделаем один шаг с разными значениями $c=c_1$ и $c=c_2$; в результате получим точки x_1, x_2; значениями эффективности будут w_1 и w_2. Пусть $F(x_1)<F(x_2)$; в качестве новой точки x возьмем x_1, в качестве нового значения c примем

$$c=c_1+\alpha\frac{F(x_1)-F(x_2)}{c_1-c_2},$$

где α — некоторый коэффициент. Такая «настройка» требует двойного просчета и целесообразно использовать ее только в тех точках и для тех методов, у которых зависимость $w_i(x)$ от c достаточно резко выражена.

Главный недостаток изложенного приема — отсутствие априорной информации о функциях $R_i(x)$, $T_i(x)$. В каждой фиксированной точке значения этих функций легко определяются с помощью обращения к i-му методу. Однако для этого требуется затратить время. Поэтому если функции $R_i(x)$, $T_i(x)$ заранее не известны, то управление будет носить двойственный (дуальный) характер: для успешного управления системой следует тратить ресурс (время расчетов ЭВМ) не только на решение задачи, но и на изучение системы и тем самым получать возможность улучшать управление.

Возможны другие разнообразные подходы, базирующиеся на различных идеях. Методы статистического последовательного анализа, ситуационного управления, теории адаптивных систем, работы по искусственному интеллекту и многие другие исследования могут быть привлечены для решения поставленной задачи. Важно, что какие бы подходы ни использовались — всегда будет простой критерий оценки из эффективности — сравнительный анализ просчетов по управляющим программам, построенным на их базе.

ЛИТЕРАТУРА

1. Аоки М. Введение в методы оптимизации. — М.: Наука, 1977.
2. Васильев Ф. П. Лекции по методам решения экстремальных задач. — М.: Изд-во МГУ, 1974.
3. Воробьев Н. Н. Числа Фибоначчи. — М.: Наука, 1969.
4. Гасс С. Линейное программирование (методы и приложения). — М.: Физматгиз, 1961.
5. Гирсанов И. В. Лекции по математической теории экстремальных задач. — М.: Изд-во МГУ, 1970.
6. Карманов В. Г. Математическое программирование. — М.: Наука, 1975.
7. Моисеев Н. Н. Элементы теории оптимальных систем. — М.: Наука, 1975.
8. Полак Э. Численные методы оптимизации. — М.. Мир, 1974.
9. Пшеничный Б. Н., Данилин Ю. М. Численные методы в экстремальных задачах. — М.: Наука, 1975.
10. Фиакко А., Мак-Кормик Г. Нелинейное программирование. Методы последовательной безусловной минимизации. — М.: Мир, 1972.
11. Химмельблау Д. Прикладное нелинейное программирование. — М.: Мир, 1975.
12. Численные методы условной оптимизации/ Ред. Ф. Гилл, У. Мюррей. — М.: Мир, 1977.
13. Юдин Д. Б., Гольштейн Е. Г. Линейное программирование. — М.: Наука, 1969.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Никита Николаевич Моисеев
Юрий Павлович Иванилов
Елена Михайловна Столярова

МЕТОДЫ ОПТИМИЗАЦИИ

М 1978 г , 352 стр с илл

Редактор *В Ю Лебедев*

Техн редактор *И Ш Аксельрод*

Корректор *З В Автонеева*

ИБ № 11388

Сдано в набор 09 06 78 Подписано к печати 03 11 78
Т 20132 Бумага 84×108¹/₃₂ Тип № 1 Литературная гарнитура Высокая печать Условн печ л 18,48 Уч изд л 17,49 Тираж 19 000 экз Заказ 1145. Цена книги 85 коп

Издательство «Наука»
Главная редакция
физико математической литературы
117071, Москва, В 71 Ленинский проспект 15

Ордена Октябрьской Революции, ордена Трудового Красного Знамени Ленинградское производственно техническое объединение «Печатный Двор» имени А М. Горького Союзполиграфпрома при Государственном комитете Совета Министров СССР по делам издательств полиграфии и книжной торговли 197136, Ленинград, П 136, Гатчинская ул , 26.

Отпечатано во 2 ой тип изд-ва «Наука»
Москва, Г-99, Шубинский пер , 10