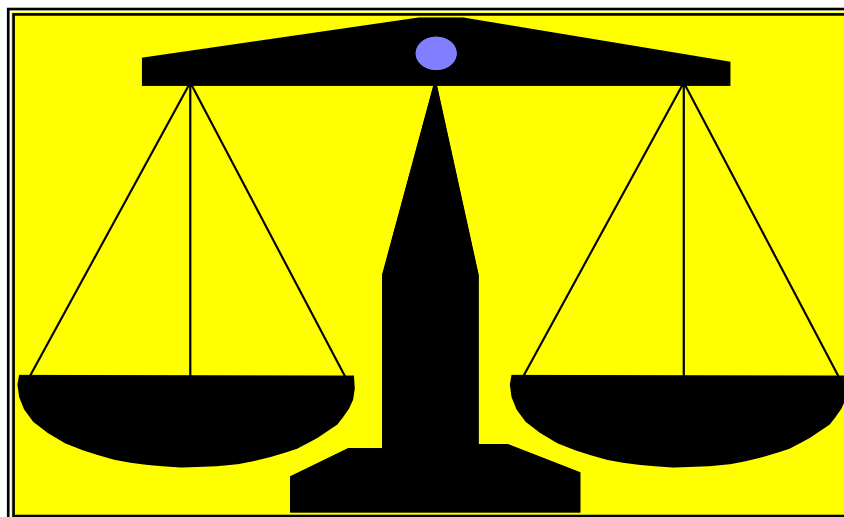


А. Л. Гольдштейн

**ТЕОРИЯ ПРИНЯТИЯ РЕШЕНИЙ.
ЗАДАЧИ И МЕТОДЫ
ИССЛЕДОВАНИЯ ОПЕРАЦИЙ
И
ПРИНЯТИЯ РЕШЕНИЙ**

Учебное пособие



ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ.....	7
Глава 1. ХАРАКТЕРИСТИКА ИССЛЕДОВАНИЯ ОПЕРАЦИЙ	8
1.1. Основные понятия и особенности исследования операций	8
1.2. Этапы операционного проекта	10
1.3. Критерий оптимальности	12
1.4. Виды математических моделей ИСО.....	14
1.5. Классы операционных задач	20
1.6. Резюме	33
Глава 2. ПОСТАНОВКА ЗАДАЧИ ОПТИМИЗАЦИИ	34
Глава 3. МЕТОДЫ КЛАССИЧЕСКОГО АНАЛИЗА	37
3.1. Основные положения теории экстремумов	37
3.2. Метод неопределенных множителей Лагранжа.....	45
3.3. Задания для самостоятельной работы.....	51
Глава 4. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ.....	56
4.1. Постановка задачи	56
4.2. Примеры задач линейного программирования.....	56
4.2.1. Задача составления рациона или как экономно питаться.....	57
4.2.2. Задача оптимального планирования	58
4.2.3. Сбалансированная транспортная задача.....	58
4.2.4. Общая распределительная задача	60
4.2.5. Задача планирования работы оборудования	62
4.2.6. Игра двух лиц с нулевой суммой как задача линейного программирования	63
4.2.7. Резюме.....	64
4.3. Формы записи задач линейного программирования и способы приведения к ним	65
4.3.1. Каноническая форма задач ЛП.....	65
4.3.2. Стандартная форма задачи ЛП	67
4.4. Упрощение модели	68
4.5. Основные понятия ЛП. Свойства задач ЛП	69
4.6. Геометрия задач ЛП.....	71
4.7. Выделение вершин допустимого множества	75
4.8. Методы решения задач ЛП	77
4.9. Симплекс-метод	78
4.9.1. Характеристика метода	78
4.9.2. Переход от одного базисного решения к другому	78
4.9.3. Признак оптимальности. Основные этапы симплекс-метода	81
4.9.4. Построение начального базисного решения	83
4.9.5. Связь между параметрами последовательных итераций.....	87

4.9.6. Алгоритм симплекс-метода	89
4.9.7. Примеры.....	93
4.9.8. Учет двусторонних ограничений	97
4.10. Модифицированный алгоритм	98
4.11. Двойственность задач ЛП	99
4.11.1. Запись двойственной задачи в симметричном случае	100
4.11.2. Интерпретация двойственной задачи	101
4.11.3. Запись двойственной задачи в общем случае	102
4.11.4. Теоремы двойственности	104
4.11.5. Двойственный симплекс-метод	110
4.12. Параметрический анализ.....	112
4.12.1. Параметрирование вектора ограничений	113
4.12.2. Параметрирование коэффициентов линейной формы	117
4.13. Задания для самостоятельной работы.....	120
Глава 5. ТРАНСПОРТНЫЕ ЗАДАЧИ.....	122
5.1. Основные модели транспортных задач	122
5.1.1. Простейшая транспортная задача (Т-задача)	122
5.1.2. Транспортная задача с ограниченными пропускными способностями (T_d - задача)	125
5.1.3. Задачи с неоднородным грузом.....	126
5.1.4. Многоиндексные задачи.....	126
5.1.5. Транспортные задачи по критерию времени	127
5.2. Метод потенциалов.....	128
5.2.1. Построение начального плана перевозок	128
5.2.2. Переход от одного плана перевозок к другому	131
5.2.3. Признак оптимальности	132
5.2.4. Алгоритм метода потенциалов	135
5.2.5. Двойственная пара транспортных задач.....	138
5.3. Приведение открытой транспортной задачи к закрытой.....	139
5.4. Метод потенциалов для T_d -задачи	142
5.5. Решение задачи по критерию времени	146
5.6. Транспортные задачи в сетевой постановке	147
5.7. Задания для самостоятельной работы.....	151
Глава 6. БЛОЧНОЕ ПРОГРАММИРОВАНИЕ.....	156
6.1. Метод декомпозиции Данцига - Вулфа	156
6.2. Решение транспортной задачи методом Данцига-Вулфа	160
6.3. Задания для самостоятельной работы.....	167
Глава 7. ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ	169
7.1. Проблема целочисленности	169
7.2. Метод отсечений	172
7.3. Метод ветвей и границ	180
7.4. Аддитивный алгоритм.....	187
7.5. Другие методы.....	192

7.6. Задания для самостоятельной работы.....	193
Глава 8. НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ.....	195
8.1. Характеристика задач.....	195
8.2. Условия оптимальности.....	197
8.3. Квадратичное программирование.....	198
8.4. Сепарабельное программирование (СП).....	203
8.5. Задачи дробно-линейного программирования.....	208
8.6. Методы "спуска".....	211
8.7. Методы одномерной минимизации.....	212
8.7.1. Метод деления шага пополам.....	212
8.7.2. Квадратичная аппроксимация.....	213
8.7.3. Метод деления интервала пополам.....	214
8.7.4. Метод золотого сечения.....	215
8.7.5. Метод Фибоначчи.....	216
8.7.6. Метод первого порядка.....	218
8.7.7. Методы второго порядка.....	218
8.8. Многомерный поиск безусловного минимума.....	219
8.8.1. Метод Гаусса-Зейделя (покоординатного спуска).....	219
8.8.2. Метод Хука-Дживса (метод конфигураций).....	220
8.8.3. Симплексный метод.....	222
8.8.4. Градиентные методы.....	226
8.8.5. Метод Ньютона.....	230
8.8.6. Методы сопряженных направлений.....	231
8.8.7. Методы случайного поиска.....	233
8.8.8. Генетические алгоритмы.....	236
8.9. Методы условной оптимизации.....	240
8.9.1. Метод проектирования градиента.....	240
8.9.2. Метод штрафных функций.....	244
8.9.3. Метод барьерных функций.....	248
8.9.4. Другие методы условной оптимизации.....	251
8.10. Задания для самостоятельной работы.....	252
Глава 9. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ	255
9.1. Как работает метод ДП	256
9.2. Функциональное уравнение ДП	260
9.3. Распределение одного вида ресурса	263
9.4. Задача организации выпуска m видов продукции	268
9.5. Задача о кратчайшем пути	275
9.6. Задача с мультипликативным критерием	277
9.7. Усложненная задача	279
9.8. Многомерные задачи динамического программирования.....	282
9.9. Снижение размерности с помощью множителей Лагранжа	285
9.10. Задания для самостоятельной работы.....	289

Глава 10. МНОГОКРИТЕРИАЛЬНЫЕ ЗАДАЧИ	
ПРИНЯТИЯ РЕШЕНИЙ	308
10.1. Основы многокритериальной оптимизации.....	308
10.1.1. Многокритериальная задача математического	
программирования	309
10.1.2. Где искать оптимальное решение	310
10.1.3. Определения	311
10.1.4. Условия оптимальности	313
10.2. Методы многокритериальной оптимизации	316
10.2.1. Методы первой группы	317
10.2.1.1. Функция полезности.....	317
10.2.1.2. Решение на основе лексикографического упорядочения	
критериев	322
10.2.1.3. Метод главного критерия.....	323
10.2.1.4. Линейная свертка	324
10.2.1.5. Максиминная свертка.....	325
10.2.1.6. Метод идеальной точки.....	326
10.2.1.7. Целевое программирование (ЦП)	329
10.2.2. Интерактивные методы	331
10.2.2.1. Метод уступок.....	332
10.2.2.2. Интерактивное компромиссное программирование .	
334	
10.2.2.3. Метод STEM.....	340
10.2.2.4. Метод взвешенных метрик Чебышева.....	341
10.2.2.5. Прогрессивный алгоритм принятия многокритериаль-	
ных решений.....	347
10.2.3. Построение эффективного множества.....	350
СПИСОК ЛИТЕРАТУРЫ	355

ПРЕДИСЛОВИЕ

Вся сознательная жизнь человека связана с принятием решений. Каждый день человек принимает десятки и сотни решений, одни из которых носят личный характер, другие – технический, третьи затрагивают судьбы людей, интересы организации, региона или страны. Еще с 19 века наука стала заниматься проблемами организационного управления и принятия решений, главным образом в рамках философии. Но только в 20 веке формируются самостоятельные научные направления, относящиеся к этой сфере. Среди них **исследование операций**, которое определяют по-разному¹, но цель понимается одинаково – на основе математической модели количественно обосновать выбираемое решение. Для выбора наилучшего решения в рамках этого направления разработаны мощные математические методы.

Пособие содержит разделы, соответствующие дисциплинам «Теория принятия решений», «Системный анализ и исследование операций». В нем рассмотрены основные понятия и типичные задачи исследования операций (глава 1). Излагаются основные методы исследования операций, нацеленные на нахождение наилучших решений. Это методы математического программирования, такие как линейное, целочисленное, нелинейное, динамическое и другие (главы 4-9). Им предшествует краткая сводка по методам классического анализа (глава 3). При этом основной акцент делается на содержательных и прикладных аспектах рассматриваемых методов. Большое место отводится проблеме принятия решений по многим критериям, так как в учебной литературе она освещена мало (глава 10). Изложение материала сопровождается множеством примеров и иллюстраций. Приведены задания для самостоятельной работы студентов. Многие из них являются оригинальными и могут быть использованы для расчетных и курсовых работ.

Подзаголовок пособия подчеркивает, что в нем представлена только часть разделов теории принятия решений, относящаяся к формализуемым задачам.

Пособие предназначено студентам специальностей по направлению «Информатика и вычислительная техника», прежде всего бакалаврам очной и заочной форм обучения. Оно может быть полезно студентам экономических специальностей, а также инженерам и аспирантам, сталкивающимся с вопросами выбора наилучших решений.

¹ Большой энциклопедический словарь «Математика» (1998) определяет исследование операций как теорию принятия оптимальных решений.

Глава 1. ХАРАКТЕРИСТИКА ИССЛЕДОВАНИЯ ОПЕРАЦИЙ

*...особенную важность имеет
задача, общая для всей практики:
как располагать средствами своими
для достижения по возможности
большей выгоды?*

П.Л.Чебышев

1.1. Основные понятия и особенности исследования операций

Термин "операционные исследования", по-видимому, впервые применил в 1938 г. А.Раув, руководитель научной группы в Бодси (Англия), отнеся его к работам по оценке эффективности операций, проводимых военно-воздушными силами. Однако сегодня больше используют американский термин "исследование операций", имеющий тот же смысл.

Возникнув в недрах военных ведомств, новая наука, развиваясь, находит применение в самых разных областях человеческой деятельности, в том числе в бизнесе. Новые потребности общества стимулировали поиск новых подходов. Эту мысль хорошо выразил один из руководителей американской корпорации RCA Дж.Чейн: "Для того чтобы... эффективно действовать... управлять внутренними и внешними взаимодействиями... принимать своевременные и разумные решения, бизнес сейчас не может больше полагаться на традиционные методы и сбор информации; рамки успеха становятся слишком узкими, затраты слишком высокими, риск слишком большим, время слишком недостаточным, а конкуренция слишком сильной для того, чтобы действовать на основе предчувствия, интуиции или прошлой истории"¹.

В настоящее время исследование операций можно рассматривать как одну из важнейших дисциплин, связанных с принятием решений, или как составную часть системного анализа. Несмотря на длительный период развития суть исследования операций остается неизменной: всесторонний анализ операции, оценка последствий возможных решений, поиск наиболее эффективных вариантов достижения цели. Не обсуждая

¹ Курс для высшего управленческого персонала. М.,1970. С.114.

различные точки зрения на исследование операций, приведем здесь только одно краткое, но, на наш взгляд, емкое определение, отражающее его главное предназначение.

Исследование операций (ИСО) - это наука о количественном обосновании оптимальных решений.

При этом под оптимальным понимается решение, наилучшее в ***определенном смысле***. Нельзя говорить об оптимальном решении вообще, корректное применение этого понятия требует конкретизации его смысла и условий, в которых принимается решение.

В то же время ***"операция"*** - весьма широкое понятие: это есть совокупность действий или мероприятий, направленных на достижение определенной цели. В ИСО описание операции включает следующее.

1. Цель операции, то есть то, ради чего проводится операция.
2. Оперирующая сторона - лицо или группа лиц, преследующих поставленную цель. В сложных операциях оперирующая сторона состоит из лица, принимающего решение (ЛПР), и аналитиков - специалистов по исследованию операций. Физически ЛПР - это одно лицо или группа лиц, наделенных правом принимать решения и несущих за них ответственность. Подготовка решений ложится на аналитиков. Разница между первыми и вторыми не только в знаниях методологии и методов ИСО, но и в информированности об операции, что весьма существенно. Причины этого кроются в сложности извлечения и представления информации, которой владеет ЛПР, или в нежелании ЛПР раскрывать все карты. В простых случаях ЛПР и аналитик могут быть в одном лице.
3. Активные средства - это, как правило, ресурсы, используемые для достижения цели.
4. Способы действий, поведения или использования активных средств. Их называют решениями, альтернативами или стратегиями в зависимости от типа операции.
5. Результаты или исходы операции.
6. Тип связи между решениями (стратегиями) и исходами операции. Он зависит от условий, в которых протекает операция.

Возьмем простой пример операции - подготовка к экзамену. Цель операции - успешная сдача экзамена. Оперирующая сторона: студент или студент (ЛПР) и методист (аналитик). Активные средства: время до экзамена, конспекты, учебные пособия и др. Существует много способов распорядиться перечисленными средствами, в том числе и написание шпаргалок. Так как у нас применяется четырехбалльная система оценки, то возможны четыре исхода сдачи экзамена. Всей операции присущи элементы случайности: самочувствие студента во время подготовки и сдачи экзамена, невозможность одинаково освоить все вопросы, настроение преподавателя, лотерейный способ выдачи экзаменационного

билета и т.п. Поэтому между стратегиями подготовки к экзамену и его результатами связь неоднозначна.

Говоря об ИСО как о самостоятельном направлении, обычно отмечают его три основные особенности: системный подход, комплексный коллектив исследователей, применение научных методов. Под системным подходом понимается комплексная методология исследования сложных систем или проблем. В этой методологии определяющим является подход к любой части системы (проблемы) с позиции системы в целом, превалирование цели системы над целями ее подсистем. Как писал С. Янг, крупный специалист в области организации, "анализ отдельных свойств системы, конечно, важен, но он становится осмысленным только тогда, когда каждое свойство оценивается в контексте целостной системы"². Другое важнейшее требование системного подхода состоит в том, что необходимо стремиться выявить все существенные факторы и взаимосвязи, влияющие на достижение цели системы. Для этого приходится расширять первоначальный объект исследования, искать скрытые от первого взгляда связи между факторами и частями системы.

Вторая особенность ИСО обусловлена необходимостью изучения и анализа проблемы с разных точек зрения, стремлением выйти за рамки стереотипов. Именно поэтому с момента возникновения исследования операций группы исследователей состояли из специалистов разного профиля (военных, математиков, физиков, психологов и др.), объединенных единой методологией. Такое комплексное исследование позволяет расширять множество альтернатив и находить действительно наилучшее решение.

Применение научных методов присуще любой науке, но в ИСО они имеют свою специфику, которая обусловлена задачей исследования и количественным характером результатов. Чтобы яснее представить эту особенность ИСО, рассмотрим, как проводится операционное исследование.

1.2. Этапы операционного проекта

Не существует строгой регламентации хода и содержания операционного исследования, но в любом выполненном проекте можно выделить характерные для ИСО этапы разработки.

1. Постановка задачи. Она включает содержательное описание задачи: объект и цель исследования, внутренние и внешние условия, ресурсы, значения параметров или их оценки, возможные способы действий и возможные результаты, другую имеющуюся информацию. Эту работу выполняют совместно ЛПР и аналитик. После тщательного анализа

² Системное управление организаций. М:Сов.радио, 1972. С.455.

первоначальной постановки аналитик уточняет с ЛПР содержание задачи по всем аспектам и особо согласовывает показатель, который предлагается в качестве критерия оптимальности.

2. Построение математической модели. Характер задач исследования операций таков, что их решение не может проводиться путем натурального эксперимента или физического моделирования. Например, выбор места и мощности нового производства, определение оптимального плана выпуска продукции, формирование портфеля заказов немислимо производить путем реализации и сравнения различных вариантов. Такая ситуация в науке не нова: так в астрономии нельзя манипулировать небесными телами, но предсказывать положение планет солнечной системы возможно благодаря использованию математической модели. Модели, и в частности математические, широко применяются в различных областях. Математические модели исследования операций отличаются своей направленностью, которая отражается в структуре модели. Математическая модель в ИСО включает:

- зависимость критерия от управляемых и неуправляемых переменных;
- уравнения, отражающие связи между переменными, например, уравнения на основе материально-энергетических балансов;
- ограничения, обусловленные реальными условиями и требованиями к показателям и переменным (неотрицательность, целочисленность, комплектность, допустимые и/или директивные значения и т.п.). В конкретных задачах могут отсутствовать отдельные составляющие модели полностью или частично за исключением критериальной функции, которая должна быть в модели обязательно.

3. Проверка адекватности модели. Математическая модель представляет собой формализованную гипотезу исследователя о реальных взаимосвязях и поведении системы. Поэтому прежде чем использовать модель для прогнозирования последствий и выбора решений, необходимо убедиться в ее адекватности системе или операции с точки зрения поставленной цели исследования. Для "прозрачных" моделей может быть достаточной качественная проверка, в сложных моделях необходим количественный анализ. В последнем случае для моделирования поведения на модели используются численные методы (иногда это называют прямой задачей: по задаваемым входам нужно определить выходы). Для осуществляемых ранее операций проверка адекватности может производиться по ретроспективным данным (при отсутствии качественных изменений в операции). В других случаях проверка проводится путем наблюдения за реакцией модели и системы на одинаковые решения. При обнаружении неадекватности модель корректируется: при качественном совпадении повысить количественную адекватность можно путем уточнения коэффициентов модели, при более

серьезных расхождениях может потребоваться изменение и/или добавление ограничений и уравнений или даже построение другого вида модели. Следует заметить, что такая проверка невозможна для вновь разрабатываемых операций, и тогда приходится довольствоваться качественным тестированием модели.

4. Поиск оптимального решения на модели. Это центральный этап операционного исследования (с математической точки зрения - обратная задача). Он заключается в определении решения, оптимального в смысле принятого критерия. Для отыскания оптимального решения на математической модели применяются методы оптимизации, главным образом методы математического программирования. Развитие последних стимулировалось и продолжает стимулироваться все возрастающими потребностями исследования операций и системного анализа.

5. Анализ оптимального решения. Сюда входит анализ чувствительности полученного решения, параметрический и вариантный анализ, выявление альтернативных оптимальных решений и др. Анализ чувствительности критерия к отклонению переменных от их оптимальных значений позволяет определить разумные требования к точности реализации оптимального решения. Результаты параметрического и вариантного анализа показывают, каким будет оптимальное решение при изменении коэффициентов модели, состава ограничений или при изменении критерия. При этом может устанавливаться значимость отдельных элементов модели, то есть их влияние на оптимальное значение критерия. В случае неединственности оптимального решения появляется дополнительная возможность выбора по показателю, который не представлен в критерии. Важное место в анализе решения отводится интерпретации полученных результатов в терминах предметной области ЛПР.

6. Внедрение результатов исследования. Здесь главное требование состоит в необходимости непосредственного участия разработчиков на всех стадиях реализации предлагаемых решений.

Таким образом, применение научных методов в ИСО отличается всесторонним количественным исследованием, основанным на математической модели и ставящим своей целью определение оптимального решения в интересах ЛПР.

1.3. Критерий оптимальности

Термин "критерий" широко используется как во всех областях знаний, так и в обыденной жизни в интуитивно понятном смысле. Ввиду особой важности этого термина для исследования операций дадим краткие пояснения. Греческое слово *kriterion* означает мерило, оценку, средство для суждения. Именно в этом смысле используется понятие

критерия в ИСО. Поставленная в операции цель может быть достигнута по-разному и в разной степени в зависимости от принимаемых решений. Критерий есть тот показатель, который характеризует (оценивает) эффективность решений с точки зрения достижения цели, а следовательно, позволяет выбрать среди них наилучшее. В ИСО применяют равнозначные термины: критерий оптимальности, критерий эффективности, целевая функция. Последний термин подчеркивает неразрывную связь критерия с целью. Таким образом, решение может быть оптимальным только в смысле конкретного критерия в пределах адекватности используемой модели.

В исследовании операций к критерию предъявляются определенные требования. Наиболее важные из них следующие.

1. Критерий должен быть **количественной** и **неслучайной** величиной.
2. Критерий должен правильно и полно отражать поставленную цель. Его можно рассматривать как количественную модель качественной цели.
3. Критерий должен иметь простой и понятный ЛПР физический смысл.
4. Критерий должен быть чувствителен к управляемым (искомым) переменным.

При исследовании действующих систем к критерию могут предъявляться дополнительные требования, такие как измеримость, статистическая однозначность, статистическая эффективность и др.

Многочисленные примеры из практики показывают огромную важность правильного выбора критерия оптимальности. Из истории второй мировой войны известен случай неверного выбора критерия для оценки эффективности мероприятий по охране караванов судов, доставляющих грузы в северные советские порты (об одном из таких караванов написан роман В. Пикуля). С целью защиты караванов от воздушных налетов немцев на судах стали устанавливать зенитные системы. Через некоторое время решили оценить эффективность принятых мер, чтобы определить дальнейшие действия, и в качестве критерия взяли число самолетов противника, сбиваемых установленными зенитными системами. Этот показатель оказался очень низким, что объяснялось непрофессиональностью орудийных расчетов, а также отсутствием стабилизационных платформ на используемых судах. Исходя из такой оценки, предлагалось демонтировать зенитные орудия, передав их береговым батареям, и искать другие способы защиты караванов. Но вовремя спохватились, поняв, что принятый критерий не отражает поставленную цель, которая заключается в повышении живучести судов, а не в уничтожении самолетов противника (поражение самолетов - это только одно из средств). Достижение такой цели с помощью рассматриваемого мероприятия должно определяться по проценту судов, приходящих в порты назначения. Проведенный анализ показал, что для караванов с зенитными системами этот критерий значимо

вырос и, следовательно, предложенный способ эффективен. А объясняется данный феномен тем, что при стрельбе из зенитных орудий немецкие летчики боялись приближаться к судам и бомбометание производили с больших высот и расстояний, что значительно снижало эффективность налетов.

В истории советского периода немало примеров, когда нарушение требований к выбору критерия приводило к печальным последствиям. Это прежде всего критерии развития экономики, например, пресловутый вал, который привел к тому, что мы выпускали самое тяжелое и энергоемкое оборудование, больше всех металла, угля, тракторов и, в то же время использовали их с самой низкой эффективностью и т.п. А сколько человеческих судеб искалечено из-за тех критериев, которые применяла КПСС к людям?!

Множество показателей, которые в ИСО используются в качестве критериев, можно условно разделить на ряд групп: социальные (среднедушевой доход, обеспеченность жильем и т.п.), экономические (прибыль, рентабельность, себестоимость и др.), технико-экономические (производительность, урожайность и др.), технико-технологические (прочность, чистота материала, другие физические или химические показатели), прочие. Они приведены в порядке убывания глобальности применения: первые применяются в системах более высокого уровня (страна, регион, предприятие), последние - в основном на уровне процесса, объекта.

Однако во многих случаях не удастся полностью отразить поставленную цель одним критерием и тем более это невозможно, когда в операции преследуется более одной цели. Например, цели типа повышение уровня жизни, улучшение экологической обстановки и т.п. нельзя "покрыть" одним критерием. В таких ситуациях вводится несколько показателей, характеризующих достижение цели. Как правило, оптимальные решения, получаемые по разным показателям-критериям, не совпадают, что создает неопределенность в выборе окончательного решения. Задачи, в которых приходится определять наилучшее решение по нескольким критериям, называются **многокритериальными** или задачами векторной оптимизации. Они составляют особый и более сложный класс задач исследования операций, который рассмотрен в последней главе настоящего пособия.

1.4. Виды математических моделей ИСО

Не останавливаясь на классификации моделей, в том числе и математических, рассмотрим их только в одном аспекте, который обуславливает принципиальные различия математических моделей и методов отыскания на них оптимальных решений.

Вид модели определяется типом связи между решениями (альтернативами, стратегиями) и результатами, который в свою очередь зависит от условий, в которых протекает операция и приходится принимать решения.

1. Решения принимаются в условиях определенности. Это значит, что каждому решению можно поставить в соответствие (пусть даже путем сложных расчетов) определенный результат, то есть имеет место детерминированный тип связи. Модели, описывающие такие ситуации, называются **детерминированными**. Этот тип модели на практике применяется наиболее широко, так как он "удобен в работе". По этой причине такие модели часто используют в качестве первого приближения и в условиях, отличающихся от ситуации определенности.

Приведем простой пример детерминированной модели. Пусть в пункте А, возле которого проходит прямая дорога, расположена пожарная часть, а на лугу в точке С - некоторое строение (рис.1.1). В случае возгорания строения пожарная машина должна быстро прибыть к месту пожара. Известны расстояния АВ и ВС и скорости движения машины по дороге v_1 и по лугу v_2 . Требуется определить кратчайший путь движения машины. Если правомерно допущение об отсутствии влияния на скорость машины каких-либо случайных факторов, то описанная ситуация характеризуется полной определенностью. Очевидно, что оптимальный маршрут машины надо искать в классе ломаных линий, включающих не более двух отрезков прямых (любой другой путь будет заведомо хуже). Такой путь полностью определяется точкой излома x - расстоянием от пункта А до места съезда машины с дороги. Выбрав в качестве критерия оптимальности время движения машины, можем представить математическую модель операции в виде

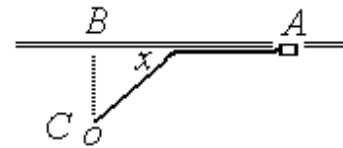


Рис.1.1

$$T = \frac{x}{v_1} + \frac{\sqrt{BC^2 + (AB - x)^2}}{v_2} \rightarrow \min, \quad 0 \leq x \leq AB.$$

Как видно из модели, каждой альтернативе в выборе маршрута (значению x) ставится в соответствие его показатель T . Детерминизм данной модели отражает определенность ситуации.

2. Решения принимаются в условиях риска. Между решениями и результатами имеет место стохастическая связь: определенному решению может соответствовать более одного результата, вероятности появления которых известны. Адекватным отображением таких условий являются **вероятностные (стохастические)** модели. Если под результатом имеется в виду значение критерия, то исходная постановка задачи (и модель!) некорректна: нельзя максимизировать или минимизировать случайную величину. В этом случае в качестве критерия следует выбирать не

исходный показатель, а одну из его вероятностных характеристик, например, математическое ожидание или дисперсию. Неоднозначность обусловлена наличием случайных факторов. Но осреднение случайных аргументов и осреднение результатов, на которые первые влияют, далеко не всегда одно и то же. Это объясняется тем, что в общем случае не выполняется равенство

$$M[f(x_1, x_2, \dots, x_n)] = f[M(x_1), M(x_2), \dots, M(x_n)], \quad (1.1)$$

где x_j - случайные величины; M - знак математического ожидания.

Рассмотрим пример такой ситуации. Пусть фирма "Апельсин" постоянно занимается продажей фруктов. Для простоты будем считать, что поставка и продажа фруктов осуществляется целыми контейнерами, а единицей времени является неделя. Спрос на фрукты C колеблется случайным образом, но вероятность спроса в случайно взятую неделю $P(C)$ известна. При заключении договора с поставщиком на очередной период фирма должна определить наиболее выгодное для нее количество контейнеров, которое будет поставляться еженедельно, если известны прибыль от реализации одного контейнера d и убыток b при его невостребовании. Так как спрос случаен, то и результат - доход за неделю D , для фиксированного числа заказываемых контейнеров n будет случайной величиной: в случае, когда спрос превысит предложение, то есть при $C > n$,

$$D = dn, \quad (1.2)$$

если же предложение окажется выше спроса ($C \leq n$), доход

$$D = dC - (n - C)b. \quad (1.3)$$

Таким образом, доход D является функцией управляемой величины n и случайного фактора C . Очевидно, что максимизация такого показателя бессмысленна. В качестве критерия оптимальности разумно взять математическое ожидание дохода за неделю, так как его максимизация обеспечит максимум дохода за весь период. Поскольку вероятность появления случаев (1.2) и (1.3) определяется $P(C)$, модель задачи будет иметь вид

$$M[D] = \bar{D} = \sum_{C=0}^n \{dC - (n - C)b\}P(C) + \sum_{C=n+1}^{\infty} dnP(C) \rightarrow \max, \quad (1.4)$$

$$n \geq 0, \text{ int}, \quad (1.5)$$

где int означает "целое". При составлении этой модели в явном виде учитывалась стохастичность ситуации и, следовательно, принимаемые по ней решения в такой же степени учитывают фактор случайности. Упрощенное представление операции может базироваться на аппроксимации реальной ситуации детерминированной. В этом случае спрос рассматривается как неслучайная величина, равная его математическому ожиданию \bar{C} . При этом доход

$$D = \begin{cases} dn, & n < \bar{C}, \\ \bar{C}d - (n - \bar{C})b, & n \geq \bar{C} \end{cases}$$

также неслучаен. На такой модели оптимальное решение, максимизирующее D , определяется просто: $n^0 = \bar{C}$.

Чтобы показать отличие результатов при использовании упрощенной модели и модели (1.4), произведем расчет для исходных данных $d=30$, $b=5$ и вероятности спроса:

C	0	1	2	3	4	5	6
$P(C)$	0	0.1	0.25	0.3	0.25	0.1	0

Вычисляем средний спрос: $C = \sum_{c=0}^6 C \cdot P(C) = 3$. Тогда по упрощенной

модели получим: $n^0=3$, $D=90$. Такой доход имел бы место при детерминированном и неизменяемом уровне спроса. Но при случайном спросе величина $D=90$ будет достигаться только в те недели, когда спрос окажется не меньше 3, а в другие недели доход будет ниже и, следовательно, средний доход за весь период станет меньше 90. Чтобы показать это и одновременно определить оптимальное число контейнеров при случайном спросе, вычислим значения среднего дохода по модели (1.4) при всех возможных n :

n	1	2	3	4	5
\bar{D}	30	56.5	74.25	81.5	80

По результатам вычислений видно, что решение $n^0=3$, полученное на детерминированной модели, не обеспечивает максимального среднего дохода. Кроме того, видно, что в условиях случайного спроса оптимальным является решение $n^*=4$, при котором средний доход составляет 81.5 против 74.25 при $n^0=3$. Это пример операции, для которой не выполняется равенство (1.1), хотя случайный фактор имеет симметричное распределение. Судя по разнице результатов на двух моделях, в данной операции стохастичность оказывает значимое влияние и поэтому ее нельзя не учитывать.

Однако наличие случайных факторов не всегда влечет за собой неоднозначность результатов. Возможны случаи, когда элементарные составляющие процесса или системы ведут себя случайно, а результаты системы в целом не случайны. Характерным примером такой системы является идеальный газ, поведение которого подчиняется детерминированному закону Бойля–Мариотта. Неслучайное поведение на макроуровне при наличии элементов случайности на микроуровне называют **стохастическим детерминизмом**.

3. Решения принимаются в условиях неопределенности. Это ситуация, противоположная первой рассмотренной. Природа неопределенности может быть различной, но в общем случае она проявляется в том, что определенному решению соответствует более одного результата, а вероятностные характеристики результатов неизвестны. Математические модели, описывающие неопределенный тип связи, разнообразны и не имеют единого названия. В частности, к этому классу относятся матричные модели, модели типа "игра", "аукционный торг", нечеткие модели.

Во многих случаях ситуацию неопределенности можно представить (или аппроксимировать) матрицей вида

Альтернативы	Состояние среды			
	Q_1	Q_2	\dots	Q_n
A_1	u_{11}	u_{12}	\dots	u_{1n}
A_2	u_{21}	u_{22}	\dots	u_{2n}
\dots	\dots	\dots	u_{ij}	\dots
A_n	u_{m1}	u_{m2}	\dots	u_{mn}

(где u_{ij} - результат (исход) выбора альтернативы A_i при условии, что среда окажется в состоянии Q_j ; u_{ij} может иметь смысл прибыли, дохода, выигрыша или затрат, проигрыша, убытков и т.п.).

Прежде чем выбирать решение на этой модели, нужно определиться с принципом оптимальности, на основе которого будут сравниваться альтернативы, так как только одно желание ЛПР получить наилучший результат не дает такой основы. Принцип оптимальности зависит от точки зрения на ситуацию ЛПР, его отношения к риску, от предположений относительно поведения среды. Наиболее характерной гипотезой поведения среды является представление, что среда ведет себя наихудшим образом ("как назло"). Это самый пессимистический взгляд на ситуацию, свойственный ЛПР, не склонному к риску. В этом случае выбор решения основывается на **принципе гарантированного результата** (иногда его называют критерием Вальда). Он состоит в том, что эффективность каждой альтернативы оценивается наихудшим из исходов, возможных при выборе данной альтернативы. Такой результат гарантируется, то есть будет не хуже, при любом фактическом состоянии среды. Теперь очевидно, что наилучшим решением в смысле принятого принципа оптимальности будет выбор той альтернативы, которая имеет наилучший гарантированный результат. Так, если u_{ij} имеет смысл прибыли, то оценкой i -й альтернативы является $\min_j u_{ij}$, а оптимальной будет альтернатива, максимизирующая эту величину, то есть A_i , на которой достигается

$\max_i \min_j u_{ij}$. Применительно к этой ситуации принцип гарантированного результата называют принципом **максимина**, а оптимальную альтернативу - **максиминной**. В условиях неопределенности только этот принцип имеет объективное обоснование и дает абсолютно надежную оценку.

Другой подход, называемый также критерием Сэвиджа, использует аналогичный прием, но по отношению к преобразованной матрице - матрице риска (сожалений) $[r_{ij}]$, где $r_{ij} = \max_i u_{ij} - u_{ij}$, то есть риск - это разность между максимально возможным выигрышем при j -м состоянии среды и выигрышем при выборе i -й альтернативы в условиях незнания о фактическом состоянии среды. Так как цель состоит в уменьшении риска, то, используя принцип **минимакса** (гарантированного риска), определим как оптимальную ту альтернативу, на которой достигается $\min_j \max_i u_{ij}$, и тогда риск не превысит этой величины ни при каком состоянии среды.

Гибкий учет отношения ЛПР к риску возможен с помощью критерия Гурвица. Если исходы имеют смысл дохода, то оптимальная альтернатива определяется из условия

$$\max_i \{ \gamma \max_j u_{ij} + (1 - \gamma) \min_j u_{ij} \},$$

где $\gamma = [0,1]$ - коэффициент риска. При ориентации на самое худшее $\gamma = 0$, что соответствует критерию Вальда, а для крайнего оптимиста $\gamma = 1$. Промежуточные значения γ отражают разный уровень риска ЛПР. Возможны и другие подходы к выбору оптимальных решений в условиях неопределенности, но все они, как и последние два, не гарантируют достижение расчетных результатов.

Как следует из вышерассмотренного, выбор вида модели требует от исследователя интуиции и опыта наряду с глубокими знаниями моделируемой области. Следует особо отметить, что построение модели основывается на представлениях аналитика, которые могут не соответствовать реальным связям в большей или меньшей степени. При этом большое значение имеют оценка влияния случайных факторов, факторов неопределенности, уровень агрегирования, допустимая сложность модели. Так, нередко возникает дилемма: построить высокоточную, но очень сложную модель, на которой можно будет получить только приближенное к оптимальному решение, либо поступиться точностью моделирования и иметь возможность применять на модели точные методы оптимизации. Какое решение окажется ближе к истинному оптимальному, заранее сказать невозможно. К сожалению, не существует готовых рецептов построения математических моделей. Это один из этапов операционного исследования, который, следуя Саати, можно отнести к области искусства.

1.5. Классы операционных задач

В настоящее время исследование операций находит применение почти во всех областях целенаправленной человеческой деятельности. Отсюда широкий диапазон математических моделей операций и методов их исследования. В то же время исследование операций развивалось и продолжает развиваться по определенным направлениям, которые различаются типами задач, и появление новых направлений обуславливается возникновением новых задач. В классическом исследовании операций выделяют классы типичных задач, рассмотрение которых позволяет лучше представить круг проблем ИСО. Наиболее характерными классами операционных задач являются:

- задачи управления запасами;
- задачи распределения;
- задачи массового обслуживания;
- задачи выбора маршрута;
- задачи замены;
- задачи упорядочения;
- задачи сетевого планирования и управления;
- состязательные задачи;
- задачи поиска.

Приведем краткую характеристику и постановку перечисленных задач.

Задачи управления запасами. Под запасами понимают неиспользуемые в данный момент ресурсы. К ним относятся материалы, оборудование, полуфабрикаты, готовая продукция, работники, финансовые средства и т.п. Проблема запасов заключается в поиске ответов на два основных вопроса: 1) сколько заказывать (закупать или производить), 2) когда или как часто заказывать. Нетривиальность этой задачи обусловлена тем, что с запасами связаны статьи затрат, по-разному изменяющиеся с изменением уровня запасов. С увеличением запасов растут затраты на хранение (складские расходы, замораживание оборотных средств, потери от порчи и старения, морального износа и т.п.) и одновременно уменьшаются затраты из-за возможной нехватки запасов (простоев производства, аварий, штрафов и др.). Кроме того, при росте объема партии снижаются затраты на подготовительно-заключительные операции, так как они не зависят или слабо зависят от величины партии. В конкретных приложениях есть и другие статьи затрат, требующие учета. В результате задача состоит в выборе таких параметров управления запасами (объема партии, периода пополнения и др.), которые обеспечивают минимум суммарных затрат, связанных с запасами.

Многие практические задачи могут ставиться как задачи управления запасами. Например, проблема пополнения штата стюардесс имеет прямое

отношение к рассматриваемой задаче: избыток стюардесс вызывает дополнительные расходы на их обучение и содержание, отсутствие стюардессы на рейсе приводит к задержке или отмене рейса, а следовательно, к большим потерям. Аналогичные вопросы возникают при выборе мощности устанавливаемого оборудования, разбиении всего заказа на партии в процессе производства, определении числа машин, закрепляемых за комбайном и других подобных ситуациях.

Задачи управления запасами имеют обширную классификацию. Выделим здесь три основных признака классификации: по виду спроса различают задачи со случайным и неслучайным спросом, по способу пополнения запасов - мгновенное и с задержкой, по виду запасов - однородные и неоднородные. Очевидно, что самыми простыми являются задачи с неслучайным спросом, с мгновенным пополнением и однородными запасами. Другую крайность составляют задачи со случайным спросом на неоднородные запасы, пополняемые со случайной задержкой.

Задачи распределения на практике встречаются наиболее часто. Они возникают в ситуациях, когда имеется ряд работ, операций или потребителей, нуждающихся в выполнении или удовлетворении, и возможны различные способы или пути их осуществления. В зависимости от уровня ресурсов различают три группы задач распределения. Для первой группы задач характерно, что наличных ресурсов достаточно для выполнения всех работ, но не хватает для выполнения каждой работы оптимальным образом. В этих условиях необходимо так распределить ресурсы между всеми работами (потребителями), чтобы достигалась наибольшая эффективность системы в целом. Показатель эффективности (критерий) определяется конкретной целью системы.

Классическим примером такой задачи является сбалансированная **транспортная задача**. С одной стороны, имеются поставщики с известными количествами груза, с другой - потребители с известными потребностями в грузе, при этом сумма потребностей равна сумме возможностей (баланс). Кроме того, для всех пар поставщик-потребитель даны затраты на перевозку единицы груза от поставщика к потребителю. Очевидно, что наилучшим вариантом удовлетворения отдельного потребителя будет поставка груза с минимальными для него затратами, однако это может привести к значительному возрастанию транспортных затрат у других потребителей. Поэтому задача состоит в определении такой схемы перевозки, при которой суммарные транспортные издержки будут минимальны. Разновидность транспортной задачи, называемая **задачей о назначениях** или задачей выбора, заключается в распределении N кандидатов по N должностям, работам или машинам при известной эффективности каждого кандидата на каждой должности с целью достижения максимальной эффективности всей системы (например,

расстановка 10 спортсменов по 10 этапам эстафеты, обеспечивающая минимальное время прохождения всех этапов).

Ко второй группе относят задачи, в которых наличных ресурсов недостаточно для выполнения всех работ или удовлетворения всех потребителей в полном объеме. Задача ставится аналогично вышеприведенной, но в ряде случаев требуется дополнительная информация о влиянии неудовлетворенного спроса на показатель эффективности, а решение должно содержать данные о том, какие работы и в каком объеме не выполняются в условиях оптимального распределения. Примерами подобных задач могут служить несбалансированная транспортная задача, большинство задач составления бюджета, задачи планирования, проектирования, составления смесей и др. Сюда же можно отнести задачу о рюкзаке, заключающуюся в наилучшем наборе предметов при ограниченном весе и/или объеме рюкзака (эта задача имеет важные инженерные приложения, связанные с оптимальным использованием ограниченных объемов стоек, отсеков, памяти и т.п.).

Задачи третьей группы отличаются тем, что уровень (объем) используемых ресурсов не фиксирован и может варьироваться в некоторых пределах. При этом затраты на ресурсы зависят от их объемов. Задача состоит в определении оптимального уровня ресурсов и оптимального распределения по критерию, учитывающему как затраты на ресурсы, так и эффективность их использования. В качестве примера можно привести проблему использования кредитов предприятием, которая особенно обостряется при высоких процентных ставках.

Задачи массового обслуживания возникают, когда имеет место поток заявок, требований или клиентов, нуждающихся в обслуживании. В общем случае заявки приходят через случайные промежутки времени и продолжительность обслуживания одной заявки также случайна. Системы, занятые обслуживанием таких потоков, называют системами массового обслуживания (СМО). Обычно в СМО выделяют 4 составляющих: поток заявок (входящий поток), очередь заявок, обслуживающие устройства, выходящий поток. Как следует из характера поступления и обслуживания заявок, в СМО может возникать как очередь заявок на обслуживание, так и очередь обслуживающих устройств (простои). Очевидно, что с любой очередью связаны потери. Различают СМО с отказами, когда при занятых устройствах пришедшая в систему заявка получает отказ и, следовательно, очереди заявок быть не может, и системы с ожиданием (с очередью), когда при занятых устройствах обслуживания заявка встает в очередь и ожидает обслуживания.

Процессы, протекающие в СМО, носят стохастический характер и поэтому для их описания применяют вероятностные модели. В качестве показателей эффективности работы СМО могут использоваться вероятность отказа, средняя длина очереди, среднее время пребывания

заявки в системе, пропускная способность (абсолютная или относительная), среднее число занятых устройств и др. В широком смысле задача массового обслуживания состоит в определении оптимальной структуры и оптимальных параметров СМО, а в узком - в выборе оптимальных параметров составляющих системы в пределах заданной структуры. В одних задачах за критерий принимают один из показателей работы СМО при ограничениях на другие показатели и на затраты на СМО, в других стремятся минимизировать затраты при обеспечении заданных уровней показателей работы СМО.

Задачи массового обслуживания особенно характерны для сферы услуг и производства. Примерами могут служить задачи организации торговли, медицинского обслуживания, ремонта, серийного и массового производства, работы вычислительного центра и отдельной ЭВМ в многозадачном и многопользовательском режимах и т.п.

Задачи выбора маршрута. В зависимости от вида искомого маршрута различают два варианта задач. Первый тип задач иногда называют задачами о кратчайшем пути. Между двумя заданными пунктами или узлами имеется конечное множество путей, состоящих из переходов (дуг), соединяющих промежуточные пункты. Один переход может входить более чем в один путь. Каждый переход характеризуется показателем или рядом показателей, в качестве которых могут быть время, длина, стоимость, расход ресурса и т.п. Требуется найти путь, кратчайший в смысле принятого критерия. Отметим, что искомым является незамкнутый путь и не требуется прохождение всех промежуточных пунктов. При детерминированных показателях используется детерминированная модель, а при случайных показателях - вероятностная модель. Такие задачи возникают при прокладке маршрутов различных транспортных средств, трасс дорог, линий электропередач, трубопроводов, выборе вариантов поведения на заданном конечном промежутке времени, расчете сетевых графиков и т.д.

Второй тип задачи выбора маршрута называется **задачей коммивояжера**. Такое название сложилось исторически и связано с поиском оптимального маршрута передвижения представителя торговой фирмы - коммивояжера. Последний, выйдя из своего города, должен обойти все города, входящие в сферу обслуживания фирмы, и вернуться обратно. При этом каждый город посещается один раз. Известны показатели переходов между всеми парами городов и требуется найти маршрут, обеспечивающий минимальные затраты времени или минимальный расход горючего, или минимальную стоимость проезда. Если показатели переходов зависят от направления движения, задача является *асимметричной*, иначе - симметричной. Эта задача отличается замкнутостью искомого маршрута и необходимостью прохождения *всех*

пунктов. В теории графов такой маршрут называют гамильтоновым циклом.

Первоначально задача коммивояжера рассматривалась как математическая головоломка, но в последние десятилетия обнаружили, что к ней сводится ряд важных практических проблем. Например, если на одном оборудовании каждый месяц нужно производить фиксированную номенклатуру изделий, а затраты на переналадку зависят от предшествующего и последующего видов изделий, то определение последовательности запуска изделий в производство, обеспечивающей минимальные затраты на переналадку в течение месяца, представляет собой типичную задачу коммивояжера. Другой пример: составляется программа для станка с ЧПУ на сверление нескольких десятков отверстий в плате и требуется определить порядок, в котором будет производиться сверление, так, чтобы общее время операции было минимальным. Совершенно очевидно, что это тоже задача коммивояжера. В чем трудности решения таких задач, если число возможных вариантов всегда конечно? При трех городах имеется два варианта решения, при четырех – уже шесть, а при 11 – более 3,6 млн. В общем случае задача коммивояжера с n городами (пунктами) имеет $(n-1)!$ замкнутых маршрутов, проходящих через все пункты. Таким образом, в реальных задачах число возможных вариантов исчисляется астрономическими величинами и в этом заключается основная проблема решения задачи.

Рассматриваемая задача может быть обобщена на m коммивояжеров. В базовом городе находится m коммивояжеров, обслуживающих всех клиентов фирмы (клиент-город). Необходимо составить m замкнутых маршрутов, охватывающих всех клиентов по одному разу и имеющих наилучший суммарный показатель, например, минимальную суммарную длину. Увеличение числа коммивояжеров повышает сложность задачи.

Задачи замены оборудования. В зависимости от типа оборудования различают два вида задач замены. В одних задачах оборудование рассматривается как единое целое, характеристики которого с течением времени ухудшаются. В результате снижается производительность, качество выполнения операций, возрастают затраты на эксплуатацию и текущие ремонты, снижается фонд полезного времени работы оборудования. С увеличением частоты замены оборудования все эти показатели будут улучшаться, но одновременно будут резко возрастать капитальные затраты и затраты, обусловленные демонтажем старого и монтажом нового оборудования. Поэтому стоит вопрос об определении моментов замены, наилучших в смысле выбранного критерия. В качестве последнего часто рассматривается прибыль, получаемая на оборудовании за определенный период времени. Вид модели замены напрямую зависит от характера изменения свойств оборудования. К оборудованию первого

типа можно отнести обрабатывающий центр, самолет, доменную печь, локомотив, пресс и т.п.

Другой вид задач замены связан с оборудованием, состоящим из большого числа относительно недорогих элементов, характеристики которых практически не меняют своих свойств, но могут внезапно полностью выходить из строя. Моменты выхода из строя, как правило, случайны. Примерами подобного оборудования могут служить электронные системы, компьютеры и др. В отличие от задач первого вида здесь наряду с моментами замены нужно определять и уровень, на котором следует проводить замену. Можно заменять отдельный элемент, плату, узел или целый блок. При этом, если уменьшается время на замену, то растет стоимость заменяющих частей и наоборот. Поэтому задача не имеет тривиального решения. Очевидно, что для математического описания подобных задач используются в основном вероятностные модели.

Задачи упорядочения возникают в связи с тем, что конечное множество независимых работ (операций) выполняется на одной группе оборудования, включающей два и более станков (обслуживающих устройств). Каждой паре операция-станок ставится в соответствие некоторый показатель. Задача заключается в определении такой последовательности выполнения независимых работ на одном и том же оборудовании, при которой достигается наилучшее значение критерия оптимальности.

В качестве примера рассмотрим классическую задачу Джонсона. Каждая из n деталей обрабатывается на m станках в одинаковом порядке, время обработки известно. Требуется определить очередность запуска деталей на обработку, обеспечивающую минимальное время обработки всех деталей. Для двух станков и двух деталей возможны только два варианта последовательности обработки, которые можно представить в виде графиков Ганта:

Станки	Время обр. деталей	
	A	B
1	5	4
2	2	5



Нетрудно подсчитать, что $T_{AB}=14$, а $T_{BA}=11$ и, следовательно, оптимальной является очередность BA.

Однако при двух станках и 10 деталях число вариантов уже превысит 3,6 млн., а для n деталей оно составляет $n!$. Поэтому простым перебором вариантов задачу не решить.

Проблема еще более усложняется с увеличением числа станков. В общем случае в оптимальном решении очередность деталей на разных станках может быть неодинаковой, что значительно увеличивает число вариантов решения. Правда, теоретические исследования дали интересный результат: в оптимальном решении очередность одинакова на первых двух и на последних двух станках (но между собой эти очередности не совпадают). Отсюда следует, что при двух и трех станках очередность на всех станках одинакова. Но для большего числа станков это неверно. Так, например, для 5 станков возможны 3 различные очередности обработки и, значит, при 10 деталях число вариантов составит $(10!)^3 = 4,78 \cdot 10^{19}$. Такие значения трудно представить.

Очевидно, что класс задач упорядочения достаточно широк. В него входят разнообразные задачи составления расписаний. Как и задачи выбора маршрута, рассмотренные задачи относятся к *комбинаторным*, сложность решения которых обусловлена их дискретностью и экспоненциальным ростом числа вариантов с увеличением размерности задачи.

Задачи сетевого планирования и управления. Одна из первых задач этого класса была поставлена и решена применительно к американской программе разработки ракет «Поларис». Программа охватывала огромное множество работ и большое число фирм-исполнителей, для координации которых требовались новые подходы. Таким образом, в отличие от задач упорядочения в задачах сетевого планирования рассматривается комплекс *взаимосвязанных* работ. Исходным является список работ, подлежащих выполнению, с известными продолжительностями и непосредственно предшествующими работами.

Взаимосвязи работ моделируются ориентированным графом, называемым сетевым графиком или просто сетью. Возможны два варианта представления сети: 1) сеть типа "работы-дуги", когда дуга отображает работу с присущими ей параметрами (показателями) и связями, а вершины - состояния объекта (программы, проекта), к которому относятся работы; 2) сеть типа "вершины-работы", когда дуги показывают только связи, а работам ставятся в соответствие вершины. Для первоначального построения сети удобнее второй вариант, но для расчетов чаще используется сеть "работы-дуги". Существуют простые алгоритмы перехода от одного представления к другому. Сеть может быть детерминированной и вероятностной. Во втором случае обычно случайным является время выполнения работ или ряд работ альтернативны с известными вероятностями необходимости их выполнения. С работами может быть связано время (простейшие сети), ресурсы, стоимость или их сочетания. Сеть может применяться один раз или многократно. По ходу выполнения работ сеть может корректироваться. В задачах сетевого планирования и управления различают анализ и синтез сети.

Анализ сети состоит в расчете времен начала и окончания работ, ранних и поздних сроков наступления событий, резервов времени работ и событий, определении *критического пути* и критических работ. Критическим называется самый длинный путь от начального события к конечному, то есть это минимальное время, за которое могут быть выполнены все работы. Под синтезом сети обычно понимают ее оптимизацию. Например, в пределах выделенных ресурсов или затрат нужно обеспечить минимальное время завершения всего комплекса работ, или наоборот, выполнить все работы к заданному сроку с минимальными затратами. На сети со случайными продолжительностями работ за критерий или основное ограничение принимается математическое ожидание длины критического пути. Кроме того, в качестве критерия может выступать вероятность завершения комплекса в заданный срок, которую следует максимизировать.

Рассмотренный класс задач характерен для больших научно-исследовательских работ, разработок сложных проектов, монтажно-строительных работ и других комплексных проектов и программ.

Состязательные задачи. Этими задачами моделируется принятие решений в ситуациях неопределенности, причем неопределенность обусловлена наличием конфликта. *Конфликт* имеет место, если в операции участвуют две и более оперирующих сторон, преследующих несовпадающие цели. Неопределенность у одной из сторон возникает в связи с неизвестностью линии поведения других сторон, в то время как результат зависит от поведения всех участников операции.

Различают две модели конфликта: *игру* и *аукционный торг*. Игра характеризуется известным количеством участников, называемых *игроками*, правилами игры, множеством возможных ситуаций (сочетаний стратегий игроков) и соответствующими им выигрышами или проигрышами (в общем случае *платежами*). По методу ведения игры различают дискретные игры, в которых игроки совершают поочередные ходы, и непрерывные, когда игроки действуют непрерывно. Последние также называют играми преследования (например, бой двух самолетов) или дифференциальными играми, так как поведение игроков описывается дифференциальными уравнениями. По количеству ходов выделяют игры с конечным и бесконечным числом шагов. Аналогичное деление производится по числу стратегий игроков. Так у противотанкового орудия имеется бесконечное число стратегий, так как огонь по нападающим танкам может быть открыт с любого расстояния, начиная с прицельной дальности стрельбы. По форме платы различают игры с нулевой суммой, когда выигрыши одних равны проигрышам других, и поэтому их также называют антагонистическими (цели полностью противоположные), и игры с ненулевой суммой, в которых выигрыши и проигрыши не совпадают. В зависимости от числа игроков говорят об играх

2, 3, ..., N лиц. Дискретную игру двух лиц с ненулевой суммой называют биматричной игрой, в ней каждой ситуации соответствует два платежа (по одному для каждого игрока).

Простейшей является дискретная игра двух лиц с нулевой суммой. Такая игра полностью представляется своей платежной матрицей, в которой отражены стратегии игроков и платежи, имеющие противоположный смысл для игроков:

Стратегии игрока A	Стратегии игрока B			
	B_1	B_2	...	B_n
A_1	u_{11}	u_{12}	...	u_{1n}
A_2	u_{21}	u_{22}	...	u_{2n}
...	u_{ij}	...
A_m	u_{m1}	u_{m2}	...	u_{mn}

Здесь u_{ij} - платеж, соответствующий ситуации A_i-B_j . В конкретной игре указывается, что понимается под u_{ij} . Например, будем дальше считать, что u_{ij} имеет смысл выигрыша игрока A и, следовательно, для игрока B это проигрыш.

Найти решение такой игры - значит определить оптимальное поведение каждого из игроков и соответствующий результат, называемый *ценой игры*. Метод решения основан на уже рассматриваемом принципе гарантированного результата, но в отличие от подхода, приведенного в разделе 1.4, он применяется обоими игроками. В данном случае этот принцип означает, что каждый из игроков при выборе своего поведения предполагает наилучшее поведение другого игрока, то есть рассчитывает на наихудший для себя вариант. Таким образом, игрок A будет определять свое поведение на основе максимина ($\max_i \min_j$), а игрок B - на основе

минимакса. Величина $v_B = \min_j \max_i u_{ij}$ называется верхней ценой игры, так как игрок B проиграет не более этой величины, если будет вести себя оптимально, следовательно, v_B - гарантированный проигрыш игрока B , а для игрока A это максимально возможный выигрыш. Нижняя цена игры $v_A = \max_i \min_j u_{ij}$ есть гарантированный выигрыш игрока A , то есть при оптимальном поведении он выиграет не меньше, в то время как для игрока B это минимально возможный проигрыш. Нетрудно доказать, что $v_B \geq v_A$, и поэтому в общем случае цена игры лежит в диапазоне $v_A \leq v \leq v_B$. Если $v_A = v_B = v$, то игра имеет решение в *области чистых стратегий*. Это значит, что каждый из игроков будет придерживаться только одной своей

стратегии, иначе говоря, одна (оптимальная) стратегия будет применяться с вероятностью единица, а все остальные - с вероятностью нуль. Такое решение соответствует *седловой* точке платежной матрицы. Когда u_{ij} - выигрыш игрока A , в седловой точке значение $u_{ij}^* = v$ и является наибольшим в столбце и наименьшим в строке. Таким образом, при наличии седловой точки решение всегда находится в области чистых стратегий, а оптимальные стратегии - это стратегии, на пересечении которых лежит седловая точка. Для примера приведем платежную матрицу и соответствующее решение, определение которого показано в последнем столбце и последней строке:

	B_1	B_2	B_3	min
A_1	2	3	7	2
A_2	10	2	1	1
A_3	5	4	8	4
A_4	9	1	12	1
max	10	4	12	

Находя максимум в последнем столбце, получаем нижнюю цену игры $v_n = 4$, а минимум из значений последней строки дает $v_b = 4$. В результате имеем равенство $v_n = v_b = v = 4$, а оптимальными стратегиями являются A_3 и B_2 . Нетрудно убедиться, что на их пересечении находится седловая точка платежной матрицы. При таком поведении игрок A гарантирует себе выигрыш не меньше 4, а игрок B проиграет не более 4, и каждому из игроков, как это видно непосредственно из матрицы, невыгодно отклоняться от найденных стратегий.

В тех случаях, когда $v_n \neq v_b$, решение игры находится в *области смешанных стратегий*. Это значит, что игроки будут применять более одной стратегии, то есть оптимальное поведение состоит в смешении нескольких стратегий в сочетании, определяемом вероятностями активных стратегий. Следовательно, в результате решения игры должно быть найдено распределение вероятностей на стратегиях каждого из игроков и цена игры.

Проиллюстрируем еще один случай:

	B_1	B_2	B_3	B_4	min
A_1	6	8	1	4	1
A_2	5	3	7	6	3
max	6	8	7	6	

Здесь $v_n = 3$, $v_b = 6$ и, следовательно, $3 \leq v \leq 6$, а решение лежит в области смешанных стратегий. Платежная матрица не имеет седловой точки. Если один из игроков имеет только две стратегии, решение можно найти графически. Для этого проводятся две оси ординат на расстоянии, которое принимается за единицу. На этих осях откладываются платежи игрока, имеющего две стратегии. В нашем примере стратегиям A_1 и A_2 , соответствуют оси ординат на которых откладываются выигрыши игрока A . Зафиксируем стратегию B_1 . Тогда игрок A выиграет 6 при выборе стратегии A_1 (в этом случае вероятность применения этой стратегии равна 1) и выиграет 5, когда будет применять только стратегию A_2 (теперь

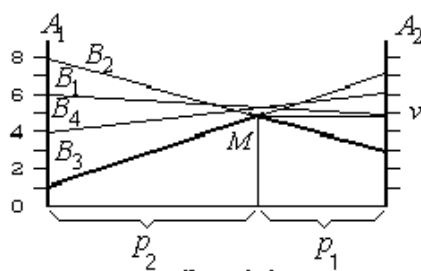


Рис. 1.2

вероятность применения этой стратегии равна 1). Если же применять стратегию A_1 с вероятностью x_1 , а стратегию A_2 с вероятностью $x_2 = 1 - x_1$, то ожидаемый выигрыш составит $6x_1 + 5(1 - x_1) = 5 + x_1$. Значит, выигрыш зависит от вероятности линейно, что и показано на рис. 1.2.

Аналогично строятся прямые, соответствующие выигрышам игрока A при фиксации других стратегий игрока B . Гарантированные выигрыши игрока A лежат на нижней грани, выделенной жирной линией. Очевидно, что игрок стремится к максимальному гарантированному выигрышу, который достигается в точке M . Следовательно, оптимальное решение для игрока A состоит в применении стратегии A_1 с вероятностью $p_1 = 4/11$ и стратегии A_2 с вероятностью $p_2 = 7/11$. При этом цена игры равна $49/11$. Обратим внимание на отсчет вероятностей: p_1 растет от 0 до 1 справа налево, а p_2 — наоборот, и значения вероятностей определяются как доли расстояния между осями координат. Теперь можно найти решение и для игрока B . Его активные стратегии (вероятности которых больше 0) определяются точкой M — это стратегии B_2 и B_3 . Вероятности применения этих стратегий находятся с помощью графика, аналогичного приведенному на рис. 1.2, но оси ординат должны соответствовать B_2 и B_3 . Они могут быть вычислены и аналитически. Отметим, что оптимальное решение в области смешанных стратегий может быть реализовано только при многократном (теоретически бесконечном) повторении игры, а цена игры есть соответствующее оптимальному поведению математическое ожидание результата (средний выигрыш или проигрыш на один розыгрыш).

Иногда можно сократить число стратегий одного из игроков или даже обоих за счет отбрасывания *доминируемых* стратегий, то есть стратегий, которые заведомо не будут активными в оптимальном решении. Для выявления таких стратегий проводят попарные сравнения стратегий одного

из игроков. Та стратегия, на которой платежи не лучше, чем на другой, для всех стратегий противника и хотя бы для одной хуже, является доминируемой и может быть отброшена. После сокращения числа стратегий у одного игрока возможно появление доминируемых стратегий у второго. Поэтому анализ можно повторять, поочередно меняя игрока, пока не удалятся все доминируемые стратегии. Если число стратегий у одного из игроков сократится до двух, игра решается графически. В общем случае решение в области смешанных стратегий находится методами линейного программирования.

Помимо рассмотренных применяются также модели коалиционных и кооперативных игр. Так, в игре n лиц ($n > 2$) с нулевой суммой в процессе игры могут образовываться объединения части игроков против остальных, если такая коалиция улучшает результаты всех объединившихся игроков, что обеспечивается побочными платежами со стороны инициатора объединения. В отличие от этого кооперативная игра может улучшать результаты всех игроков за счет предварительных договоренностей с заключением обязывающих соглашений (в играх с ненулевой суммой). Очевидно, что кооперация возможна и в игре двух лиц.

Одним из классов бескоалиционных игр являются позиционные игры. Они моделируют процессы последовательного принятия решений. Игра состоит в переходе из одного состояния игры в другое, происходящем при каждом выборе действия одним из игроков или случайным образом. Последовательные состояния и называют позициями. Выбор игрока может происходить при полной и неполной информации о его позиции. Примерами таких игр являются шахматы, шашки, домино и др. Позиционная игра может быть нормализована, то есть сведена к матричной игре.

Игровые модели находят применение в основном при исследовании военных операций и в экономике. Если вторая оперирующая сторона представляет собой некую среду с неизвестными вероятностями состояний, то такая ситуация также может моделироваться игрой, которая называется игрой против природы.

Модели типа **аукционного торга** применяются, когда степень неопределенности выше, чем предполагает модель игры. Так, например, в аукционном торге неизвестно даже число участников, нельзя составить принятую в игре платежную матрицу. Разработка и исследование моделей этого типа находятся в начальной стадии.

Задачи поиска. Процесс поиска связан с двумя видами ошибок. Из-за невозможности охвата всего множества объектов, среди которых могут быть искомые, возникает **ошибка выборки**. При исследовании выборки могут иметь место **ошибки наблюдения**, выражающиеся в том, что не обнаруживается (пропускается) искомый объект, входящий в выборку, или

другой объект принимается за искомый (ложное обнаружение). Любые ошибки приводят к потерям, упущению прибыли и т.п.

Для поиска необходимы ресурсы (в общем случае он требует затрат). При ограниченных ресурсах на поиск увеличение выборки уменьшает ошибку выборки, но одновременно возрастает ошибка наблюдения, а при уменьшении объема выборки - наоборот. Задача заключается в определении объема выборки и стратегии поиска внутри нее, при которых достигается наибольшая эффективность поиска. Характерным примером такой задачи является контроль качества деталей или изделий при серийном или массовом производстве. Основным ресурсом при этом выступают контролеры ОТК, число которых всегда ограничено. Чем больше выборка, тем меньше негодных деталей пройдет мимо контролера, но тем меньше времени у него на проверку каждой детали, попавшей в выборку, а значит, больше вероятность пропуска негодной детали или отбраковки годной. Бракованные детали, не обнаруженные контролером, могут проявить себя только при испытаниях готовой продукции или у потребителя, что влечет убытки для производителя. Задача контроля качества - организовать его так, чтобы убытки свелись к минимуму.

Задача поиска может ставиться шире, если количество ресурсов, выделяемых на поиск, может варьироваться в некоторых заданных пределах. Так как за ресурсы надо платить, то увеличение ресурсов не обязательно ведет к повышению эффективности. Поэтому задача состоит в определении оптимального уровня ресурсов и оптимального их использования. Так на стадии разработки проекта организации контроля качества может возникнуть необходимость определения числа контролеров, объема выборки и стратегии контроля внутри выборки, при которых будет обеспечиваться максимальная прибыль от выпускаемых изделий.

Некоторые области применения задач поиска:

- организация ревизий;
- бухгалтерские операции;
- обнаружение объектов (потерпевших аварию, заблудившихся, пожаров и т.п.);
- военная разведка;
- разведка полезных ископаемых;
- организация контроля качества;
- хранение и поиск информации;
- размещение рекламы в районе или городе.

Следует заметить, что модели поиска разработаны слабо. В основном они носят частный характер.

1.6. Резюме

Реальные проблемы далеко не всегда могут сводиться к одной из рассмотренных задач. Нередко в одной проблеме сплетается ряд задач, разделить которые не представляется возможным. Так, в машиностроении обработка деталей производится партиями, при этом определение объема партий как задача управления запасами связано через затраты с графиком запуска-выпуска деталей, а нахождение оптимального графика требует, чтобы было известно время обработки на всех операциях, но последние напрямую зависят от объема партии. Таким образом, здесь воедино связаны две типовые задачи: управления запасами и упорядочения.

Как правило, на практике возникают именно такие комплексные задачи. Но знание типичных задач облегчает поиск решения задач более сложных. Естественно, что приведенный перечень задач исследования операций не является исчерпывающим, да и никакой другой не может претендовать на абсолютную полноту, так как разнообразие задач не имеет границ.

Почти для каждого класса типичных задач можно указать наиболее эффективные методы решения. В то же время один метод или одна группа методов могут быть эффективны для ряда задач, а некоторые задачи, например, задачи поиска не имеют методов, характерных для всего данного класса. В связи с тем, что этап нахождения наилучшего решения является центральным в исследовании операций, последующие разделы пособия посвящены в основном рассмотрению методов оптимизации.

Глава 2. ПОСТАНОВКА ЗАДАЧИ ОПТИМИЗАЦИИ

По Лейбницу наш мир является наилучшим из всех миров, и поэтому его законы можно описать экстремальными принципами.

К. Зигель

В мире не происходит ничего, в чем бы не был виден смысл какого-нибудь максимума или минимума.

Л. Эйлер

Как было показано выше, центральным этапом исследования операций является отыскание наилучшего решения из всех возможных в рассматриваемой операции, причем наилучшего в заранее определенном смысле. В математической модели операции этот смысл описывается *целевой функцией* (критерием оптимальности) $f(\mathbf{X})$, а возможные решения - *областью допустимых решений* (допустимым множеством) D . Таким образом, в общем случае задача оптимизации ставится так: найти такой вектор $\mathbf{X}^* \in D$, который доставляет оптимум функции $f(\mathbf{X})$, то есть

$$f(\mathbf{X}) \rightarrow \underset{\mathbf{X} \in D}{\text{opt}} \quad (2.1)$$

и, следовательно, $\mathbf{X}^* = \arg \text{opt } f(\mathbf{X})$. Если решение \mathbf{X}^1 предпочтительнее решения \mathbf{X}^2 при $f(\mathbf{X}^1) > f(\mathbf{X}^2)$, то говорят о *задаче максимизации*. Если же \mathbf{X}^1 предпочтительнее \mathbf{X}^2 при $f(\mathbf{X}^1) < f(\mathbf{X}^2)$, то задача оптимизации представляет собой *задачу минимизации*.

В задачах исследования операций множество D задается конечным числом условий, в общем случае в виде равенств, неравенств и прямых ограничений на переменные. Тогда задача оптимизации принимает вид

$$f(\mathbf{X}) \rightarrow \text{opt}, \quad (2.2)$$

$$\varphi_i(\mathbf{X}) \geq 0, i = \overline{1, m_1}, \quad (2.3)$$

$$\psi_i(\mathbf{X}) = 0, i = \overline{m_1 + 1, m}, \quad (2.4)$$

$$\mathbf{X} \in P, P \subset R^n \quad (2.5)$$

и называется *задачей математического программирования*. Условия (2.3), (2.4) иногда называют функциональными, а также неявными ограничениями. Множество P обычно представляет собой параллелепипед, то есть

$$P = \{\mathbf{X} \in R^n \mid \alpha_j \leq x_j \leq \beta_j, j = \overline{1, n}\}, \quad (2.6)$$

при этом часто $\alpha_j=0$, а $\beta_j=+\infty$. Прямые ограничения на переменные называют также явными.

Другой большой класс задач оптимизации составляют задачи оптимального управления. В них цель описывается функционалом, а искомыми являются функции. Такие задачи возникают в основном при разработке и исследовании динамических или пространственно протяженных систем. Здесь они не рассматриваются.

Важнейшими характеристиками задачи оптимизации являются свойства функций, входящих в модель, и размерность задачи. Размерность задачи математического программирования определяется двумя величинами: числом искоемых переменных n (т.е. размерностью вектора \mathbf{X}) и числом условий-ограничений m . Существенное значение имеют также структура математической модели задачи, наличие случайных факторов, непрерывность или дискретность переменных.

Многообразие задач оптимизации бесконечно, и сегодня не существует универсального метода, с помощью которого можно было бы решить любую задачу. Однако уже создано много методов, каждый из которых имеет ограниченную область применения, причем эти области частично пересекаются, а в совокупности эти методы дают мощное средство решения широкого круга задач оптимизации и, в частности, задач математического программирования.

Укрупненно можно выделить следующие методы:

- классический анализ (включая метод множителей Лагранжа);
- линейное программирование;
- динамическое программирование;
- дискретное программирование;
- нелинейное программирование;
- стохастическое программирование.

Многие авторы не разделяют методы классического анализа и нелинейного программирования, но исторически эти методы значительно удалены, кроме того, в вузах методы поиска экстремума даются в курсе высшей математики, и они находят применение в других методах оптимизации как вспомогательное средство. Поэтому их целесообразно выделить в отдельную группу и рассмотреть в сжатом виде перед изучением современных методов.

Методы линейного программирования ориентированы на модели, в которых *все функции линейны*, а переменные непрерывны. Дискретность (в частности, целочисленность) переменных – особенность методов дискретного программирования. Динамическое программирование применимо к задачам, в которых процесс принятия решений можно представить как последовательность шагов. Существенное влияние случайных факторов приводит к стохастическим моделям, для нахождения

решения на которых применяются одноименные методы. Нелинейное программирование включает весьма широкий спектр методов решения задач с нелинейными функциями различной структуры.

Все названные выше группы методов за исключением первого имеют общее название **методов математического программирования**. Выбор метода для решения конкретной прикладной задачи исследования операций является далеко не тривиальным делом. При этом необходимо учитывать свойства задачи, о которых говорилось выше, и характеристики не только метода, но и его реализации на ЭВМ. Для нетипичных задач выбор метода требует определенного искусства и интуиции, источником которых являются прежде всего знания и опыт.

Глава 3. МЕТОДЫ КЛАССИЧЕСКОГО АНАЛИЗА

Когда величина является максимальной или минимальной, в этот момент она не течет ни вперед, ни назад.

И. Ньютон

Рассматриваемые здесь методы известны с XVIII века. Они применимы к гладким или кусочно-гладким функциям, т.е. предполагается дифференцируемость на всей области определения функции за исключением отдельных точек.

3.1. Основные положения теории экстремумов

Опуская строгие определения и доказательства, приведем в этом разделе основные понятия и результаты теории экстремумов, имеющие прикладное значение.

Любая задача оптимизации может быть представлена как задача максимизации или как задача минимизации: для перехода от одного вида задачи к другому достаточно изменить знак целевой функции. Так, задача минимизации функции $f=(x-5)^2$ эквивалентна задаче максимизации функции $f_1=-(x-5)^2$, и наоборот.

Различают локальный и глобальный оптимумы (иногда применяют термины-синонимы: относительный и абсолютный оптимумы). Значение целевой функции, наилучшее на всей области D , называют глобальным оптимумом, а соответствующую точку X^* – точкой глобального экстремума или глобальным решением. Если же $f(X^*)$ – наилучшее значение на окрестности точки X^* в области D , то говорят о локальном экстремуме.

Другой признак, по которому различают задачи и экстремумы, – наличие условий-ограничений, накладываемых на искомые переменные. Если модель задачи оптимизации не содержит условий (область D совпадает со всем евклидовым пространством), то имеем задачу на безусловный экстремум. В иных случаях, то есть при наличии ограничений, говорят о задаче на условный экстремум.

Очевидны следующие фундаментальные свойства задачи оптимизации. Для одной и той же целевой функции условный экстремум не может быть лучше безусловного. Сужение (расширение) множества D не может улучшить (ухудшить) глобальный экстремум.

Решение задачи оптимизации. Не для всех задач может быть найдено оптимальное решение. Простейший пример: нельзя найти минимум функции $f=10-x$ как при отсутствии ограничений, так и при условии $x \geq 5$. Для существования решения задачи оптимизации целевая функция и допустимое множество должны обладать определенными свойствами. В общем случае существование решения устанавливается следующей теоремой Вейерштрасса.

Теорема. Всякая функция, непрерывная на непустом замкнутом и ограниченном множестве, обладает наибольшим и наименьшим значениями, которые достигаются либо внутри множества, либо на его границе.

Условия теоремы **гарантируют** существование глобального оптимума, однако конкретная задача может иметь решение и на неограниченном или открытом множестве. На практике задачи исследования операций удовлетворяют условиям приведенной теоремы. Отметим, что теорема не требует непрерывности производных целевой функции.

Для определения глобального экстремума необходимо выявить и исследовать все точки, подозреваемые на экстремум. Эти точки называют также *экстремальными* или *критическими*. Из необходимых условий экстремума и теоремы Вейерштрасса следует, что возможны только три вида экстремальных точек:

- точки, в которых все частные производные первого порядка целевой функции одновременно обращаются в нуль, – **стационарные** точки;
- точки, в которых одна или большее число частных производных первого порядка перестают существовать (терпят разрыв);
- точки на границе допустимого множества.

Для случая функции одной переменной и $a \leq x \leq b$ на рис.3.1 показаны все три вида точек: A, B, C - стационарные точки, D, E, F - точки разрыва производной, a, b - граничные точки. При этом в точке D имеем глобальный максимум, а в точке a - глобальный минимум.

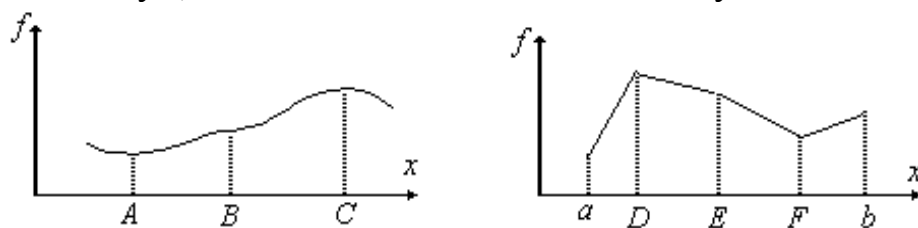


Рис.3.1

Очевидно, что не всякая "подозрительная" точка представляет интерес. Среди точек первых двух видов следует отбраковать те, в которых нет экстремума (например, точки B и E на рис.3.1) и которые не

соответствуют искомому типу экстремума. С этой целью используются **достаточные условия** экстремума.

Для функции одной переменной имеется несколько вариантов достаточных условий. Если $f(x^* \pm \varepsilon) > f(x^*)$, где ε - малая положительная величина, в точке x^* имеем локальный минимум. Сменив знак неравенства на обратный, получим условие максимума. Второй вариант заключается в проверке знака первой производной в точках $(x^* - \varepsilon)$ и $(x^* + \varepsilon)$: смена знака с минуса на плюс указывает на минимум, а с плюса на минус - на максимум. Наконец, для дважды дифференцируемой функции в точке x^* - минимум, если вторая производная положительна, и максимум, если она отрицательна. Для непрерывно дифференцируемой функции точке x^* соответствует экстремум, если в ней порядок первой ненулевой производной четный (в частности, второй).

При исследовании функции одной переменной могут оказаться полезными следующие простые свойства. У непрерывной функции максимумы и минимумы чередуются между собой. Поэтому, если между установленными точками максимума и минимума (или наоборот) имеется одна "подозрительная" точка, то в ней экстремума нет. Если функция строго возрастает на интервале $[a, b]$, т.е. $df/dx > 0$ на всем интервале, то $x_{max}^* = b$, а $x_{min}^* = a$. Это справедливо и при нестрогом возрастании, но тогда решение может быть не единственным. Если производная положительна в точке a , то $x_{max}^* \neq a$, если же она отрицательна, то $x_{min}^* \neq a$.

Определение экстремума облегчается, когда функция обладает свойством выпуклости или вогнутости. Это свойство устанавливается с помощью различных соотношений. Для одномерного случая справедливы следующие определения.

Функция $f(x)$ выпукла на интервале $[a, b]$, если для любых точек $x_1, x_2 \in [a, b]$ выполняется неравенство

$$f\left(\frac{x_1 + x_2}{2}\right) \leq \frac{f(x_1) + f(x_2)}{2}. \quad (3.1)$$

Функция $f(x)$ выпукла на интервале $[a, b]$, если для любых точек $x_1, x_2 \in [a, b]$ и произвольного числа $\theta \in [0, 1]$ выполняется неравенство

$$f(\theta x_1 + (1 - \theta)x_2) \leq \theta f(x_1) + (1 - \theta)f(x_2). \quad (3.2)$$

Дважды дифференцируемая на интервале $[a, b]$ функция $f(x)$ выпукла, если на этом интервале

$$f''(x) \geq 0. \quad (3.3)$$

Условия (3.1), (3.2) не требуют дифференцируемости функции. Их геометрический смысл иллюстрируется рис.3.2. Для выпуклой функции хорда, соединяющая две произвольные точки графика функции, лежит не ниже этой части графика. По условию (3.1) это устанавливается проверкой только в точке $(x_1 + x_2)/2$, а согласно (3.2) при изменении θ от 0 до 1 мы

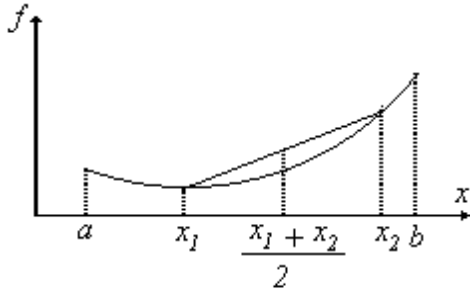


Рис.3.2

движемся от точки x_2 к точке x_1 одновременно по графику функции (левая часть неравенства) и по хорде (правая часть неравенства), то есть проверка осуществляется на всем отрезке $[x_1, x_2]$.

Изменение знаков неравенств в (3.1)-(3.3) на противоположные приводит к определению вогнутой функции. Если знаки неравенств заменить строгими,

исключить случай $x_1 = x_2$ и граничные значения θ , то получим определение строго выпуклой или строго вогнутой функции. Очевидно, что если $f(x)$ выпукла, то $-f(x)$ – вогнута, и наоборот.

В теории оптимизации наиболее употребительно второе определение выпуклости. Оно легко обобщается на n -мерные функции: функция $f(\mathbf{X})$ выпукла на выпуклом множестве D , если для любых двух точек $\mathbf{X}^{(1)}, \mathbf{X}^{(2)} \in D$ и произвольного числа $\theta \in [0, 1]$ выполняется неравенство

$$f(\theta \mathbf{X}^{(1)} + (1 - \theta) \mathbf{X}^{(2)}) \leq \theta f(\mathbf{X}^{(1)}) + (1 - \theta) f(\mathbf{X}^{(2)}), \quad (3.4)$$

где $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}$ – n -мерные векторы переменных.

Рассмотрим пример. Покажем, что функция $f(x) = x^2$ выпукла на интервале $[-1, 1]$. Для этой функции условие выпуклости (3.2) принимает вид

$$(\theta x_1 + (1 - \theta)x_2)^2 \leq \theta x_1^2 + (1 - \theta)x_2^2.$$

Справедливость данного неравенства для $\theta = 0$ и $\theta = 1$ очевидна. Теперь докажем, что оно выполняется и для $0 < \theta < 1$. Возведя в квадрат левую часть и сделав элементарные преобразования, получим

$$\theta^2 x_1^2 - \theta x_1^2 + (1 - \theta)^2 x_2^2 - (1 - \theta)x_2^2 + 2\theta(1 - \theta)x_1 x_2 \leq 0$$

или

$$\theta(1 - \theta)x_1^2 + \theta(1 - \theta)x_2^2 - 2\theta(1 - \theta)x_1 x_2 \geq 0.$$

Так как $\theta(1 - \theta) \neq 0$, то из этого неравенства следует

$$x_1^2 + x_2^2 - 2x_1 x_2 \geq 0 \quad \text{или} \quad (x_1 - x_2)^2 \geq 0,$$

что всегда справедливо, а при $x_1 \neq x_2$ имеем строгое неравенство. Таким образом, функция $f(x) = x^2$ является строго выпуклой.

Приведем некоторые полезные свойства выпуклых (вогнутых) функций на выпуклом множестве D .

1. Если функции f_1, f_2, \dots, f_m выпуклы на выпуклом множестве D и числа C_1, C_2, \dots, C_m не отрицательны, то функция $\sum_{i=1}^m C_i f_i(\mathbf{X})$ также выпукла на D .

2. Если функции f_1, f_2, \dots, f_m выпуклы на множестве D , то выпукла на D и функция $f(\mathbf{X}) = \max_i f_i(\mathbf{X})$.

3. Если функция $f(X)$ выпукла на множестве D , то для любого числа α множество $\{X \in D | f(X) \leq \alpha\}$ является выпуклым.

4. Функция $f(X)$, строго выпуклая (вогнутая) на D , имеет не более одного локального минимума (максимума). Любой локальный минимум (максимум) является глобальным.

5. Функция $f(X)$, строго выпуклая (вогнутая) на D , достигает наибольшего (наименьшего) значения на границе множества D .

Теперь остановимся на особенностях исследования функций многих переменных. Определение стационарных точек даже функции одной переменной может вызвать затруднения, если необходимое условие приводит к сложному нелинейному уравнению (например, трансцендентному). При n переменных необходимые условия дают систему n уравнений, в общем случае нелинейных. Решение системы нелинейных уравнений представляет собой самостоятельную сложную задачу, и здесь не обойтись без методов нелинейного программирования. Но последние можно применить непосредственно к функции, не используя необходимые условия, что нередко оказывается более эффективным.

Если стационарные точки найдены, то из них нужно выделить те, в которых функция имеет экстремум искомого типа. При этом недостаточно исследовать поведение функции по отдельным переменным. Так, например, для функции двух переменных малые отклонения от стационарной точки A как по x_1 , так и x_2 приводят к уменьшению значения функции (рис.3.3), что, казалось бы, указывает на наличие в этой точке максимума. Однако в точке A нет ни максимума, ни минимума. Действительно, если повернуть оси координат на 45° , то при отклонении по x_1' функция возрастает, а по x_2' - убывает. Такую точку называют **седловой** (аналог точки перегиба функции одной переменной). Точно так же в случае функции многих переменных стационарная точка является седловой, если функция ведет себя по-разному в окрестности этой точки. Таким образом, для установления экстремума необходимо исследовать всю окрестность стационарной точки, что можно сделать с помощью ряда Тейлора. В нем слагаемые первого порядка содержат первые производные, которые в стационарной точке равны нулю; слагаемые порядка выше второго стремятся к нулю быстрее слагаемых второго порядка. Следовательно, локальные свойства функции определяются производными второго порядка. Опуская выкладки, приведем *достаточные условия экстремума* в окончательном виде.

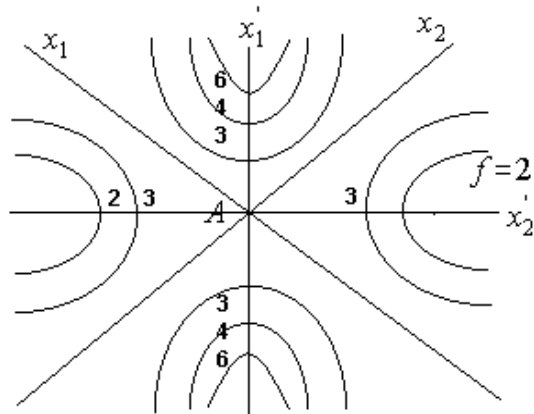


Рис.3.3

Составим квадратную матрицу n -го порядка из вторых частных производных (матрицу Гессе или гессиан):

$$\begin{bmatrix} f_{x_1x_1} & f_{x_1x_2} & \cdots & f_{x_1x_n} \\ f_{x_2x_1} & f_{x_2x_2} & \cdots & f_{x_2x_n} \\ \cdots & \cdots & \cdots & \cdots \\ f_{x_nx_1} & f_{x_nx_2} & \cdots & f_{x_nx_n} \end{bmatrix}$$

где $f_{x_ix_j} = f_{x_jx_i}$ - частные производные функции $f(X)$ по x_i и x_j в исследуемой стационарной точке. В стационарной точке имеем локальный минимум, если все диагональные определители матрицы Гессе положительны, то есть

$$\Delta_i > 0, \quad i=1, n \quad (3.5)$$

и локальный максимум, если

$$\Delta_i < 0, \quad i=1, 3, 5, \dots, \quad \Delta_i > 0, \quad i=2, 4, 6, \dots \quad (3.6)$$

В частности, для функции двух переменных условия минимума (3.5) имеют вид

$$\Delta_1 = f_{x_1x_1} > 0, \quad \Delta_2 = f_{x_1x_1}f_{x_2x_2} - f_{x_1x_2}^2 > 0, \quad (3.7)$$

а условия максимума (3.6)

$$\Delta_1 = f_{x_1x_1} < 0, \quad \Delta_2 = f_{x_1x_1}f_{x_2x_2} - f_{x_1x_2}^2 > 0. \quad (3.8)$$

Из приведенных выражений видно, что увеличение размерности функции ведет к непропорционально увеличивающемуся объему вычислений, необходимых для проверки достаточных условий. Еще большие трудности возникают при исследовании точек разрыва производных. Поэтому на практике часто отказываются от проверки достаточных условий, а точку глобального экстремума находят прямым сравнением значений функции во всех критических точках.

При решении задач на условный экстремум приходится выявлять также "подозрительные" точки третьего вида - точки на границе допустимой области. В задачах с одной переменной граница D - это точки на концах интервала, а в многомерном случае граница представляет собой линию или поверхность и ее исследование с целью выявления экстремальных точек становится весьма проблематичным.

Проиллюстрируем это на примере следующей задачи:

$$f(X) \rightarrow \min, \quad (3.9)$$

$$\varphi_1(X) \geq 0, \quad (3.10)$$

$$\varphi_2(X) \geq 0, \quad (3.11)$$

$$\varphi_3(X) \geq 0. \quad (3.12)$$

Как решать такую задачу? Сначала найдем все точки, в которых может быть экстремум. Для этого нам потребуется решить ряд задач меньшей размерности.

Задача 1: найти экстремальные точки (далее - э.т.) внутри допустимой области. С этой целью решаем задачу на безусловный экстремум: $f(X) \rightarrow \min$. Предположим, что необходимые условия экстремума дают систему уравнений, которая поддается решению, и мы находим все э.т. Из них оставим только те, которые удовлетворяют условиям (3.10)-(3.12). Можно также отсеять точки, в которых не минимум, но для этого надо проверять достаточные условия.

Задача 2: найти э.т. на границе допустимой области, порождаемой ограничением (3.10). Для этого нужно решить задачу на условный экстремум

$$f(X) \rightarrow \min, \quad \varphi_1(X)=0.$$

Здесь возможны два подхода: метод исключения и метод множителей Лагранжа. Применим первый из них (второй будет рассмотрен в следующем разделе). Из условия-равенства выразим одну переменную через остальные и подставим в $f(X)$. Таким образом, задачу на условный экстремум с n переменными сводим к задаче на безусловный экстремум с $n-1$ переменной. Снова предположим, что нам удастся разрешить систему необходимых условий для этой задачи и найти ее э.т. Оставляем только те из них, которые удовлетворяют неравенствам (3.11) и (3.12).

Задача 3 аналогична задаче 2, но рассматривается граница $\varphi_2(X)=0$ и оставляемые э.т. должны удовлетворять неравенствам (3.10) и (3.12). Такова же и *задача 4*, в которой исследуются точки на границе $\varphi_3(X)=0$.

Следующие три задачи заключаются в исследовании границ, образованных одновременно двумя ограничениями. Так, *задача 5* имеет вид

$$\begin{aligned} f(X) &\rightarrow \min, \\ \varphi_1(X) &= 0, \\ \varphi_2(X) &= 0. \end{aligned}$$

Предполагая, что два уравнения можно разрешить относительно двух переменных, и подставив выражения этих переменных в $f(X)$, сведем задачу 5 к задаче на безусловный экстремум размерности $n-2$. Если она разрешима, получим соответствующие э.т., которые следует "просеять" через условие (3.12). Оставшиеся точки добавляем к ранее выделенным.

Задачи 6 и 7 аналогичны задаче 5, но границы порождаются другими парами условий. Еще одна граница образуется пересечением границ всех трех условий. Для ее исследования решаем *задачу 8*:

$$\begin{aligned} f(X) &\rightarrow \min, \\ \varphi_1(X) &= 0, \\ \varphi_2(X) &= 0, \\ \varphi_3(X) &= 0. \end{aligned}$$

Если три представленных уравнения разрешимы относительно трех переменных, то задача 8 сводится к задаче на безусловный экстремум размерности $n-3$. Так как при этом использовались все условия исходной

задачи, то проверять найденные э.т. на принадлежность допустимой области не надо. Последняя задача 9 состоит в определении глобального минимума. Для этого вычисляются значения функции $f(X)$ во всех точках, выявленных в задачах 1-8, и из них выбирается наименьшее.

Рассмотренный пример высвечивает все проблемы, возникающие при решении задачи математического программирования методом классического анализа. На каждом этапе решения нам приходилось делать оговорки типа "если", "предполагая". Невыполнение какой-либо одной из них создает неразрешимую ситуацию. Но и при возможности реализации приведенной схемы трудоемкость решения задачи весьма велика.

Завершая раздел, рассмотрим в качестве примера задачу исследования операций, которая может быть решена вышеприведенным способом. Это простая детерминированная задача управления запасами в следующей постановке. За период времени T склад должен поставить потребителю R единиц продукции, причем поставка должна быть равномерной. Пополнение склада производится партиями без задержки с момента заказа. Допустим дефицит продукции на складе, но он вызывает потери. Известны: C_1 - затраты на хранение единицы продукции в единицу времени; C_2 - потери от дефицита единицы продукции в единицу времени; C_3 - затраты на пополнение склада одной партией (не зависят от объема партии).

Требуется определить, как часто нужно пополнять склад и в каком объеме.

Построим математическую модель задачи. Характер изменения запасов на складе и продукции у потребителя показан на рис.3.4.

Неизвестными на графике являются: q - объем партии; s - максимальный запас на складе; $t_{ц}$ - продолжительность одного цикла, t_1 - часть времени цикла, в течение которого запас есть; t_2 - часть времени

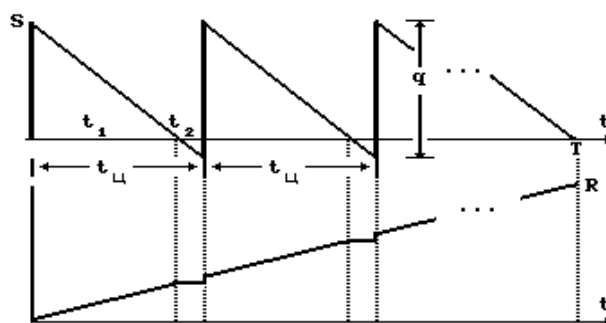


Рис. 3.4

цикла, когда имеется дефицит. За критерий оптимальности принимаются суммарные затраты, связанные с запасами.

Общие затраты определим как затраты за цикл, умноженные на число циклов m . Затраты за цикл следующие:

- затраты на хранение за время $t_1 - \frac{s}{2} C_1 t_1$,
- потери от дефицита в течение времени $t_2 - C_2 t_2 (q - s) / 2$,
- затраты на пополнение склада одной партией - C_3 .

Таким образом, суммарные затраты

$$Q = \left(\frac{s}{2} C_1 t_1 + \frac{q-s}{2} C_2 t_2 + C_3 \right) m. \quad (3.13)$$

Учитывая, что $m = R / q = T / t_{\text{ц}}$, получаем

$$t_{\text{ц}} = \frac{T}{R} \cdot q. \quad (3.14)$$

Из подобия треугольников (см. рис.3.4) и (3.14) имеем

$$t_1 = \frac{s}{q} t_{\text{ц}} = \frac{s}{R} T, \quad t_2 = \frac{q-s}{q} t_{\text{ц}} = \frac{q-s}{R} T.$$

Подставив выражения t_1 , t_2 и m в (3.13), приходим к окончательному виду целевой функции

$$Q(s, q) = \frac{s^2 C_1 T}{2q} + \frac{(q-s)^2 C_2 T}{2q} + \frac{C_3 R}{q} \rightarrow \min_{s, q}. \quad (3.15)$$

Формализация задачи завершается записью ограничений, которые вытекают из физического смысла переменных:

$$s \geq 0, \quad q \geq 0, \quad q - s \geq 0. \quad (3.16)$$

Таким образом, поставлена задача оптимизации с двумя переменными и тремя условиями. Малая размерность, дифференцируемость целевой функции и простота условий позволяют решать эту задачу методами классического анализа. Здесь мы не будем приводить решение, оставляя его для самостоятельной работы (см. раздел 3.3).

3.2. Метод неопределенных множителей Лагранжа

В классическом анализе этот метод ориентирован на решение задач на условный экстремум, когда условия имеют вид равенств:

$$f(X) \rightarrow \text{extr}, \quad \psi_i(X) = 0, \quad i = \overline{1, m}. \quad (3.17)$$

Очевидно, что задача (3.17) имеет смысл при $m < n$, когда система равенств имеет неединственное решение. В предыдущем разделе к таким задачам применялся метод подстановки (исключения зависимых переменных), который сводит исходную задачу к задаче на безусловный экстремум с меньшим числом переменных. На первый взгляд может показаться, что это очень удобный прием. Но на практике далеко не всегда можно разрешить m уравнений относительно m переменных, а в тех случаях, когда это возможно, получающиеся выражения часто оказываются слишком сложными.

Метод множителей Лагранжа тоже преобразует исходную задачу в задачу на безусловный экстремум, но, в противоположность предыдущему методу, за счет увеличения числа неизвестных.

Для лучшего понимания сути метода сначала покажем, как получаются необходимые условия условного экстремума в задаче с двумя переменными, а затем сделаем обобщение.

Итак, найдем экстремум функции $f(x_1, x_2)$ при условии

$$\psi(x_1, x_2) = 0. \quad (3.18)$$

Этим равенством задается допустимая область D , в которой одна из переменных становится зависимой. Пусть зависимой переменной будет x_2 . В стационарной точке производные первого порядка по независимым переменным равны нулю. В нашем случае имеется только одна независимая переменная - x_1 , поэтому в нуль обращается полная производная целевой функции по x_1 . Используя правила дифференцирования неявной функции, получаем:

$$\frac{df}{dx_1} = f_{x_1} + f_{x_2} \frac{dx_2}{dx_1} = 0, \quad (3.19)$$

где f_{x_1}, f_{x_2} - частные производные по x_1 и x_2 соответственно.

В области D функция ψ постоянна, следовательно, производная этой функции по независимой переменной равна нулю:

$$\frac{d\psi}{dx_1} = \psi_{x_1} + \psi_{x_2} \frac{dx_2}{dx_1} = 0, \quad (3.20)$$

В точке экстремума выполняются одновременно условия (3.19) и (3.20).

Исключая из них $\frac{dx_1}{dx_2}$, приходим к равенству

$$\frac{f_{x_1}}{\psi_{x_1}} = \frac{f_{x_2}}{\psi_{x_2}},$$

которое показывает, что в точке условного экстремума соблюдается равенство отношений частных производных целевой функции и функции условия по одноименным переменным. Обозначим это отношение постоянной - λ . Полученное равенство

$$\frac{f_{x_1}}{\psi_{x_1}} = \frac{f_{x_2}}{\psi_{x_2}} = -\lambda$$

можно представить в виде

$$\left. \begin{aligned} f_{x_1} + \lambda \psi_{x_1} &= 0, \\ f_{x_2} + \lambda \psi_{x_2} &= 0. \end{aligned} \right\} \quad (3.21)$$

Таким образом, в точке условного экстремума необходимо выполнение условий (3.21). Но они содержат три неизвестных при двух

равенствах. Добавляя к ним исходное условие задачи (3.18), получаем три уравнения с тремя неизвестными. К такому же результату мы придем, если будем решать задачу на безусловный экстремум расширенной целевой функции вида

$$F(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda \psi(x_1, x_2). \quad (3.22)$$

Действительно, взяв производные F по x_1 и x_2 и приравняв их нулю, получим равенства (3.21), а $dF/d\lambda=0$ дает (3.18). Новая функция F называется *функцией Лагранжа*, а неизвестная постоянная λ - *неопределенным множителем Лагранжа*.

Данная задача имеет простую геометрическую интерпретацию. На рис. 3.5 показаны линии равного уровня функции f и допустимая область, представляющая собой кривую, на которой $\psi(x_1, x_2)=0$. Безусловный экстремум находится в точке B , а условный - в точке A , в которой касательные к линии уровня и к кривой $\psi=0$ совпадают. Следовательно, в точке

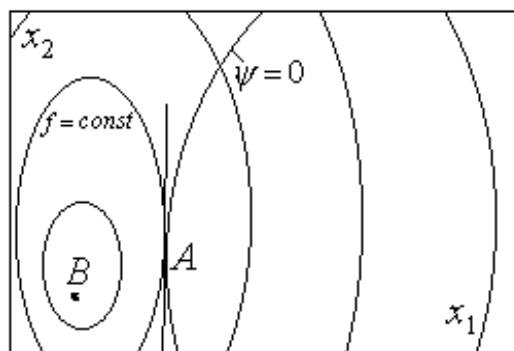


Рис.3.5

условного экстремума производные dx_1/dx_2 для функций f и ψ равны, что и было получено выше (формулы (3.19) и (3.20)).

Канторович Л.В. придает множителю λ экономический смысл. Если под $f(X)$ понимать доход, соответствующий плану выпуска продукции X , а под $\psi(X)$ - ограничение на издержки ресурса, то λ имеет смысл «цены» этого ресурса: доказывается, что производная оптимального значения функции f по ограничению ψ с точностью до знака равна λ . Иначе говоря, множитель Лагранжа характеризует влияние изменения ресурса на оптимальное значение критерия. В такой трактовке λ называют **маргинальной оценкой** ресурса (в теории двойственности λ является двойственной переменной).

Можно показать, что и в общем случае справедлив такой подход. Опуская доказательства, приведем окончательный результат. Для задачи (3.17) функция Лагранжа записывается в виде

$$F(\mathbf{X}, \Lambda) = f(\mathbf{X}) + \sum_{i=1}^m \lambda_i \psi_i(\mathbf{X}), \quad (3.23)$$

где Λ - вектор неопределенных множителей Лагранжа размерности m . Отсюда следует, что функция Лагранжа представляет собой сумму целевой функции и парных произведений множителей на левые части условий. Необходимые условия по X дают n уравнений:

$$\frac{\partial F}{\partial x_i} = 0, i = \overline{1, n}.$$

Добавляя к ним m равенств $\psi_i(\mathbf{X}) = 0$, получаем полную систему уравнений для отыскания $n+m$ неизвестных.

Несмотря на увеличение числа неизвестных этот подход приводит к более простой системе уравнений, чем при использовании метода исключения переменных, и поэтому более предпочтителен. К функции Лагранжа, естественно, приложимы также достаточные условия экстремума (3.5), (3.6).

Применение метода Лагранжа проиллюстрируем двумя примерами. Первый пример призван показать технику и преимущество метода. Пусть требуется найти максимум функции $f = x_1 + x_2 + x_1 x_2$ при условии

$$x_1^2 + x_2^2 = R^2. \quad (3.24)$$

Сначала применим метод исключения. Из (3.24) выразим x_2

$$x_2 = \sqrt{R^2 - x_1^2},$$

и, подставив в целевую функцию, получим

$$f = x_1 + (1 + x_1)\sqrt{R^2 - x_1^2}.$$

В результате пришли к задаче на безусловный максимум функции одной переменной. Взяв производную по x_1 и приравняв ее нулю, после элементарных преобразований имеем

$$\sqrt{R^2 - x_1^2} + R^2 - x_1 - 2x_1^2 = 0.$$

Способ нахождения x_1 из этого уравнения не очевиден. Даже в такой простой задаче применение метода исключения вызывает затруднения.

Теперь воспользуемся методом Лагранжа. Для нашей задачи функция Лагранжа записывается в виде

$$F(x_1, x_2, \lambda) = x_1 + x_2 + x_1 x_2 + \lambda(R^2 - x_1^2 - x_2^2).$$

Необходимые условия дают два уравнения:

$$\frac{\partial F}{\partial x_1} = 1 + x_2 - 2\lambda x_1 = 0,$$

$$\frac{\partial F}{\partial x_2} = 1 + x_1 - 2\lambda x_2 = 0,$$

которые при фиксированном λ представляют собой линейную систему с двумя неизвестными. В данном случае она решается особенно просто: из симметрии относительно переменных следует, что в стационарной точке они равны между собой. Поэтому сразу имеем

$$x_1^* = x_2^* = -\frac{1}{1 - 2\lambda}.$$

Подставив это выражение в (3.24), получаем

$$\frac{2}{(1-2\lambda)^2} = R^2 \text{ или } \frac{1}{(1-2\lambda)} = \pm \sqrt{2}.$$

Таким образом, условный экстремум может достигаться в точках

$$x_1^* = x_2^* = \pm \frac{R}{\sqrt{2}}.$$

Легко проверить, что положительные значения соответствуют максимуму при любом R , а отрицательные - минимуму только в определенном интервале R . (Убедиться в этом можно, записав достаточные условия экстремума функции F по переменным x_1 и x_2 .)

Во втором примере рассмотрим одну из типичных задач исследования операций и увидим, что даже в общем случае применение метода множителей Лагранжа оказывается полезным, приводя к интересным качественным результатам.

Задача распределения ресурсов. В системе имеется m видов ресурсов в количестве $V_k, k = \overline{1, m}$, и N предприятий, нуждающихся в этих ресурсах. Однако ресурсов недостаточно для удовлетворения каждого предприятия оптимальным образом. Прибыль предприятия зависит как от количества выделяемых ему ресурсов, так и от управления на уровне предприятия:

$$Z_i = z_i(v^i, u^i).$$

Здесь $v^i = (v_1^i, v_2^i, \dots, v_m^i)$ - вектор ресурсов, выделяемых i -му предприятию; $u^i = (u_1^i, u_2^i, \dots, u_n^i)$ - вектор индивидуальных управлений i -го предприятия. Необходимо так распределить имеющиеся ресурсы, чтобы суммарная прибыль предприятий была максимальной.

Модель задачи имеет вид:

$$Z = \sum_{i=1}^N z_i(v^i, u^i) \rightarrow \max_{v, u}, \quad (3.25)$$

$$\sum_{i=1}^N v_k^i = V_k, k = \overline{1, m}, \quad (3.26)$$

$$\forall u^i \in D^i. \quad (3.27)$$

Допустимое множество управлений D^i не задано в явном виде, и его свойства не оговариваются.

Реальные задачи такого типа из-за их большой сложности не удается решать «в лоб». Поэтому задачу разбивают, если это возможно, на ряд более простых задач. Такой прием называют **декомпозицией** задачи. В нашем случае декомпозиция возможна, так как в задаче (системе) явно просматриваются два уровня принятия решений: на верхнем уровне распределяются все ресурсы исходя из цели системы в целом, а на нижнем каждое предприятие стремится использовать выделяемые ему ресурсы наилучшим образом в собственных интересах (рис. 3.6). Следовательно,

при таком разбиении имеем одну задачу верхнего уровня и N задач нижнего уровня.

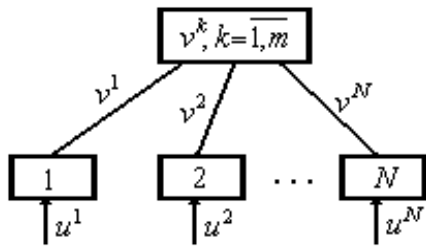


Рис. 3.6

Но задачи обоих уровней связаны между собой: для решения задач одного уровня надо знать решение другого. Выйти из замкнутого круга можно в тех случаях, когда удастся найти параметрическое решение задач одного из уровней. В рассматриваемой системе таким является нижний уровень.

Каждая из N задач нижнего уровня ставится так: найти зависимость максимальной прибыли предприятия от выделяемых ему ресурсов. Для этого нужно решить задачу

$$z_i(v_i, u_i) \rightarrow \max_{u_i \in D^i} \quad (3.28)$$

для всех возможных v^i . Метод решения задачи (3.28) здесь не рассматриваем, так как он может быть выбран только при явном задании модели. В результате решения этой задачи получим искомые функции $\bar{z}_i(v_i)$ и $u^i(v^i)$. Тогда задача (3.25)-(3.27) преобразуется в задачу верхнего уровня

$$Z = \sum_{i=1}^N \bar{z}_i(v^i) \rightarrow \max_{v^i}, \quad (3.29)$$

$$\sum_{i=1}^N v_k^i = V_k, k = \overline{1, m}. \quad (3.30)$$

Если функции \bar{z}_i дифференцируемы (что будем предполагать), то к этой задаче применим метод множителей Лагранжа. Расширенная функция (функция Лагранжа) задачи (3.29), (3.30) имеет вид

$$F(v, \Lambda) = \sum_{i=1}^N \bar{z}_i(v^i) + \sum_{k=1}^m \lambda_k \left(V_k - \sum_{i=1}^N v_k^i \right). \quad (3.31)$$

Необходимые условия экстремума функции F дают систему mN уравнений

$$\frac{\partial F}{\partial v_k^i} = \frac{\partial \bar{z}_i(v^i)}{\partial v_k^i} - \lambda_k = 0, \quad i = \overline{1, N}, \quad k = \overline{1, m},$$

которую перепишем в виде

$$\frac{\partial \bar{z}_i(v^i)}{\partial v_k^i} = \lambda_k, \quad i = \overline{1, N}, \quad k = \overline{1, m}. \quad (3.32)$$

Так как правая часть в (3.32) не зависит от i , то из этой системы следует цепочка равенств

$$\frac{\partial \bar{z}_i(v^i)}{\partial v_k^1} = \frac{\partial \bar{z}_2(v^2)}{\partial v_k^2} = \dots = \frac{\partial \bar{z}_N(v^N)}{\partial v_k^N}, \quad k = \overline{1, m}. \quad (3.33)$$

Из каждой цепочки составляется $N-1$ независимых равенств, то есть всего получаем $m(N-1)$ уравнений, содержащих mN искомым величин v_k^i . Недостающие m уравнений дают исходные условия (3.30). Если система (3.33) нелинейна, то нахождение стационарных точек становится весьма проблематичным. Однако независимо от разрешимости системы (3.33), (3.30) мы можем сделать важный вывод, дающий качественное представление о свойствах оптимального распределения. Действительно, из (3.33) следует, что при оптимальном распределении ресурсов производные критериев всех потребителей по одному и тому же ресурсу равны между собой. Чтобы лучше понять смысл этого свойства, вспомним, что производная есть предел отношения приращения функции к приращению аргумента. Следовательно, при оптимальном распределении ресурсов реакция (изменение критерия) всех потребителей ресурсов на малое изменение одного и того же ресурса *одинакова*. Этот вывод справедлив для любых систем распределения рассмотренного класса. В частности, при распределении одного ресурса полученное свойство позволяет решить задачу графическим способом (рис.3.7): решение соответствует точкам на графиках функций $\bar{z}_i(v^i)$, в которых касательные параллельны и $\sum v^i = V$.

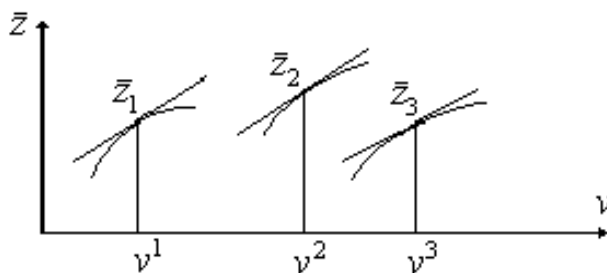


Рис. 3.7

Подводя итог рассмотрению методов классического анализа, отметим, что они могут применяться для решения задач оптимизации небольшой размерности и с простой структурой множества D . Задачи математического программирования, к которым сводится большинство задач исследования операций, требуют более мощных методов. Изучению этих методов посвящены главы 4-9.

Для закрепления пройденного материала в следующем разделе предлагается самостоятельно исследовать две задачи, одна из которых рассматривалась в разделе 3.1.

3.3. Задания для самостоятельной работы

Задание включает решение и анализ двух задач исследования операций на основе классических методов.

Задача управления запасами

Условия этой задачи были описаны в конце раздела 3.1. Там же была построена математическая модель задачи.

Критерий - суммарные затраты, связанные с управлением запасами,

$$Q(s, q) = \frac{s^2 C_1 T}{2q} + \frac{(q - s)^2 C_2 T}{2q} + \frac{C_3 R}{q} \rightarrow \min.$$

Условия: $n=R/q$; $t_{ц}=T/n=Tq/R$; $t_1=TS/R$, $t_2=(q-S)T/R$.

Ограничения: $S \geq 0$, $q \geq 0$, $S \leq q$.

Известными величинами в задаче являются R , T , C_1 , C_2 и C_3 .

Задание:

1. Найти в общем виде оптимальные q^* , $t_{ц}^*$, S^* , Q^* .
2. Получить численные значения всех величин (по п.1); доказать, что получен \min .
3. Построить зависимости Q^* от двух заданных параметров (отдельно для каждого).
4. Найти в общем виде и вычислить оптимальные величины для случая, когда дефицит недопустим (можно использовать переход $C_2 \rightarrow \infty$); доказать, что получен \min .
5. Для случая отсутствия дефицита построить на одном графике зависимости от q : затрат на хранение, затрат на выпуск партий и суммарных затрат.
6. Сделать анализ результатов (в т.ч. сравнить два случая) и соответствующие выводы.

Варианты заданий представлены в табл. 1.

Таблица 1

№ варианта	C_1 , руб./сут.ед.)	C_2 , руб./сут.ед.)	C_3 , руб./партия	R , ед.	T , сутки	Параметры
1	0,015	0,03	350	10000	150	C_2, R
2	0,02	0,05	180	8500	120	C_1, T
3	0,01	0,03	120	15000	200	C_1, C_3
4	0,03	0,07	200	20000	180	C_2, C_3
5	0,05	0,04	300	32000	365	R, T
6	0,012	0,08	250	18000	220	C_1, R
7	0,008	0,02	320	40000	200	C_2, T
8	0,013	0,08	140	25000	360	C_3, R
9	0,04	0,05	150	5000	90	C_3, T
10	0,1	0,25	100	12000	120	C_1, T
11	0,035	0,1	290	30000	250	C_1, C_2
12	0,12	0,3	310	18000	170	C_2, C_3
13	0,06	0,2	220	7500	300	C_1, R
14	0,025	0,07	370	28200	210	C_3, R
15	0,09	0,35	160	15600	350	C_2, R
16	0,06	0,21	380	55000	400	R, T
17	0,16	0,4	95	6700	280	C_1, C_3
18	0,07	0,8	130	11500	310	C_3, R
19	0,05	0,05	260	9500	270	C_3, T

№ варианта	C_1 , руб./сут.ед.)	C_2 , руб./сут.ед.)	C_3 , руб./партия	R , ед.	T , сутки	Параметры
20	0,2	0,5	400	35400	380	C_1, C_2
21	0,06	0,3	330	27200	290	C_3, R
22	0,045	0,1	210	18500	330	C_1, T
23	0,23	0,42	390	42000	365	C_2, R
24	0,12	0,18	315	29100	300	C_1, R
25	0,055	0,25	270	36700	380	C_2, T

Задача размещения

Условия. В кампусе с прямоугольной сетью дорог известно расположение пунктов A, B, C, D, E, F, M - источников и приемников информации. Для создания сети планируется построить два пункта для концентраторов: $K1$ и $K2$. Известно количество каналов между пунктами и концентраторами, а также между концентраторами. Все каналы

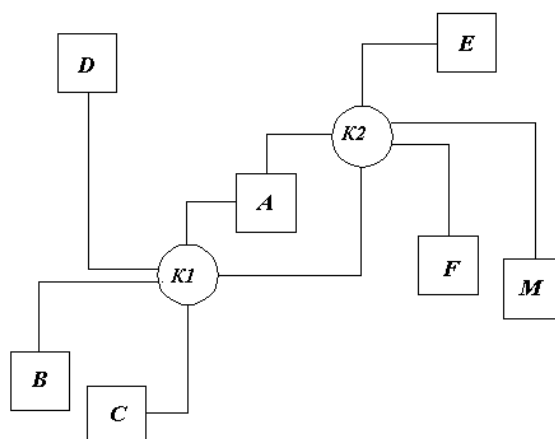


Рис. 3.8

однотипные. Схема расположения пунктов в общем случае показана на рис. 3.8.

Требуется найти оптимальное расположение концентраторов.

Задание:

1. Построить модель задачи.
2. Получить оптимальное решение (или решения).
3. Построить схему кампуса с пунктами концентраторов, связями между пунктами и указанием числа каналов.

Варианты исходных данных приведены в табл. 2 и 3. Прочерк означает отсутствие пункта (см. табл. 2) или каналов между соответствующими пунктами (см. табл. 3).

Указания. Построение модели следует начинать с содержательного определения критерия и введения искомых переменных. Чтобы избежать ошибки при математической записи критерия, нужно четко представлять себе физический смысл величины, принятой в качестве критерия. Кроме того, необходимо учитывать конфигурацию сети дорог. При отыскании

наилучшего решения следует использовать классический анализ. Для этого необходимо выявить все точки, подозреваемые на экстремум (возможно существование только трех типов экстремальных точек), затем путем непосредственного вычисления значения критерия в этих точках определить наилучшее решение. Для облегчения решения задачи следует провести анализ полученной модели на предмет возможной декомпозиции (разделения) исходной задачи на ряд задач меньшей размерности. При этом декомпозиция должна быть обоснованной. На схеме каждая связь между пунктами должна быть представлена отдельной линией с указанием числа каналов.

Таблица 2

№ вар-та	A		B		C		D		E		F		M	
	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
1	10	10	0	5	-	-	-	-	20	30	-	-	25	0
2	10	15	-	-	0	0	0	30	-	-	20	5	-	-
3	15	10	0	0	-	-	5	30	-	-	-	-	30	0
4	10	12	0	12	-	-	-	-	30	30	30	0	-	-
5	15	0	-	-	-	-	0	20	30	40	-	-	-	-
6	25	10	-	-	0	0	-	-	40	20	-	-	-	-
7	0	10	-	-	-	-	0	20	-	-	10	0	20	0
8	20	30	0	0	-	-	-	-	-	-	30	10	-	-
9	40	10	-	-	0	5	0	20	-	-	20	0	-	-
10	20	20	0	20	-	-	-	-	-	-	-	-	40	0
11	0	40	0	20	10	0	-	-	-	-	-	-	30	20
12	30	10	-	-	-	-	0	15	30	30	30	0	-	-
13	10	30	-	-	0	0	-	-	-	-	20	0	25	5
14	15	15	-	-	-	-	0	40	-	-	-	-	40	0
15	20	30	0	10	-	-	5	30	-	-	30	0	-	-
16	5	10	-	-	0	0	0	30	20	50	-	-	-	-
17	10	0	-	-	-	-	0	20	30	40	30	0	-	-
18	20	18	5	0	0	10	-	-	40	30	-	-	-	-
19	15	10	-	-	-	-	0	30	-	-	25	0	-	-
20	30	20	-	-	5	0	0	40	-	-	-	-	30	10
21	10	20	0	0	-	-	-	-	25	30	-	-	-	-
22	40	30	-	-	0	0	-	-	90	50	90	20	-	-
23	30	15	-	-	30	0	-	-	45	40	-	-	60	10
24	5	30	2	10	-	-	2	50	-	-	30	10	-	-
25	0	0	-	-	-	-	0	30	40	20	-	-	40	0
26	20	40	5	5	10	10	-	-	-	-	60	70	-	-
27	30	10	-	-	-	-	5	50	60	40	-	-	50	0
28	10	12	0	8	-	-	-	-	20	30	-	-	25	0

Таблица 3

№ вар-та	Количество рейсов между пунктами								
	<i>A- K1</i>	<i>A- K2</i>	<i>K1-K2</i>	<i>B- K1</i>	<i>C- K1</i>	<i>Д- K1</i>	<i>Е- K2</i>	<i>F-K2</i>	<i>М-K2</i>
1	8	15	10	20			25		12
2	15	10	15		20	10		30	
3	5	10	20	10		10			30
4	10	20	5	20			10	15	
5	15	10	10			25	35		
6	10	5	5		20		30		
7	20	10	10			30		15	10
8	20	20	5	30				20	
9	30	20	10		20	20		10	
10	5	5	20	15					20
11	10	20	10	20	5				25
12	30	10	10			30	5	5	
13	20	5	5		20			10	10
14	20	20	10			30			40
15	5	15	20	15		20		30	
16	10	10	10		20	5	20		
17	20	5	15			30	40	10	
18	10	10	5	20	15		30		
19	5	10	5			20		15	
20	30	5	10		20	25			30
21	12	10	5	20			25		
22	40	20	30		30		30	50	
23	10	10	5		20		25		30
24	5	5	30	20		15		25	
25	10	15	10			25	20		30
26	30	20	15	25	40			35	
27	20	10	50			30	25		22
28	10	20	12	20			25		15

Глава 4. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

4.1. Постановка задачи

Задача, модель которой содержит только линейные функции искомых переменных, называется **задачей линейного программирования** (ЛП).

В общем случае модель задачи ЛП имеет вид

$$L = C_1x_1 + C_2x_2 + \dots + C_jx_j + \dots + C_nx_n \rightarrow opt \quad (4.1)$$

при ограничениях:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2 \\ \dots \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \\ x_j \geq 0, \quad j = \overline{1, n_1} \quad (n_1 \leq n), \end{array} \right. \quad (4.2)$$
$$(4.3)$$

где L – критерий (целевая функция), называемый также **линейной формой**;
 n – количество переменных;

C_i – параметры (коэффициенты) критерия, не все $C_i = 0$;

(4.2) – функциональные условия (ограничения);

a_{ij} – параметры условий (могут быть любыми действительными числами, но одновременно все не могут равняться нулю при $i = \text{const}$). Во многих случаях они имеют смысл удельных величин (расхода или затрат на единицу переменной, содержания в единице переменной и т. п.).

b_i – параметры (свободные члены), отражающие возможности по ресурсам, допустимые или требуемые значения показателей и т.п. (могут быть любыми действительными числами).

На часть или все переменные накладывается условие неотрицательности (4.3).

Задача состоит в определении таких значений переменных, удовлетворяющих условиям (4.2) и (4.3), которые доставляют в зависимости от контекста максимум или минимум линейной форме.

4.2. Примеры задач линейного программирования

На первый взгляд линейные модели могут показаться малоприменимыми для описания реальных задач принятия решений. Однако это неверно. Большое число задач из области экономики, бизнеса,

финансов, планирования, организации, управления и др. представимы как задачи ЛП. В качестве примеров рассмотрим несколько типичных ситуаций.

4.2.1. Задача составления рациона или как экономно питаться

При правильном питании человек должен потреблять в день определенное количество питательных веществ и витаминов. Содержание их в продуктах известно. Известна также цена продуктов. Задача состоит в том, чтобы составить самый дешевый набор продуктов (рацион), обеспечивающий заданное качество питания.

Обозначим:

x_i – количество i -го продукта в рационе,

a_{ij} – содержание i -го питательного вещества в единице j -го продукта,

b_i – потребность в ингредиенте (норма).

Чтобы не загромождать модель большим числом данных, ограничимся тремя ингредиентами, но для общности возьмем разные варианты задания нормы. Все исходные данные удобно представить в таблице (табл. 4.1).

Таблица 4.1

Ингредиенты	Продукты				Норма	
	1	2	3	4	min	max
Жиры	a_{11}	a_{12}	a_{13}	a_{14}		b_1
Белки	a_{21}	a_{22}	a_{23}	a_{24}	b_2	
Витамины	a_{31}	a_{32}	a_{33}	a_{34}	b_3	b_3^{max}
Цена	C_1	C_2	C_3	C_4		

Как следует из условий задачи, в качестве критерия, по которому определяется лучший вариант рациона, нужно взять суммарные затраты на продукты. Ограничения должны отражать требования по количеству ингредиентов в рационе: левая часть – фактическое поступление с продуктами, правая – норму, а отношение между ними определяется видом задания нормы (не меньше или не больше). Неотрицательность переменных очевидна из их физического смысла. В итоге приходим к следующей модели:

$$L = C_1x_1 + C_2x_2 + C_3x_3 + C_4x_4 \rightarrow \min$$

при условиях:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 \leq b_1; \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 \geq b_2; \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 \geq b_3; \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 \leq b_3^{\text{max}}, \\ \forall x_i \geq 0. \end{cases}$$

Построение модели завершено.

4.2.2. Задача оптимального планирования

Предприятие выпускает несколько видов продукции.

Известны нормы расхода сырья на единицу продукции, прибыль предприятия от реализации единицы продукции, количество сырья, которым располагает предприятие в планируемом периоде. Необходимо составить план производства, обеспечивающий максимальную прибыль.

Представим данные в виде таблицы (табл.4.2):

Таблица 4.2

Виды сырья	Продукция		Количество сырья
	1	2	
1	2	3	19
2	2	1	13
3	0	3	12
4	3	0	17
Прибыль	7	5	

Во втором и третьем столбцах указаны нормы расхода сырья на единицу продукции.

Очевидно, что критерием качества плана является прибыль. Введем переменные x_1 и x_2 – количество продукции 1-го и 2-го вида. Тогда модель задачи будет иметь вид

$$L = 7x_1 + 5x_2 \rightarrow \max$$

$$\begin{cases} 2x_1 + 3x_2 \leq 19; \\ 2x_1 + x_2 \leq 13; \\ 3x_2 \leq 12; \\ 3x_1 \leq 17, \\ x_1, x_2 \geq 0. \end{cases}$$

Необходимо найти такие x_1 и x_2 , которые удовлетворяют всем условиям и доставляют максимум критерию L .

4.2.3. Сбалансированная транспортная задача

Транспортная задача - это модель ситуации, в которой требуется найти оптимальный план перевозки некоторого груза из конечного числа пунктов поставки (отправления) с заданными объемами производства в конечное число пунктов потребления (назначения) с требуемыми объемами потребностей при известных затратах на перевозку единицы груза между каждой парой пунктов поставки и потребления. Предполагается, что удельные затраты не зависят от количества перевозимого груза. Здесь под оптимальным понимается план, минимизирующий суммарные затраты на перевозки.

В качестве примера рассмотрим задачу с двумя пунктами отправления и тремя пунктами назначения, схема которой показана на рис.4.1. Здесь a_1 и a_2 – количество груза, которым располагают пункты отправления, b_1, b_2, b_3 – потребности в грузе пунктов назначения.

Задача является сбалансированной, если суммарная потребность равна суммарной возможности. В нашем примере это значит, что

$$\sum_{i=1}^2 a_i = \sum_{j=1}^3 b_j.$$

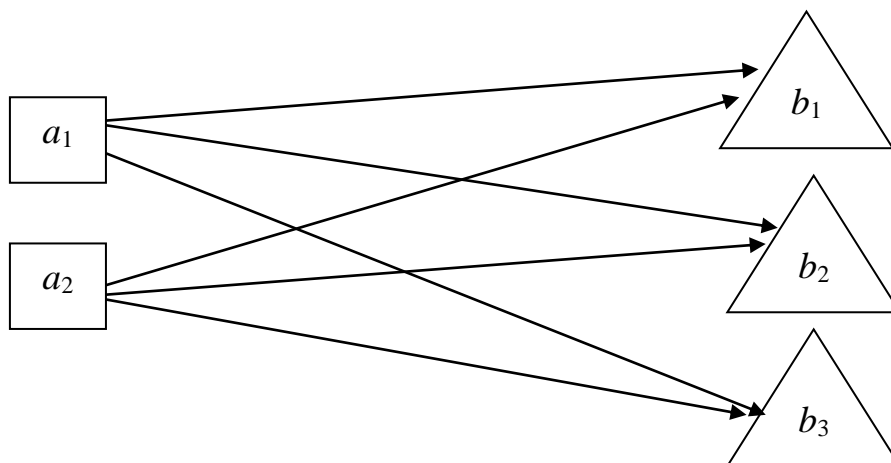


Рис. 4.1.

Если ввести обозначения:

x_{ij} – количество груза, перевозимого из i -го пункта отправления в j -й пункт назначения;

C_{ij} – затраты на перевозку единицы груза из i -го пункта отправления в j -й пункт назначения,

то исходные данные вместе с переменными можно представить в одной таблице (табл. 4.3):

Таблица 4.3

Пункты	В ₁	В ₂	В ₃	Количество груза
А ₁	$\begin{matrix} C_{11} \\ X_{11} \end{matrix}$	$\begin{matrix} C_{12} \\ X_{12} \end{matrix}$	$\begin{matrix} C_{13} \\ X_{13} \end{matrix}$	a_1
А ₂	$\begin{matrix} C_{21} \\ X_{21} \end{matrix}$	$\begin{matrix} C_{22} \\ X_{22} \end{matrix}$	$\begin{matrix} C_{23} \\ X_{23} \end{matrix}$	a_2
Потребность в грузе	b_1	b_2	b_3	$\sum_{i=1}^2 a_i = \sum_{j=1}^3 b_j.$

Так как необходимо минимизировать суммарные затраты по перевозке, то целевая функция запишется в виде

$$L = C_{11}x_{11} + C_{12}x_{12} + C_{23}x_{23} \rightarrow \min.$$

Каждый пункт назначения должен получить требуемое количество груза. Отсюда следуют равенства, соответствующие этим пунктам

$$B_1: x_{11} + x_{21} = b_1;$$

$$B_2: x_{12} + x_{22} = b_2;$$

$$B_3: x_{13} + x_{23} = b_3.$$

Поскольку задача сбалансированная, весь груз из пунктов отправления должен быть вывезен. Это требование отражается в модели двумя равенствами:

$$A_1: x_{11} + x_{12} + x_{13} = a_1;$$

$$A_2: x_{21} + x_{22} + x_{23} = a_2.$$

Наконец, физический смысл переменных накладывает на них ограничение неотрицательности

$$\forall x_{ij} \geq 0.$$

В результате мы получили модель транспортной задачи, содержащей только линейные функции. Очевидно, что характер модели не изменится при увеличении числа пунктов.

4.2.4. Общая распределительная задача

Транспортная задача является частным случаем линейной задачи распределения. В общем случае задача распределения ставится следующим образом. Имеется список работ или операций, каждая из которых требует затрат некоторых видов ограниченных ресурсов. Необходимо так распределить ресурсы по работам, чтобы получить наилучший результат в смысле принятого критерия.

Как и в предыдущих примерах, исходные данные задачи можно представить в виде таблицы, где C_{ij} – доход, прибыль или затраты при выделении единицы ресурса i -го вида на j -ю работу (табл. 4.4):

Таблица 4.4

Виды ресурсов	Работы (операции)				Количество ресурсов
	1	2	...	n	
1	C_{11}	C_{12}	...	C_{1n}	a_1
2	C_{21}	C_{22}	...	C_{2n}	a_2
...
m	C_{m1}	C_{m2}	...	C_{mn}	a_m
Потребность	b_1	b_2	...	b_n	

Если C_{ij} – прибыль или выигрыш, то имеем задачу максимизации, если же C_{ij} – затраты или убытки, то задачу минимизации.

К распределительным задачам относятся и *транспортные задачи*. Пример такой задачи приведен выше. Нетрудно видеть, что в *транспортной задаче* единицы измерения ресурсов и потребностей

одинаковы. Если же единицы измерения не совпадают, то имеет место *общий случай задачи распределения*.

Другим важным частным случаем задачи распределения (и транспортной задачи) является *задача о назначениях* или *задача выбора*. Она отличается от общего случая тем, что по условиям задачи с каждым ресурсом можно оперировать только как с единым целым, он либо направляется на работу, либо нет. Работе также требуется только один вид ресурса, причем целиком. Иначе говоря, потребности и ресурсы неделимы. Следовательно, в задаче о назначениях всегда $a_i=1$, $b_j=1$ и x_{ij} – булевы, $\forall ij$.

Как и транспортные, общие распределительные задачи могут быть **сбалансированные** и **несбалансированные** в зависимости от соотношения между суммарными потребностями и ресурсами.

Рассмотрим пример общей распределительной задачи.

Пусть с железнодорожной станции необходимо отправить n видов грузов в количестве b_j , $j=1, 2, \dots, n$. Станция может использовать m видов вагонов в количестве a_i , $i=1, 2, \dots, m$. Для каждого вида вагона известна норма загрузки i -го вагона j -м грузом – B_{ij} . Известны затраты на погрузку i -го вагона j -м грузом – C_{ij} . Необходимо организовать отправку груза наилучшим образом, т.е. с минимальными затратами на погрузку.

Как видно, здесь ресурсы – вагоны разных типов (единицы измерения – штуки), а потребность – груз (тонны или m^3). Единицы измерения разные, следовательно, это задача общего вида.

Введем переменные x_{ij} – количество вагонов i -го типа, отводимых под j -й груз. В качестве критерия берем суммарные затраты на погрузку всего груза.

Тогда приходим к модели задачи, которая включает:

критерий

$$L = \sum_{i=1}^m \sum_{j=1}^n C_{ij} x_{ij} \rightarrow \min,$$

ограничения-равенства, отражающие необходимость отправить все виды грузов в полном объеме,

$$\sum_{i=1}^m B_{ij} x_{ij} = b_j, \quad j = \overline{1, n},$$

ограничения-неравенства, учитывающие ограниченное число вагонов каждого вида,

$$\sum_{j=1}^n x_{ij} \leq a_i, \quad i = \overline{1, m}$$

и условия неотрицательности переменных

$$x_{ij} \geq 0, \quad \forall i, j.$$

Размерность задачи определяется значениями n и m .

Принципиальное отличие данной модели от транспортной содержится в первой группе ограничений – это наличие коэффициентов B_{ij} . Если бы все B_{ij} равнялись единице, мы имели бы модель несбалансированной транспортной задачи (и тогда размерности x_{ij} , b_j и a_i совпадают).

4.2.5. Задача планирования работы оборудования

Фирма получила заказ на производство своей продукции. Оборудование, которым располагает фирма, является взаимозаменяемым. Известны нормы времени работы оборудования по каждому виду продукции. Требуется найти наилучший вариант выполнения заказа,

Очевидно, что за критерий следует принять время выполнения заказа, и чем оно меньше, тем лучше. Введем переменные x_{ij} – количество продукции i -го вида, произведенного на j -м оборудовании.

Ограничившись тремя видами продукции и двумя видами оборудования, представим исходные данные и все переменные в таблице (табл. 4.5).

Таблица 4.5

Продукция	Оборудование		Заказ
	1	2	
1	4 x_{11}	10 x_{12}	70
2	7 x_{21}	5 x_{22}	90
3	3 x_{31}	6 x_{32}	45

Цифры в правом верхнем углу клеток второго и третьего столбцов показывают норму времени на производство единицы продукции.

Составим модель задачи. Требование обязательного выполнения заказа порождает 3 ограничения-равенства:

$$\begin{aligned} x_{11} + x_{12} &= 70, \\ x_{21} + x_{22} &= 90, \\ x_{31} + x_{32} &= 45. \end{aligned} \quad (*)$$

Для записи критерия подсчитаем время работы первого оборудования

$$4x_{11} + 7x_{21} + 3x_{31}$$

и время работы второго оборудования

$$10x_{12} + 5x_{22} + 6x_{32}.$$

Время выполнения заказа T будет определяться максимальным из них. Таким образом, целевая функция задачи принимает вид

$$T = \max \left(\begin{matrix} 4x_{11} + 7x_{21} + 3x_{31}, \\ 10x_{12} + 5x_{22} + 6x_{32} \end{matrix} \right) \rightarrow \min.$$

Очевидно, что она и, следовательно, модель в целом не являются линейными. Однако такая модель легко преобразуется к линейной. Действительно, величина T как максимальная из двух времен должна быть не меньше каждого из них, то есть

$$\begin{aligned} 4x_{11} + 7x_{21} + 3x_{31} &\leq T, \\ 10x_{12} + 5x_{22} + 6x_{32} &\leq T. \end{aligned} \quad (**)$$

Рассматривая T как переменную, запишем критерий следующим образом:

$$L = T \rightarrow \min.$$

Этот линейный критерий совместно с условиями (**) эквивалентен исходной нелинейной целевой функции.

В итоге мы пришли к линейной модели, которая включает критерий L , ограничения (*), (**) и условия неотрицательности

$$x_{ij} \geq 0, \quad \forall i, j.$$

4.2.6. Игра двух лиц с нулевой суммой как задача линейного программирования

Рассмотрим платежную матрицу игры двух лиц, не имеющую седловых точек,

	B_1	B_2	...	B_n
A_1	U_{11}	U_{12}	...	U_{1n}
A_2	U_{21}	U_{22}	...	U_{2n}
...
A_m	U_{m1}	U_{m2}	...	U_{mn}

где платежи U_{ij} имеют смысл выигрышей игрока A .

Как известно, такая игра имеет решение в области смешанных стратегий. Пусть $X = (x_1, x_2, \dots, x_m)$ – распределение вероятностей на стратегиях игрока A . Тогда согласно принципу гарантированного результата этот игрок будет выбирать такое поведение (распределение X^*), которое максимизирует наименьший ожидаемый выигрыш

$$\max_X \min \left(\sum_i U_{i1} x_i, \sum_i U_{i2} x_i, \dots, \sum_i U_{in} x_i \right).$$

Обозначим через v минимальный ожидаемый выигрыш

$$v = \min \left(\sum_i U_{i1} x_i, \sum_i U_{i2} x_i, \dots, \sum_i U_{in} x_i \right).$$

Отсюда очевидно, что v не больше каждого ожидаемого выигрыша, и так как цель – максимальный выигрыш, то приходим к следующей эквивалентной задаче

$$L = v \rightarrow \max$$

при ограничениях

$$\left. \begin{aligned} \sum_i U_{i1} x_i &\geq v \\ \sum_i U_{i2} x_i &\geq v \\ &\dots \\ \sum_i U_{in} x_i &\geq v \end{aligned} \right\}$$

$$\sum_{i=1}^m x_i = 1,$$

$$\forall x_i \geq 0.$$

Это обычная линейная задача, оптимальное значение критерия которой $L^* = v^*$ есть цена игры. Ее решение определяет оптимальное поведение игрока A .

Для игрока B линейная модель строится аналогично, но тот же критерий минимизируется, так как v уже имеет смысл максимального проигрыша, а ограничения на вероятности y_j соответствуют строкам платежной матрицы и записываются как неравенства “меньше или равно”.

4.2.7. Резюме

Рассмотренные модели задач объединяет одно свойство: все функции, входящие в модель (целевая и ограничения), являются линейными относительно искомых переменных. Нетрудно видеть, что задача описывается линейной моделью, если справедливы 3 свойства:

- аддитивности (сложение составляющих затрат, прибыли, времени и т.д.);
- пропорциональности (прибыль, расход пропорциональны количеству продукции, услуг);
- непрерывности переменных.

Несмотря на сходство в основном, приведенные модели отличаются по виду экстремума (\max или \min) и/или по виду ограничений (равенства, неравенства, знаки неравенств). Возможны также задачи, в которых часть переменных не имеет ограничений по знаку (например, температура в градусах Цельсия, величина дебаланса, отклонение от заданного значения и т.п.). Чтобы абстрагироваться от этих несущественных отличий при

использовании методов решения линейных задач, модели приводят к некоторому единому виду. Основными являются две формы представления задач ЛП, которые рассматриваются ниже.

4.3. Формы записи задач линейного программирования и способы приведения к ним

4.3.1. Каноническая форма задач ЛП

Задача ЛП представлена в канонической форме, если в ее модели *все* функциональные условия имеют вид равенств и *все* переменные ограничены по знаку. Направление цели не имеет существенного значения, но для однозначности канонического представления будем иметь в виду максимизацию критерия. Тогда модель задачи ЛП в канонической форме записывается следующим образом

$$L = \sum_{j=1}^{\hat{n}} C_j x_j \rightarrow \max$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1\hat{n}}x_{\hat{n}} = b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2\hat{n}}x_{\hat{n}} = b_2; \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{m\hat{n}}x_{\hat{n}} = b_m; \end{cases}$$

$$\forall x_j \geq 0.$$

Если использовать векторно-матричные представления, то получим

$$L = \mathbf{C}^T \mathbf{X} \rightarrow \max$$

$$\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \dots + \mathbf{A}_{\hat{n}} x_{\hat{n}} = \mathbf{B} \quad \text{или} \quad \mathbf{A} \mathbf{X} = \mathbf{B}$$

$$\mathbf{X} \geq \mathbf{0},$$

где верхний индекс **T** означает транспонирование; $\mathbf{C} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{\hat{n}} \end{bmatrix}$ – вектор

коэффициентов целевой функции; $\mathbf{A}_j = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix}$ – векторы условий, $j = \overline{1, \hat{n}}$;

$\mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$ – вектор ограничений или вектор свободных членов, иногда

используют сокращение ССЧ (столбец свободных членов);

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1\bar{n}} \\ a_{21} & a_{22} & \dots & a_{2\bar{n}} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{m\bar{n}} \end{bmatrix} - \text{матрица условий; } \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{\bar{n}} \end{bmatrix} - \text{вектор переменных;}$$

\bar{n} – число переменных в канонической форме, оно не меньше числа переменных в исходной модели n .

Любую задачу ЛП можно привести к каноническому виду. Возможны 3 случая несоответствия исходной модели каноническому представлению. В каждом из них простое преобразование позволяет получить требуемый вид.

1. Если в исходной постановке критерий минимизируется, то изменив знак критерия на обратный, приходим к задаче максимизации, т.е. если $L = \sum_{j=1}^{\bar{n}} C_j x_j \rightarrow \min$, то $L' = -\sum_{j=1}^{\bar{n}} C_j x_j \rightarrow \max$.

2. В исходной модели есть неравенства. При этом способ преобразования зависит от знака неравенства. В случае неравенства $\sum_{j=1}^n a_{ij} x_j \leq b_i \geq 0$ очевидно, что разность правой и левой части будет неотрицательной и неизвестной величиной, которую можно принять за новую переменную:

$$b_i - \sum_{j=1}^n a_{ij} x_j = x_{n+i} \geq 0.$$

Отсюда получаем следующее равенство:

$$\sum_{j=1}^n a_{ij} x_j + x_{n+i} = b_i.$$

Таким образом, чтобы привести рассмотренное неравенство к равенству, нужно к левой части неравенства прибавить новую переменную.

Аналогично поступаем с неравенством $\sum_{j=1}^n a_{ij} x_j \geq b_i \geq 0$. Но теперь новая переменная обозначает разность левой и правой части

$$x_{n+i} = \sum_{j=1}^n a_{ij} x_j - b_i \geq 0$$

и равенство записывается в виде

$$\sum_{j=1}^n a_{ij} x_j - x_{n+i} = b_i,$$

то есть чтобы неравенство типа “больше или равно” привести к равенству, следует из левой части вычесть новую переменную. В отличие от исходных переменных такие вновь вводимые переменные будем называть *дополнительными*. Нетрудно видеть, что они по определению являются

неотрицательными, что соответствует каноническому представлению модели.

3. Некоторые переменные исходной модели не имеют ограничения на знак. Исключение таких переменных производится следующим способом.

Пусть x_k – переменная, которая может иметь любой знак. Введем две неотрицательные переменные $x'_k \geq 0$, $x''_k \geq 0$, и во всей модели заменяем x_k их разностью:

$$x_k = x'_k - x''_k.$$

Таким образом, последние два случая преобразования к каноническому виду приводят к увеличению числа переменных, и поэтому всегда $\hat{n} \geq n$.

Пример 4.1. Исходная модель:

$$\begin{aligned} L &= 2x_1 - x_2 - 5x_3 \rightarrow \min \\ 3x_1 + 7x_3 &\geq 4 \\ -6x_2 + x_3 &= 10 \\ x_1 + x_2 &\leq 12 \\ x_1 &\geq 0, x_3 \geq 0. \end{aligned}$$

Каноническая модель:

$$\begin{aligned} L' &= -2x_1 + x'_2 - x''_2 + 5x_3 \rightarrow \max \\ 3x_1 + 7x_3 - x_4 &= 4 \\ -6x'_2 + 6x''_2 + x_3 &= 10 \\ x_1 + x'_2 - x''_2 + x_5 &= 12 \\ x_1, x'_2, x''_2, x_3, x_4, x_5 &\geq 0. \end{aligned}$$

4.3.2. Стандартная форма задачи ЛП

Точные методы решения задач ЛП ориентированы на каноническую форму записи модели. Однако в геометрических представлениях удобнее стандартная форма. Мы будем понимать под стандартной модель, в которой *все* функциональные ограничения имеют вид неравенств и *все* переменные неотрицательные. Как и выше, тип экстремума не имеет существенного значения.

Таким образом, стандартная форма модели имеет вид

$$\begin{aligned} L &= \sum_{j=1}^{\bar{n}} C_j x_j \rightarrow \max \\ \begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1\bar{n}}x_{\bar{n}} \geq / \leq b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2\bar{n}}x_{\bar{n}} \geq / \leq b_2; \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{m\bar{n}}x_{\bar{n}} \geq / \leq b_m; \end{cases} \\ \forall x_j &\geq 0. \end{aligned}$$

Та же модель в векторно-матричных обозначениях:

$$L = \mathbf{C}^T \mathbf{X} \rightarrow \max$$

$$\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \dots + \mathbf{A}_n x_n \geq / \leq \mathbf{B} \quad \text{или} \quad \mathbf{A} \mathbf{X} \geq / \leq \mathbf{B}$$

$$\mathbf{X} \geq \mathbf{0}.$$

Здесь символ / означает “или”. Число переменных \tilde{n} при отсутствии неограниченных по знаку переменных не больше n . Соответственно матрица \mathbf{A} и вектор \mathbf{X} имеют меньшие размеры, чем в канонической модели.

Поясним преобразование равенств в неравенства. Пусть в исходной модели имеется q равенств. Решив эту систему уравнений относительно первых q переменных, получим

$$x_k = b'_k - \sum_{j=q+1}^n a'_{kj} x_j, \quad k = \overline{1, q}.$$

Используя эти равенства, исключаем x_k из целевой функции и ограничений, уменьшая тем самым количество переменных на q . Однако число ограничений не изменяется, так как для сохранения неотрицательности исключенных переменных должны выполняться неравенства

$$\sum_{j=q+1}^n a'_{kj} x_j \leq b'_k, \quad k = \overline{1, q}.$$

Таким образом, все ограничения задачи будут записаны в виде неравенств.

4.4. Упрощение модели

В ряде случаев анализ модели задачи позволяет сократить ее размерность как по числу ограничений, так и по числу переменных. С этой целью проводятся эквивалентные преобразования условий и/или линейной формы, называемые иногда нормированием модели.

Горизонтальное нормирование заключается в таком преобразовании условий модели, которое обеспечивает равенство всех свободных членов (правых частей).

Пусть ограничения задачи имеют вид

$$\sum_j a_{ij} x_j \leq b_i, \quad \forall b_i > 0.$$

Каждое неравенство разделим на свое b_i и умножим на одно и тоже некоторое положительное число b . Тогда получим

$$\sum_j a'_{ij} x_j \leq b, \quad \forall i,$$

где

$$a'_{ij} = a_{ij} \frac{b}{b_i},$$

а в качестве b можно взять любое натуральное число, в частности, полученное как целая часть среднего:

$$b = \left\lfloor \frac{1}{m} \sum_{i=1}^m b_i \right\rfloor.$$

Так как правые части всех неравенств одинаковы, то, если найдется 2 ограничения, для которых выполняются отношения

$$a'_{kj} \leq a'_{sj}, \quad \forall j,$$

ограничение k можно отбросить как лишнее.

Вертикальное нормирование касается линейной формы. Умножив и разделив коэффициенты критерия на константу C , получим

$$L = \sum_j Cx'_j,$$

где

$$x'_j = \frac{C_j}{C} x_j,$$

а величину C можно взять порядка среднего значения из C_j .

Если эту же операцию выполнить с ограничениями, подвергнутыми горизонтальному нормированию, то придем к неравенствам

$$\sum_j a''_{ij} x'_j \leq b, \quad \forall i,$$

где

$$a''_{ij} = a'_{ij} \frac{C}{C_j}.$$

В преобразованной таким способом модели сравним коэффициенты условий при переменных. Если найдется такая пара переменных, для которых выполняются неравенства

$$a''_{ir} \geq a''_{iq}, \quad \forall i,$$

то переменную x_r можно исключить, так как она заведомо не войдет в оптимальное решение.

4.5. Основные понятия ЛП. Свойства задач ЛП

Множество $D = \{x \in \mathbb{R}^n \mid AX \leq B, X \geq 0\}$ называется *допустимым множеством задач ЛП*. Иначе говоря, это множество всех решений, удовлетворяющих всем ограничениям задачи. Поэтому форма записи модели не влияет на D .

Любое решение $X \in D$ называют *допустимым решением* задачи ЛП.

Допустимое решение X^* является оптимальным для задачи максимизации, если выполняется неравенство

$$L(\mathbf{X}^*) \geq L(\mathbf{X}), \forall \mathbf{X} \in \mathbf{D}.$$

Чтобы определить свойства допустимого множества задачи ЛП, обратимся к терминам теории множеств.

Множество называется *замкнутым*, если оно содержит и свою границу, в противном случае оно *открытое*. Множество может быть *ограниченным*, если на нем все переменные ограничены снизу и сверху, и *неограниченным*, если хотя бы одна переменная на нем не ограничена. Непрерывное множество *выпукло*, если вместе с любыми двумя точками оно содержит и весь соединяющий их отрезок, иначе множество будет *невыпуклым*.

Пусть условия задачи записаны в стандартной форме

$$\sum a_{ij}x_j \leq b_i, \quad i = \overline{1, m}.$$

Так как в условия входят только непрерывные переменные, а отношения являются нестрогими, то порождаемое ими множество непрерывно и замкнуто. Возьмем одно неравенство. При двух переменных граница области, где выполняется это неравенство, представляет собой прямую, а множество значений переменных, удовлетворяющих строгому неравенству, расположено с одной стороны границы. Таким образом, неравенству соответствует полуплоскость, которая является выпуклым множеством. При трех переменных границей будет плоскость, а соответствующее множество – выпуклым полупространством. При большем числе переменных неравенство порождает многомерное выпуклое полупространство с границей – гиперплоскостью.

Из теории множеств известно, что пересечение выпуклых множеств выпукло (если оно не пустое). В задаче ЛП число неравенств, а, значит, число полупространств, конечно. Их пересечение и дает допустимое множество \mathbf{D} . В связи с этим дадим 2 определения.

Пересечение конечного числа выпуклых полупространств, если оно не пустое, называется **выпуклым многогранным множеством**.

Ограниченное выпуклое многогранное множество называется **выпуклым многогранником**.

Характерными примерами выпуклых многогранников являются различные пирамиды и призмы.

Таким образом, допустимое множество задачи ЛП может быть или выпуклым многогранным множеством, или выпуклым многогранником, или пустым. Других вариантов быть не может.

Критерий $L = \sum C_j X_j$ – линейная функция и поэтому удовлетворяет условиям как выпуклости, так и вогнутости одновременно.

Из теории экстремумов известно, что максимум вогнутой функции или минимум выпуклой функции на выпуклом множестве может достигаться только на границе.

Аналогичный вывод следует из рассмотрения критических точек целевой функции. Так как производная

$$\frac{\partial L}{\partial X_j} = C_j = \text{Const},$$

то точек, в которых она равна нулю или терпит разрыв, нет. Поэтому, если оптимум существует, то он может быть только на границе. Это важнейшее свойство задач ЛП.

Теперь введем понятие разрешимой задачи. Задача ЛП называется **разрешимой**, если она имеет хотя бы одно оптимальное решение, и **неразрешимой** в противном случае. Следует обратить внимание на то, что в этом определении не фигурируют допустимые решения, так как их наличие недостаточно для разрешимости задачи, что видно из нижеприводимых утверждений.

При решении задач ЛП возможны только три случая:

- 1) Условия задачи противоречивы (несовместны), допустимое множество пустое и, следовательно, задача неразрешима.
- 2) Условия задачи совместны, но допустимое множество неограниченно. Тогда возможны два исхода:
 - а) если критерий неограничен на этом множестве, то задача неразрешима;
 - б) если критерий ограничен, то задача разрешима.
- 3) Условия непротиворечивы и множество является выпуклым многогранником. В этом случае задача всегда разрешима.

Из приведенных утверждений, справедливость которых будет показана в следующем разделе, заключаем, что причиной неразрешимости задачи ЛП может быть либо неограниченность критерия, либо противоречивость ограничений.

Если модель задачи представлена в каноническом виде, то можно воспользоваться теоремой Кронекера-Капелли для системы линейных уравнений. Согласно теореме система совместна, если ранги основной и расширенной матриц равны, и несовместна в противном случае. Однако из-за трудоемкости вычисления ранга этот способ не используется, а неразрешимость задачи, если она имеет место, обнаруживается в процессе движения к оптимуму или при анализе результатов.

4.6. Геометрия задач ЛП

Геометрические представления позволяют лучше уяснить рассмотренные свойства задач ЛП.

Размерность пространства задачи k равна разности числа переменных и числа ограничений-равенств. Поэтому для модели в стандартной форме $k = \bar{n}$, а для канонической модели $k = \bar{n} - m$. Как отмечалось раньше,

значение k не зависит от формы записи модели и является фиксированным для конкретной задачи.

Очевидно, что геометрические построения возможны для $k \leq 3$. Сначала мы рассмотрим задачи на плоскости ($k=2$), а затем сделаем обобщение на многомерное пространство.

В качестве первого примера решим графически задачу оптимального планирования из разд. 4.2.2:

Исходная модель	Каноническая модель
$L=7x_1+5x_2 \rightarrow \max,$	$L=7x_1+5x_2 \rightarrow \max,$
1) $2x_1+3x_2 \leq 19,$	$2x_1+3x_2+x_3=19,$
2) $2x_1+x_2 \leq 13,$	$2x_1+x_2+x_4=13,$
3) $3x_2 \leq 12,$	$3x_2+x_5=12,$
4) $3x_1 \leq 17,$	$3x_1+x_6=17,$
5) $x_1 \geq 0,$	$\forall x_j \geq 0.$
6) $x_2 \geq 0.$	

Чтобы решить задачу, надо сначала построить допустимое множество, а затем по целевой функции найти на нем точку или множество с максимальным значением критерия.

Допустимое множество задачи находится как пересечение допустимых множеств, построенных по отдельным условиям задачи. В нашем примере таких множеств 6.

Построение множества начинается с определения его границы. Возьмем условия неотрицательности переменных:

$x_1 \geq 0$, следовательно, уравнение границы $x_1=0$ (ось x_2) и допустимое множество – правая полуплоскость (включая границу); $x_2 \geq 0$, уравнение границы $x_2=0$ (ось x_1), допустимое множество – верхняя полуплоскость. Таким образом, при неотрицательности всех переменных допустимое множество задачи ЛП всегда лежит в первом квадранте.

Теперь перейдем к построению допустимых множеств по функциональным условиям. Граница по 1-му условию $2x_1+3x_2=19$ (из канонической модели имеем альтернативное уравнение $x_3=0$) представляет собой прямую, для построения которой достаточно нанести любые 2 точки. Например, положив $x_1=0$, из уравнения получим $x_2=19/3$ (точка на оси x_2); вторую точку возьмем на оси x_1 : $x_2=0$ и тогда $x_1=19/2$. Чтобы определить, с какой стороны прямой выполняется условие, достаточно проверить одну точку вне границы. В качестве такой точки проще всего брать начало координат, в котором левая часть условия всегда равна нулю.. Наше условие 1) выполняется в этой точке, следовательно, допустимое множество – юго-западная полуплоскость. В противном случае получили бы северо-восточную полуплоскость.

Аналогично записываем уравнения границ и проводим соответствующие прямые по условию 2): из исходной модели $2x_1+x_2=13$ и канонической модели $x_4=0$; по условию 3): $3x_2=12$ и $x_5=0$; по условию 4): $3x_1=17$ и $x_6=0$. Нетрудно убедиться, что каждое из этих условий выполняется с той стороны своей границы, с которой находится начало координат.

Находя пересечение шести построенных допустимых множеств, получаем **выпуклый** шестиугольник (рис. 4.2) – частный случай выпуклого многогранника.

Из бесконечного множества допустимых решений, принадлежащих этому многоугольнику, необходимо найти то, которое дает максимум критерия. Для этого нам нужно знать “поведение” критерия на допустимом множестве. С этой

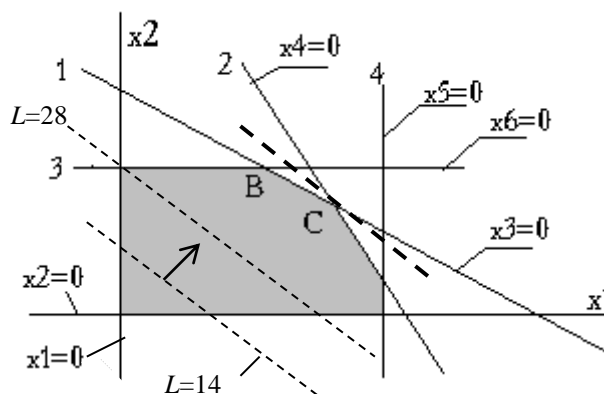


Рис. 4.2

целью построим линии уровня критерия $L=Const.$. Так как критерий линейный, то линии уровня – это множество параллельных прямых с постоянным градиентом (направлением возрастания значений критерия). Поэтому для определения этого направления достаточно иметь 2 линии уровня. Конкретные значения L можно брать любые, но так, чтобы соответствующие прямые оказались в области уже выполненных построений.

Полагая $L=14$, получаем уравнение линии уровня $7x_1+5x_2=14$. Аналогично для $L=28$ имеем уравнение линии уровня $7x_1+5x_2=28$. На рис. 4.2 они представлены тонкими пунктирными линиями. Теперь ясно, что критерий увеличивается в направлении, показанном стрелкой. Чтобы найти оптимальное решение, не нужно строить новые линии уровня. Достаточно мысленно перемещать прямую, параллельную линиям уровня, в направлении стрелки до тех пор пока она не выйдет на границу допустимого множества задачи. В нашем примере такое предельное положение соответствует прохождению линии уровня через точку C . Следовательно, в точке C критерий достигает максимального значения, а соответствующее решение является оптимальным. Координаты точки C – оптимальные значения переменных, можно снять с графика или для большей точности найти как решение системы 2-х уравнений границ, образующих эту точку:

$$\begin{aligned} 2x_1^* + 3x_2^* &= 19, \\ 2x_1^* + x_2^* &= 13. \end{aligned}$$

Отсюда имеем: $x_1^*=5$, $x_2^*=3$. Значения остальных переменных вычисляются однозначно из канонической модели после подстановки в ее уравнения x_1^* и x_2^* . В результате получаем оптимальное решение задачи: $x_1^*=5$, $x_2^*=3$, $x_3^*=x_4^*=0$, $x_5^*=3$, $x_6^*=2$ и, следовательно, $L^*=50$.

Таким образом, чтобы добиться максимальной прибыли в 50 единиц, необходимо производить 5 изделий первого типа и 3 изделия второго типа. При этом будут полностью использованы ресурсы 1-го и 2-го вида ($x_3^*=x_4^*=0$), а по ресурсам 3-го и 4-го вида образуются остатки в количестве 3 и 2 соответственно ($x_5^*=3$, $x_6^*=2$). Задача имеет **единственное оптимальное** решение, так как линия уровня $L=50$ соприкасается с допустимым множеством только в одной точке. ▲

Очевидно, что при данном допустимом множестве задача всегда будет иметь решение независимо от коэффициентов целевой функции, потому что в любом случае будет существовать предельное положение линии уровня критерия. Однако оптимальное решение может быть и не единственным.

Пусть в нашем примере при тех же ограничениях изменились коэффициенты критерия:

$$L_1=4x_1+6x_2 \rightarrow \max.$$

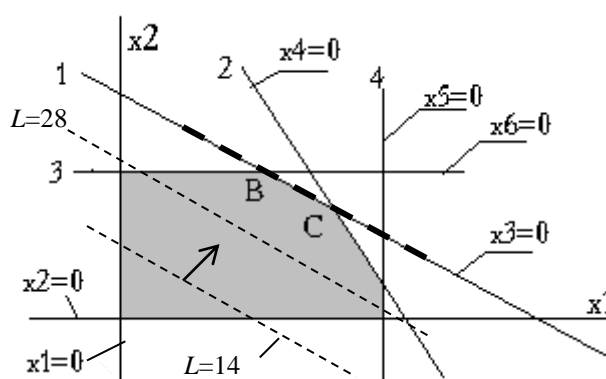


Рис. 4.3

Так как условия не изменились, то допустимое множество остается прежним. Если повторить построение линий уровня, то они будут параллельны стороне ВС (коэффициенты при переменных в критерии и 1-м ограничении пропорциональны). Поэтому в предельном положении линия уровня критерия совпадает с границей по 1-му ограничению

(рис. 4.3). В таком случае получим множество (бесконечное) оптимальных решений, лежащее на стороне ВС, каждому из которых соответствует одно и то же максимальное значение критерия L_1 .

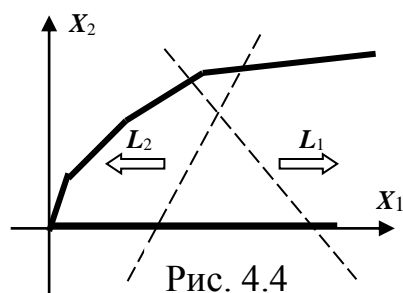


Рис. 4.4

Приведенные построения подтверждают высказанное ранее утверждение о том, что на ограниченном допустимом множестве (выпуклом многограннике) задача всегда разрешима.

Теперь выясним, что будет, если допустимое множество неограниченно. Не конкретизируя модель, рассмотрим

поведение двух целевых функций на одном неограниченном множестве (рис. 4.4). Направление улучшения критериев показано стрелками. Здесь мы видим разное поведение критериев, приводящее к различным результатам.

а) Значение критерия L_1 улучшается в направлении неограниченности множества и, следовательно, критерий неограничен на допустимом множестве, то есть он может улучшаться до бесконечности – допустимые решения есть, а оптимального нет.

б) Улучшение критерия L_2 ограничено на допустимом множестве, следовательно, решение существует (в этом примере оно единственное, но, очевидно, возможны и множественные решения на неограниченном множестве).

Рассмотрим другой случай (рис. 4.5). Пересечение допустимых множеств, порожденных тремя функциональными ограничениями задачи и условиями неотрицательности переменных, оказывается пустым. Следовательно, допустимых решений нет и задача неразрешима.

Результаты геометрического анализа задачи ЛП в двумерном пространстве легко обобщаются на многомерное пространство. Вместо прямых границами множеств будут гиперплоскости, а непустым допустимым множеством задачи – выпуклый многогранник или выпуклое многогранное множество. Для нахождения решения мысленно перемещаем гиперплоскость уровня критерия в направлении его улучшения пока имеют общие точки с допустимым множеством. Если достигается предельное положение, то задача разрешима. При этом гиперплоскость может касаться только одной вершины множества – задача имеет единственное решение, ребра или грани допустимого множества – существует множество решений соответственно на ребре или грани, которые полностью определяются своими крайними точками (вершинами). При отсутствии предельного положения критерий неограниченно улучшается и задача неразрешима.

Таким образом, если задача ЛП разрешима, то оптимальное решение обязательно достигается в вершине допустимого множества. Поэтому оптимальное решение следует искать не на всей границе, а только в вершинах допустимого множества.

4.7. Выделение вершин допустимого множества

Из последнего вывода возникает вопрос: как отличить вершины допустимого множества от других решений задачи? Для поиска ответа снова обратимся к геометрическим представлениям.

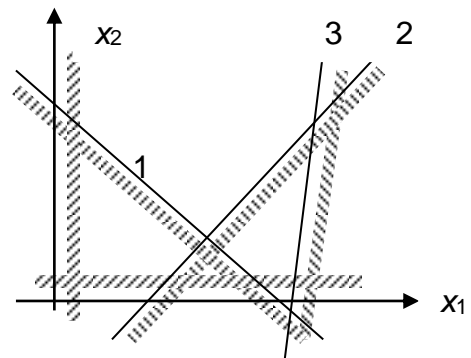


Рис. 4.5

Пусть каноническая модель содержит \bar{n} переменных и m линейно-независимых равенств. Тогда размерность пространства переменных $k = \bar{n} - m$. Как показано выше (рис. 4.2), на каждой границе допустимого множества одна из переменных равна нулю. В k -мерном пространстве вершина образуется пересечением k гиперплоскостей. Поэтому в ней k переменных заведомо равны нулю и только m переменных могут быть ненулевыми, т.к. $\bar{n} - k = \bar{n} - \bar{n} + m = m$. Если из вершины сместиться в любом направлении, то число ненулевых переменных увеличивается. Так при сдвиге из вершины С (рис. 4.2) по ребру в сторону вершины В к ненулевым переменным добавляются x_4 . В точках, не лежащих на границах условий, все переменные не равны нулю. Из линейной алгебры известно, что решение системы уравнений с рангом m содержит m базисных переменных и $\bar{n} - m$ свободных (небазисных). Если все свободные переменные равны нулю, то решение называется **базисным**. Следовательно, каждой вершине множества D соответствует некоторое базисное решение системы равенств.

На самом деле в вершине могут пересекаться более k гиперплоскостей (на плоскости – больше двух прямых; в трехмерном пространстве, например, вершина не в основании пирамиды образуется пересечением плоскостей, число которых может быть больше 3-х – по числу вершин многоугольника в ее основании). Тогда в нуль обращается более k переменных. Такие базисные решения (вершины) называют **вырожденными**, и задачи, имеющие хотя бы одно вырожденное решение, также называют вырожденными. Число “лишних” плоскостей (v) определяет степень вырожденности. Поэтому в общем случае в базисном решении число ненулевых переменных равно $m - v$ и можно определить **базисное решение как решение, в котором число ненулевых переменных не больше m** . В любом другом решении таких переменных больше m .

Однако не каждое базисное решение соответствует вершине допустимого множества, так как пересечение k или более гиперплоскостей имеет место и вне этого множества. Это наглядно видно и на рис.4.2, где $k = 2$. Учитывая, что на каждой границе одна переменная равна нулю, с одной стороны границы эта переменная будет положительной, а с другой отрицательной. В допустимом множестве **все** переменные неотрицательны. Таким образом, мы легко отделяем базисные решения, соответствующие вершинам множества D , от базисных решений, им не соответствующих. Базисное решение с неотрицательными переменными будем называть допустимым базисным решением или **опорным планом (решением)**.

Вывод: оптимальное решение задачи ЛП следует искать среди опорных решений, геометрически – в вершинах (крайних точках) допустимого множества.

Так как число вершин всегда конечно, то, казалось бы, задача может быть легко решена путем исследования всех вершин. Оценим такую возможность.

Известно, что число базисных решений системы линейных уравнений

$$C_n^r = \frac{n!}{r!(n-r)!}.$$

с n переменными и рангом r определяется сочетанием

Из них примерно 40% опорных. Возьмем небольшую задачу ЛП: 10 неравенств с 10 переменными. В канонической форме будем иметь систему уравнений с 20 переменными и рангом $r=10$. Получаем 184756 базисных решений и, значит, порядка 70 тысяч вершин (опорных решений). Столько раз нужно вычислить координаты вершин и значение критерия, а затем сравнить. Если учесть, что реальные задачи содержат сотни и тысячи ограничений и переменных, то становится ясным, что такой способ решения неприемлем даже при самых мощных компьютерах. В таких случаях приходится обращаться к соответствующим методам решения линейных задач.

4.8. Методы решения задач ЛП

Методы линейного программирования - численные методы решения оптимизационных задач, сводящихся к формальным моделям линейного программирования.

Существуют конечные и итеративные методы решения задач ЛП.

Итеративный метод сходится к точному решению асимптотически на бесконечном числе итераций, а при конечном дает приближенное решение.

Конечные методы позволяют получить точное решение за конечное число шагов. Они делятся на

- универсальные, применимые для любых задач ЛП;
- специальные, предназначенные для решения определенных классов задач ЛП.

Последние учитывают специфику класса и поэтому более эффективны, чем универсальные.

Итеративные методы интересны теоретически, а на практике находят применение конечные методы. Начало развитию методов линейного программирования было положено в работах Канторовича Л.В. в конце тридцатых годов 20 века. Позднее был разработан целый ряд эффективных методов, первым из которых стал симплекс-метод.

4.9. Симплекс-метод

4.9.1. Характеристика метода

Данный метод является основополагающим среди универсальных методов. Он разработан Данцигом в 1947 году. В одной из первых задач, решенных этим методом, допустимое множество представляло собой симплекс – выпуклый многогранник, у которого число вершин на единицу больше размерности пространства. Отсюда и произошло название метода. В некоторых отечественных монографиях (авторы Гольштейн Е.Г., Юдин Д.Б. и др.) его называют методом последовательного улучшения плана.

Симплекс-метод реализует направленный перебор допустимых базисных решений в виде итеративного процесса. На каждой итерации осуществляется переход по ребру допустимого множества от одной вершины (крайней точки) к другой, смежной исходной, в которой значение критерия лучше или, в редких случаях, не хуже, чем в исходной. Поскольку число крайних точек конечно, а целевая функция линейна, то такой процесс сходится за конечное число шагов к глобальному оптимуму (для разрешимой задачи).

В результате симплекс метод позволяет отыскать оптимальное решение, просматривая значительно меньше вершин по сравнению с их общим числом. Строгих оценок достаточного числа итераций для достижения оптимального решения нет. Однако экспериментально установлено, что реальное число итераций находится в пределах $m \div 3m$, а наиболее вероятно $(1,5 \div 2)m$. Так, для вышерассмотренного примера вместо десятков тысяч вершин метод “пройдет” не более 30.

В симплекс-методе можно выделить три основные компоненты:

- 1) Способ построения начального базисного решения.
- 2) Процедуру перехода от одного базисного решения к другому.
- 3) Признак оптимальности.

Неразрешимость задачи определяется по ходу работы алгоритма.

4.9.2. Переход от одного базисного решения к другому

Здесь нам понадобятся некоторые понятия линейной алгебры..

Векторы A_1, A_2, \dots, A_S являются *линейно-независимыми*, если равенство $k_1A_1 + k_2A_2 + \dots + k_SA_S = 0$ выполняется только при $k_1 = k_2 = \dots = k_S = 0$. Признаком линейной независимости векторов является ненулевое значение определителя, составленного из этих векторов, так как однородная система имеет единственное (нулевое) решение только при таком определителе.

Если есть система линейно-независимых векторов, то любой другой вектор может быть выражен в виде их *линейной комбинации* и притом единственным образом:

$$A_p = \alpha_1 A_1 + \alpha_2 A_2 + \dots + \alpha_s A_s, \quad p \notin [1, S].$$

В канонической форме условия записываются в виде

$$\sum_{j=1}^{\bar{n}} A_j x_j = B. \quad (4.4)$$

Пусть система (4.4) имеет базисное решение:

$$\begin{cases} x_j = x_j^{(0)} > 0, & j = \overline{1, m}; \\ x_j = 0, & j > m. \end{cases} \quad (4.5)$$

Тогда из (4.4) следует

$$\sum_{i=1}^m A_i x_i^{(0)} = B. \quad (4.6)$$

Так как система (4.6) совместна, то ее определитель не равен нулю и, значит, векторы, входящие в (4.6), являются линейно-независимыми. Для их обозначения введем следующее понятие: система m линейно-независимых векторов, соответствующих базисным переменным, называется *базисом*. Таким образом, каждой экстремальной точке соответствует своё базисное решение и свой базис.

Теперь, имея исходные базисное решение (4.5) и базис $\{A_i\}^{(0)} = \{A_1, A_2, \dots, A_m\}$, построим новое базисное решение. Как следует из геометрических представлений, смежные вершины отличаются по составу базисных переменных только одной. Поэтому новое решение можно получить вводом в исходное небазисной переменной с одновременным переводом одной базисной переменной в небазисные.

Пусть вводимой будет переменная с индексом $r \notin [1, m]$, принимающая в новом решении некоторое положительное значение $x_r^{(1)} = \theta > 0$.

В новом решении, как в любом допустимом, условия (4.4) также должны выполняться, поэтому имеем:

$$\sum_{i=1}^m A_i x_i^{(1)} + A_r x_r^{(1)} = B. \quad (4.7)$$

Задача состоит в том, чтобы определить $X^{(1)}$ по $X^{(0)}$. С этой целью сделаем несложные преобразования. Выразим вектор A_r через исходный базис:

$$A_r = A_1 \alpha_{1r} + A_2 \alpha_{2r} + \dots + A_m \alpha_{mr}. \quad (4.8)$$

Так как известен базис, то известны (или находятся решением этой системы) коэффициенты разложения α_{ir} . Умножим левую и правую части равенства (4.8) на θ :

$$\theta A_r = \theta A_1 \alpha_{1r} + \theta A_2 \alpha_{2r} + \dots + \theta A_m \alpha_{mr}. \quad (4.9)$$

Вычитая (4.9) из (4.6), получим:

$$\sum_{i=1}^m A_i x_i^{(0)} - \theta \sum_{i=1}^m A_i \alpha_{ir} = B - A_r \theta,$$

или окончательно:

$$\sum_{i=1}^m A_i (x_i^{(0)} - \theta \alpha_{ir}) + A_r \theta = B. \quad (4.10)$$

Сравнивая равенства (4.7) и (4.10), видим, что правые части равны, а левые содержат одну и ту же систему векторов. Поэтому коэффициенты при одноименных векторах должны совпадать. Приравнявая их, получаем искомые соотношения:

$$\begin{cases} x_i^{(1)} = x_i^{(0)} - \theta \alpha_{ir}, & i = \overline{1, m}, \\ x_r^{(1)} = \theta, & i = r. \end{cases} \quad (4.11)$$

Однако решение (4.11) может быть недопустимым, если не оговорить возможные значения θ . Предположим, что среди коэффициентов α_{ir} есть положительные. Тогда с увеличением значения θ соответствующие переменные могут стать отрицательными. Поэтому для допустимости решения $X^{(1)}$ необходимо, чтобы θ было ограничено сверху:

$$\theta \leq \theta_0 = \min \frac{x_i^{(0)}}{\alpha_{ir}} \quad \text{для } \alpha_{ir} > 0. \quad (4.12)$$

С учетом (4.12) решение (4.11) всегда будет допустимым, но число ненулевых переменных в нем может превышать m , так как добавлена x_r , а значит, оно может быть небазисным. Если же в качестве значения θ выбрать θ_0 , то одна из переменных $x_i^{(1)}$, $i \in \overline{1, m}$ станет равной нулю, а решение (4.11) – базисным.

Пусть минимум в (4.12) достигается на переменной x_k . Тогда базисные переменные в новом опорном решении будут вычисляться по формулам:

$$\begin{cases} x_1^{(1)} = x_1^{(0)} - \theta_0 \alpha_{1r}, \\ x_2^{(1)} = x_2^{(0)} - \theta_0 \alpha_{2r}, \\ \dots \\ x_{k-1}^{(1)} = x_{k-1}^{(0)} - \theta_0 \alpha_{k-1,r}, \\ x_{k+1}^{(1)} = x_{k+1}^{(0)} - \theta_0 \alpha_{k+1,r}, \\ \dots \\ x_m^{(1)} = x_m^{(0)} - \theta_0 \alpha_{mr}, \\ x_r^{(1)} = \theta_0. \end{cases} \quad (4.13)$$

Этому решению соответствует новый базис $\{A_i\}^{(1)} = \{A_1, \dots, A_{k-1}, A_r, A_{k+1}, \dots, A_m\}$. Таким образом, переход к новому базисному решению произошел путем замены переменной X_k на X_r , соответственно в базисе – A_k на A_r .

Рассмотрим возможные ситуации при переходе от одного решения к другому. Как было показано выше, при существовании положительных

коэффициентов α_{ir} достигается новое **базисное** решение (смежная вершина), что иллюстрируется рис. 4.6-а. Если же все α_{ir} неположительны, величина θ , а это значение вводимой переменной, не ограничена сверху. Следовательно, введение такой переменной не приведет к получению базисного решения (достижению новой вершины). Это признак того, что соответствующее ребро допустимого множества, а значит, и само множество оказываются неограниченными (рис. 4.6-б).

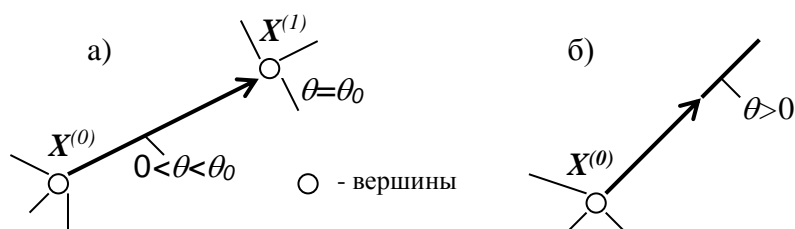


Рис. 4.6

При вычислении θ_0 минимум может достигаться более чем на одном индексе. При этом обнуляется более одной переменной из входящих в исходное решение. Следовательно, в новом решении будут базисные переменные с нулевым значением, что означает попадание в **вырожденное базисное** решение.

Если исходное решение вырожденное и нулевой переменной соответствует коэффициент $\alpha_{kr} > 0$, то согласно (4.12) $\theta_0 = 0$ и значения переменных не изменяются. Однако состав базиса и базисных переменных изменится – произойдет замена $x_k^{(0)} = 0$ на $x_r^{(1)} = 0$. При высокой степени вырожденности теоретически возможно закливание, то есть возврат к старому базису, но на практике такие случаи не встречались.

4.9.3. Признак оптимальности. Основные этапы симплекс-метода

Имея текущее базисное решение, необходимо выяснить, является ли оно оптимальным. При положительном ответе процедура поиска оптимума завершается. В противном случае значение критерия может быть улучшено при правильном выборе нового опорного решения (смежной вершины). Признак оптимальности и позволяет установить статус решения и определить последующие действия.

Он легко выводится из связи смежных решений. Пусть на k -й итерации получили базисное решение $x^{(k)}$ с критерием $L^{(k)}$. При смещении из этой вершины на ребро (см. рис. 4.6 -а) решение изменяется согласно (4.11):

$$\begin{aligned} x^{k+1}_i &= x^{(k)}_i - \theta \alpha_{ir}, \quad i=1, 2, \dots, m. \\ x^{k+1}_r &= \theta. \end{aligned}$$

Используя приведенные соотношения, определим критерий в новом решении:

$$\begin{aligned} L^{(k+1)} &= \sum_i C_i x_i^{(k+1)} + C_r x_r^{(k+1)} = \sum_i C_i (x_i^{(k)} - \theta \alpha_{ir}) + C_r \theta = \\ &= \sum_i C_i x_i^{(k)} - \sum_i C_i \theta \alpha_{ir} + C_r \theta = L^{(k)} - \theta \left(\sum_i C_i \alpha_{ir} - C_r \right) \end{aligned}$$

или, введя обозначения

$$z_r = \sum_i C_i \alpha_{ir}, \quad (4.14)$$

$$\Delta_r = z_r - C_r, \quad (4.15)$$

получаем

$$L^{(k+1)} = L^{(k)} - \theta \Delta_r. \quad (4.16)$$

Параметр Δ_r называют **относительной оценкой переменной**. Ее смысл очевиден, если вспомнить, что θ - значение новой переменной. Δ_r показывает, как изменится значение критерия при введении единицы новой переменной ($\theta=1$). Поэтому, если есть отрицательные оценки, текущее решение может быть улучшено при введении в число базисных соответствующей переменной.

Если же все оценки будут больше или равны нулю, то тогда ни одна переменная не может улучшить текущее решение, следовательно,

условие $\forall \Delta_j \geq 0$ и является признаком оптимальности.

Покажем, что для базисных переменных относительные оценки равны нулю. Так как базисный вектор не выражается через другие, то только один коэффициент $\alpha_{rr}=1$, остальные равны нулю. Из (4.14) следует $z_r=C_r$, а из (4.15) - $\Delta_r=0$. Таким образом, достаточно проверять знаки оценок только небазисных переменных.

Полученным формулам можно дать экономическую интерпретацию.

Пусть рассматривается задача планирования, в которой X_j – количество продукции j -го вида, C_j – стоимость единицы произведенной продукции.

Тогда $\mathbf{X}^{(0)}$ – план производства, включающий первые m видов продукции. Попытаемся изменить этот план. Включим в него еще один вид продукции r . Так как ресурсы не меняются, это возможно только при одновременном уменьшении продукции, входящей в план $\mathbf{X}^{(0)}$. Величина этого изменения определяется коэффициентами α_{ir} . Действительно, как следует из (4.11), α_{ir} показывает, насколько должно измениться производство продукции i -го вида при введении в план единицы продукции r . В экономической литературе такой показатель называют *маргинальной нормой замещения*. Значит, объем выпуска каждого вида продукции сокращается на α_{ir} , а суммарная стоимость на величину $Z_r = \sum_i C_i \alpha_{ir}$. Поэтому Z_r трактуют как *маргинальную цену* (снижение

стоимости произведенной продукции на каждую единицу продукции r). В то же время, единица продукции r дает прирост стоимости C_r , называемый *маргинальным доходом*. Очевидно, что если $C_r > Z_r$, то есть $\Delta_r < 0$, то такая корректировка плана выгодна.

Если несколько переменных имеют отрицательные оценки, то возникает необходимость выбора из них самой перспективной. Очевидно, что наибольшее улучшение критерия на очередном шаге даст переменная, для которой произведение $\theta_0 \Delta_j$ минимально. Однако такой выбор требует большого объема вычислений, так как для каждой переменной с отрицательной оценкой требуется находить θ_0 , а таких переменных в реальной задаче могут быть тысячи. Поэтому обычно выбирают переменную с наименьшей оценкой.

Таким образом, укрупненно можно выделить следующие этапы симплекс-метода:

1. Построение начального неотрицательного базисного решения.
2. Анализ оценок. При этом возможны три ситуации:
 - а) все оценки неотрицательны, следовательно, вычисления прекращаются, так как выполнен признак оптимальности.
 - б) имеются отрицательные оценки, но, по крайней мере, одной из них (например, Δ_j) соответствует вектор, для которого все коэффициенты разложения неположительны ($\forall \alpha_{ij} \leq 0$). В этом случае θ не ограничено сверху и, как следует из (4.16), критерий неограниченно возрастает. Решение прекращается.
 - в) для каждой отрицательной оценки есть $\alpha_{ij} > 0$. Решение может быть улучшено выполнением 3 этапа.
3. Переход к новому базисному решению путем введения переменной с минимальной оценкой и выводом переменной, на которой достигается минимум в (4.12).

Этапы 2 и 3 повторяются до выполнения одного из условий останова.

Из перечисленных этапов не раскрытым остался первый этап, рассматриваемый ниже.

4.9.4. Построение начального базисного решения

Симплекс-метод основан на переходах от одного допустимого базисного решения к другому (смежному). Как показано ранее, базисное решение включает m неотрицательных переменных, называемых базисными, при нулевых значениях остальных (небазисных или свободных) переменных.

Чтобы начать движение к оптимуму, необходимо иметь начальное базисное решение. Оно может быть получено из модели, представленной в канонической форме. При этом выбор базисных переменных зависит от вида условий исходной модели, но в любом случае каждому условию

соответствует своя базисная переменная (предполагается линейная независимость m условий).

Рассмотрим возможные варианты построения начального решения.

1. Исходная модель представлена неравенствами “ \leq ”:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}, \quad (b_i \geq 0).$$

Для приведения к каноническому виду в каждое неравенство вводится дополнительная переменная:

$$\begin{cases} \sum_{j=1}^n a_{1j} x_j + x_{n+1} = b_1; \\ \dots \\ \sum_{j=1}^n a_{mj} x_j + x_{n+m} = b_m; \end{cases} \quad \forall x_{n+i} \geq 0.$$

Если положить $x_j=0, j=1, 2, \dots, n$, то дополнительные переменные $x_{n+i}=b_i \geq 0 \ (i=1, 2, \dots, m)$ удовлетворяют всем требованиям допустимого базисного решения: выполняются все условия задачи и число базисных переменных равно m . Очевидно, что этому базисному решению соответствует **единичный** базис $\{A_i\}^{(0)} = \{A_{n+1}, A_{n+2}, \dots, A_{n+m}\}$. В этом случае не надо вычислять коэффициенты разложения небазисных векторов по базису. Действительно, в системе уравнений

$$A_j = \sum_{i=1}^m A_{n+i} \alpha_{n+i,j}$$

каждый коэффициент входит только в одно уравнение с множителем +1. Поэтому ее решением будет

$$\alpha_{n+i,j} = a_{ij},$$

то есть коэффициенты разложения равны соответствующим компонентам раскладываемого вектора условий.

2. Исходная модель представлена неравенствами “ \geq ”:

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = \overline{1, m}.$$

Тогда соответствующая каноническая модель

$$\begin{cases} \sum_{j=1}^n a_{1j} x_j - x_{n+1} = b_1, \\ \dots \\ \sum_{j=1}^n a_{mj} x_j - x_{n+m} = b_m, \end{cases} \quad \forall x_{n+i} \geq 0$$

включает дополнительные переменные со знаком “минус”. Если из них образовать базисное решение, то оно будет недопустимым, так как из модели следует $x_{n+i} = -b_i < 0, \forall i$.

В этом случае строится **искусственное** базисное решение, в котором все переменные неотрицательные, но не выполняется часть функциональных ограничений. Здесь возможны два варианта.

В первом из них в каждое равенство канонической модели вводится **искусственная** переменная $x_{n+m+i} > 0$:

$$\sum_{j=1}^n a_{ij}x_j - x_{n+i} + x_{n+m+i} = b_i, \forall i.$$

Полагая все исходные и дополнительные переменные равными нулю, получаем искусственное базисное решение $x_{n+m+i} = b_i > 0, i = \overline{1, m}$. Очевидно, что в нем все исходные неравенства не выполняются. Векторы с одноименными индексами образуют начальный единичный базис.

Этот прием очень простой, но приводит к значительному увеличению числа переменных. Второй вариант устраняет этот недостаток.

Найдем в канонической модели уравнение с наибольшей правой частью. Пусть таким будет последнее уравнение, то есть.

$$b_m = \max_i b_i.$$

Вычитая из него по отдельности каждое уравнение (кроме его самого), получаем преобразованную систему

$$\left\{ \begin{array}{l} \sum_j a'_{1j}x_j - x_{n+m} + x_{n+1} = b'_1, \\ \sum_j a'_{2j}x_j - x_{n+m} + x_{n+2} = b'_2, \\ \dots \\ \sum_j a'_{m-1,j}x_j - x_{n+m} + x_{n+m-1} = b'_{m-1}, \\ \sum_j a_{mj}x_j - x_{n+m} = b_m, \end{array} \right.$$

где $a'_{ij} = a_{mj} - a_{ij}, \quad b'_i = b_m - b_i$.

Если теперь положить $x_j=0$ ($j=1,2,...,m$) и $x_{n+m}=0$, то дополнительные переменные $x_{n+i}=b'_i \geq 0$ ($i=1,2,...,m-1$) могут быть приняты в качестве базисных. Однако при этом не хватает одной базисной переменной и последнее уравнение не выполняется. Введем в него искусственную переменную x_{m+n+1}

$$\sum_j a_{mj}x_j - x_{n+m} + x_{m+n+1} = b_m,$$

которая и будет недостающей базисной переменной ($x_{m+n+1}=b_m$). Таким образом, получено искусственное базисное решение, содержащее независимо от числа ограничений только одну искусственную переменную. Соответствующий ему базис, как и ранее, является единичным:

$$\{A_i\}^{(0)} = \{A'_{n+1}, A'_{n+2}, \dots, A'_{n+m-1}, A'_{n+m+1}\}.$$

При переходе от одного базисного решения к другому допустимое решение достигается только тогда, когда **все искусственные переменные** станут равными нулю. Для ускорения вывода этих переменных из числа базисных (обнуления) рекомендуется придавать им большой негативный вес путем модификации критерия:

$$L' = L - M \sum_{i=1}^m x_{n+m+i}, \text{ для первого варианта,}$$

$$L' = L - Mx_{m+n+1}, \text{ для второго варианта,}$$

где M - большое положительное число, такое, что $M \gg \max |C_j|$ (в задаче на минимум знак минус заменяется на плюс).

Если при выполнении признака оптимальности хотя бы одна искусственная переменная останется положительной, то это будет означать, что задача неразрешима из-за противоречивости условий: не выполняться будут те условия, в которые входят ненулевые искусственные переменные.

3.В исходной модели условия заданы в виде равенств

$$\sum_{j=1}^n a_{ij}x_j = b_i \geq 0, \quad i = \overline{1, m}.$$

Для построения базисного решения введем в каждое равенство искусственную переменную:

$$\sum_{j=1}^n a_{ij}x_j + x_{n+i} = b_i.$$

Тогда базисное решение будет состоять из искусственных переменных $x_{n+i} = b_i, i = \overline{1, m}$, базис – из единичных векторов при этих переменных, а исходный критерий модифицируется:

$$L' = L - M \sum_{i=1}^m x_{n+i}.$$

Число искусственных переменных может быть меньше, если в исходной системе есть переменные, входящие со знаком плюс только в одно уравнение. Такие переменные принимаются за базисные, а искусственные переменные в соответствующие условия не вводятся..

4.Исходная модель содержит все виды ограничений (общий случай).

Предварительно условия группируются по виду. В каждой группе определяются базисные переменные одним из способов, описанных выше.

Очевидно, что при таком подходе начальный базис будет единичным и, следовательно, не потребуются вычислять коэффициенты разложения небазисных векторов в начальном решении.

4.9.5. Связь между параметрами последовательных итераций

Процедура перехода от вершины к вершине связана с пересчетом большого числа параметров. К ним относятся коэффициенты разложения α_{ij} , относительные оценки Δ_j , базисные переменные и критерий. В симплекс-методе для их вычисления применяются обобщенные преобразования Жордана – Гаусса.

Пусть $\{A_i\}^{(0)}$, $i \in I^0 = \{1, 2, \dots, k, \dots, m\}$ – исходный базис. Смежный базис отличается от него только одной компонентой:

$$\{A_i\}^{(1)}, i \in I^1 = \{1, 2, \dots, r, \dots, m\}.$$

Если из множества I^0 исключить индекс k , а из I^1 удалить r , то эти множества будут тождественно равны.

Возьмем вектор A_j такой, что $j \notin I^0$ & $j \notin I^1$. Он может быть представлен как через исходный, так и смежный базисы:

$$A_j = \sum_{i \in I^0} A_i \alpha_{ij} \quad (4.17)$$

$$A_j = \sum_{i \in I^1} A_i \alpha'_{ij} \quad (4.18)$$

Зная коэффициенты разложения по исходному базису α_{ij} , найдем коэффициенты разложения по смежному базису α'_{ij} . Для этого в (4.17) заменим вектор A_k на вектор A_r . Из представления A_r через исходный базис

$$A_r = \sum_{i \in I^0} A_i \alpha_{ir} = \sum_{\substack{i \in I^0 \\ i \neq k}} A_i \alpha_{ir} + A_k \alpha_{kr} \quad (4.19)$$

получаем

$$A_k = \frac{1}{\alpha_{kr}} \left(A_r - \sum_{\substack{i \in I^0 \\ i \neq k}} A_i \alpha_{ir} \right). \quad (4.20)$$

В (4.17) выделим слагаемое с вектором A_k

$$A_j = \sum_{\substack{i \in I^0 \\ i \neq k}} A_i \alpha_{ij} + A_k \alpha_{kj}$$

и заменим в нем A_k выражением (4.20):

$$A_j = \sum_{\substack{i \in I^0 \\ i \neq k}} A_i \alpha_{ij} + \frac{\alpha_{kj}}{\alpha_{kr}} \left(A_r - \sum_{\substack{i \in I^0 \\ i \neq k}} A_i \alpha_{ir} \right).$$

После преобразования получаем:

$$A_j = \sum_{\substack{i \in I^0 \\ i \neq k}} A_i \left(\alpha_{ij} - \frac{\alpha_{kj}}{\alpha_{kr}} \cdot \alpha_{ir} \right) + A_r \cdot \frac{\alpha_{kj}}{\alpha_{kr}} \quad (4.21)$$

В формулах (4.18) и (4.21) вектор A_j представлен через одну и ту же систему векторов. Приравнявая коэффициенты при одноименных векторах, находим искомые зависимости:

$$\alpha'_{ij} = \begin{cases} \alpha_{ij} - \frac{\alpha_{kj}}{\alpha_{kr}} \cdot \alpha_{ir}, & i = 1, 2, \dots, k-1, k+1, \dots, m; \\ \frac{\alpha_{kj}}{\alpha_{kr}}, & i = r; \\ \overline{\alpha_{ij}}, & j = \overline{1, n}, \end{cases} \quad (4.22)$$

где \overline{n} - число переменных в канонической модели.

В базисном решении выполняется равенство

$$\sum_{i=1}^m A_i x_i = B,$$

в котором x_i - базисные переменные. Переобозначим вектор ограничений B на A_0 :

$$\sum_{i=1}^m A_i x_i = A_0.$$

Отсюда очевидно, что x_i - это коэффициенты разложения вектора ограничений по текущему базису, то есть с учетом принятой индексации

$$A_0 = \sum_{i=1}^m A_i \alpha_{i0} \quad \text{и} \quad x_i = \alpha_{i0}.$$

Значит, для базисных переменных справедливы соотношения (4.22). Этого и следовало ожидать, так как полученные ранее формулы (4.11) и (4.12) для пересчета базисных переменных являются частным случаем (4.22). Действительно, достаточно заменить в них x_i на α_{i0} , чтобы прийти к (4.22).

Теперь покажем, что относительные оценки и критерий также могут рассматриваться как коэффициенты разложения. Очевидно, что соотношения (4.22) справедливы для любых смежных базисов и не зависят от размерности последних.

Введем расширенные векторы следующим образом:

$$\overline{A}_j = \begin{bmatrix} A_j \\ C_j \end{bmatrix}, \quad j = \overline{0, n}.$$

Так как размерность векторов увеличилась на единицу, базис должен включать $m+1$ векторов. В качестве недостающего базисного вектора возьмем единичный вектор

$$\bar{A}_{m+1} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \\ 1 \end{bmatrix},$$

в котором единица стоит на позиции $m+1$. Такой вектор линейно независим от расширенных векторов условий. Как и раньше, рассмотрим два смежных расширенных базиса, образованных векторами с индексами $\bar{I}^0 = \{I^0, m+1\}$ и $\bar{I}^1 = \{I^1, m+1\}$.

Представим небазисный вектор \bar{A}_j через расширенный базис:

$$\bar{A}_j = \sum_{i \in I^0} \bar{A}_i \bar{\alpha}_{ij} + \bar{A}_{m+1} \alpha_{m+1,j}. \quad (4.23)$$

Так как первые m компонент вектора \bar{A}_{m+1} равны нулю, первые m уравнений в (4.23) тождественны (4.17) и, следовательно, $\bar{\alpha}_{ij} = \alpha_{ij}$. Новыми здесь являются только коэффициенты $\alpha_{m+1,j}$. Чтобы выяснить их смысл, запишем последнее уравнение системы (4.23):

$$C_j = \sum_{i \in I^0} C_i \alpha_{ij} + \alpha_{m+1,j}.$$

Отсюда имеем

$$\alpha_{m+1,j} = C_j - \sum_{i \in I^0} C_i \alpha_{ij} = C_j - Z_j = -\Delta_j, \quad j = \overline{1, n};$$

$$\alpha_{m+1,0} = 0 - \sum_{i \in I^0} C_i \alpha_{i0} = -\sum C_i x_i = -L,$$

то есть оценки и критерий тоже являются коэффициентами разложения.

Таким образом, все параметры симплекс-метода математически представляют собой коэффициенты разложения, а рекуррентные формулы (4.22) справедливы для $i = \overline{1, m+1}$, $j = \overline{0, n}$.

4.9.6. Алгоритм симплекс-метода

Резюмируя изложенное в предыдущих разделах, опишем алгоритм симплекс-метода. По модели в каноническом виде определяется начальное базисное решение. Последующие действия проводятся в специальных таблицах, называемых **симплекс-таблицами**. В них представляются результаты итераций, которые завершаются при выполнении признака оптимальности или обнаружении неразрешимости задачи.

Полная симплекс-таблица имеет следующую структуру.

Таблица /			C_0	C_1	C_2	...	C_r	...	$C_{\bar{n}}$	θ
i	C_{si}	базис A_{si}	$B=A_0$	A_1	A_2	...	A_r	...	$A_{\bar{n}}$	
1	C_{s1}	A_{s1}	$\alpha_{10}=x_{s1}$	α_{11}	α_{12}	...	α_{1r}	...	$\alpha_{1\bar{n}}$	
2	C_{s2}	A_{s2}	$\alpha_{20}=x_{s2}$	α_{21}	α_{22}	...	α_{2r}	...	$\alpha_{2\bar{n}}$	
...
k	C_{sk}	A_{sk}	$\alpha_{k0}=x_{sk}$	α_{k1}	α_{k2}	...	α_{kr}	...	$\alpha_{k\bar{n}}$	θ_0
...
m	C_{sm}	A_{sm}	$\alpha_{m0}=x_{sm}$	α_{m1}	α_{m2}	...	α_{mr}	...	$\alpha_{m\bar{n}}$	
$m+1$	$-\alpha_{m+1,j}$		L	Δ_1	Δ_2	...	Δ_r	...	$\Delta_{\bar{n}}$	
$m+2$	z_j		z_0	z_1	z_2	...	z_r	...	$z_{\bar{n}}$	

Здесь C_j – коэффициенты линейной формы L , C_{si} – коэффициенты в L при базисных переменных (подмножество C_j), A_{si} – базисные векторы, si – индекс базисного компонента на позиции i . Жирной линией выделена главная часть таблицы, все ее элементы подчиняются соотношениям (4.22). Первая и последняя строки и второй столбец таблицы являются вспомогательными, они обязательны только в начальной таблице.

Примечание. В линейной алгебре во второй строке и третьем столбце таблицы используют обозначения переменных x_j и x_i вместо векторов A_j и A_{si} соответственно. Поскольку элементы главной части таблицы являются коэффициентами разложения векторов, принятые здесь обозначения, на наш взгляд, более корректны.

Алгоритм состоит из предварительного и основного этапов.

На предварительном этапе сначала определяется начальное базисное решение одним из методов, рассмотренных выше. Исходя из него заполняется начальная симплекс-таблица. В третий столбец заносятся m базисных векторов, а во второй – соответствующие коэффициенты из L (или из верхней строки) в порядке следования условий в модели (на первую позицию ставится вектор при базисной переменной из первого условия и т.д.).

Так как начальный базис единичный, элементы главной части таблицы кроме последней строки не вычисляются, а берутся прямо из модели (в столбец A_0 заносятся правые части условий, в A_j – коэффициенты при x_j).

Для получения относительных оценок используются формулы (4.11) и (4.12). Значения z_j находятся как скалярное произведение векторов $C_b=[C_{si}]$

и A_j (суммируются произведения одноименных компонент). Вычитая из нижней строки верхнюю, получаем L и оценки Δ_j .

Основной этап является итерационным.

Очередная итерация заканчивается заполнением симплекс-таблицы за исключением столбца θ . Пусть завершилась l -я итерация. Цикл начинается с анализа оценок Δ_j в таблице l . Если нет отрицательных оценок, значит, выполнен признак оптимальности. В этом случае возможны два вывода:

- 1) если не вводились искусственные переменные или они равны нулю, получено оптимальное решение;
- 2) если хотя бы одна искусственная переменная не равна нулю, задача неразрешима из-за противоречивости условий.

При невыполнении признака оптимальности анализируются столбцы с отрицательными оценками. Если среди них обнаружится столбец, в котором все коэффициенты разложения неположительны, то есть $\alpha_{ij} \leq 0, \forall i$, то задача неразрешима по причине неограниченности критерия на допустимом множестве. В противном случае выбирается минимальная (отрицательная) оценка

$$\Delta_r = \min \Delta_j.$$

Она определяет столбец A_r , называемый **направляющим** или разрешающим, или ведущим столбцом. Мы будем придерживаться первого термина.

Заполняется столбец θ . Значения θ вычисляются делением элементов столбца A_0 на положительные элементы направляющего столбца

$$\theta_i = \frac{x_{si}}{\alpha_{ir} > 0}.$$

По минимальному значению θ определяется **направляющая строка** k :

$$\min \theta_i = \theta_k.$$

На пересечении направляющей строки и направляющего столбца находится **направляющий элемент** α_{kr} . Тем самым определена переменная x_r , которая становится базисной, и переменная x_{sk} , выводимая из числа базисных (она становится равной нулю).

Заполняется таблица $l+1$. В ней отражается смена базиса: вектор A_{sk} заменяется вектором A_r , соответственно вместо C_{sk} ставится C_r , остальные базисные элементы остаются на месте. Элементы главной части таблицы вычисляются согласно (4.22):

$$\alpha_{ij}^{l+1} = \begin{cases} \alpha_{ij}^l - \frac{\alpha_{kj}^l \cdot \alpha_{ir}^l}{\alpha_{kr}^l}, & i \neq r; \\ \frac{\alpha_{kj}^l}{\alpha_{kr}^l}, & i = r. \end{cases}$$

Эти формулы применяются следующим образом. Элементы строки, которая была направляющей, находятся делением строки на направляющий элемент. Для вычисления остальных элементов можно использовать правило прямоугольника: в таблице l элемент α_{ij}

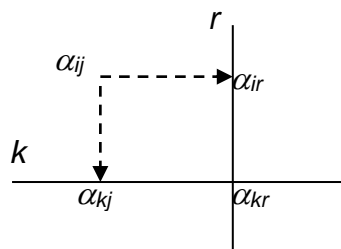


Рис.4.7

проектируется на направляющий столбец и направляющую строку (рис.4.7). В вершинах образовавшегося прямоугольника находятся все элементы, входящие в рекуррентную формулу. Теперь, вычитая из проектируемого элемента произведение элементов в двух других противолежащих вершинах прямоугольника, деленное на направляющий элемент, получаем

новое значение элемента. При этом так вычисляются элементы только небазисных столбцов, так как в базисных столбцах всегда имеем единичные векторы.

После заполнения главной части таблицы возвращаемся на начало основного этапа.

Для контроля вычислений можно проводить повторный счет оценок, используя вспомогательные строки z и C .

Наглядное представление алгоритма дает блок-схема, приведенная на рис. 4.8.

Замечание. При выборе направляющего столбца и направляющей строки может иметь место неоднозначность из-за достижения минимума более чем на одном индексе. При этом можно выбирать любой из них либо для однозначного выбора добавить правило, например, при нескольких индексах брать наименьший.

Из неоднозначности выбора строки следует, что новое базисное решение будет вырожденным. При степени вырожденности больше единицы теоретически возможно закливание. Для его устранения в теории предложена ε -задача, соответствующая малому деформированию вектора ограничений, которое приводит к замене вырожденной вершины невырожденными. Из решения этой задачи выведено более сложное правило выбора направляющей строки:

В строках с минимальным θ находятся отношения θ^I элементов 1-го столбца к элементам направляющего и выбирается строка с минимальным θ^I . Если же этот выбор снова неоднозначен, то вычисляются отношения θ^{II} элементов 2-го и направляющего столбца в строках с минимальным θ^I , и т.д. до достижения однозначного выбора. Это правило гарантирует от закливания. Однако на реальных задачах сталкиваться с закливанием не приходилось, и поэтому изложенное правило представляет больше теоретический интерес.

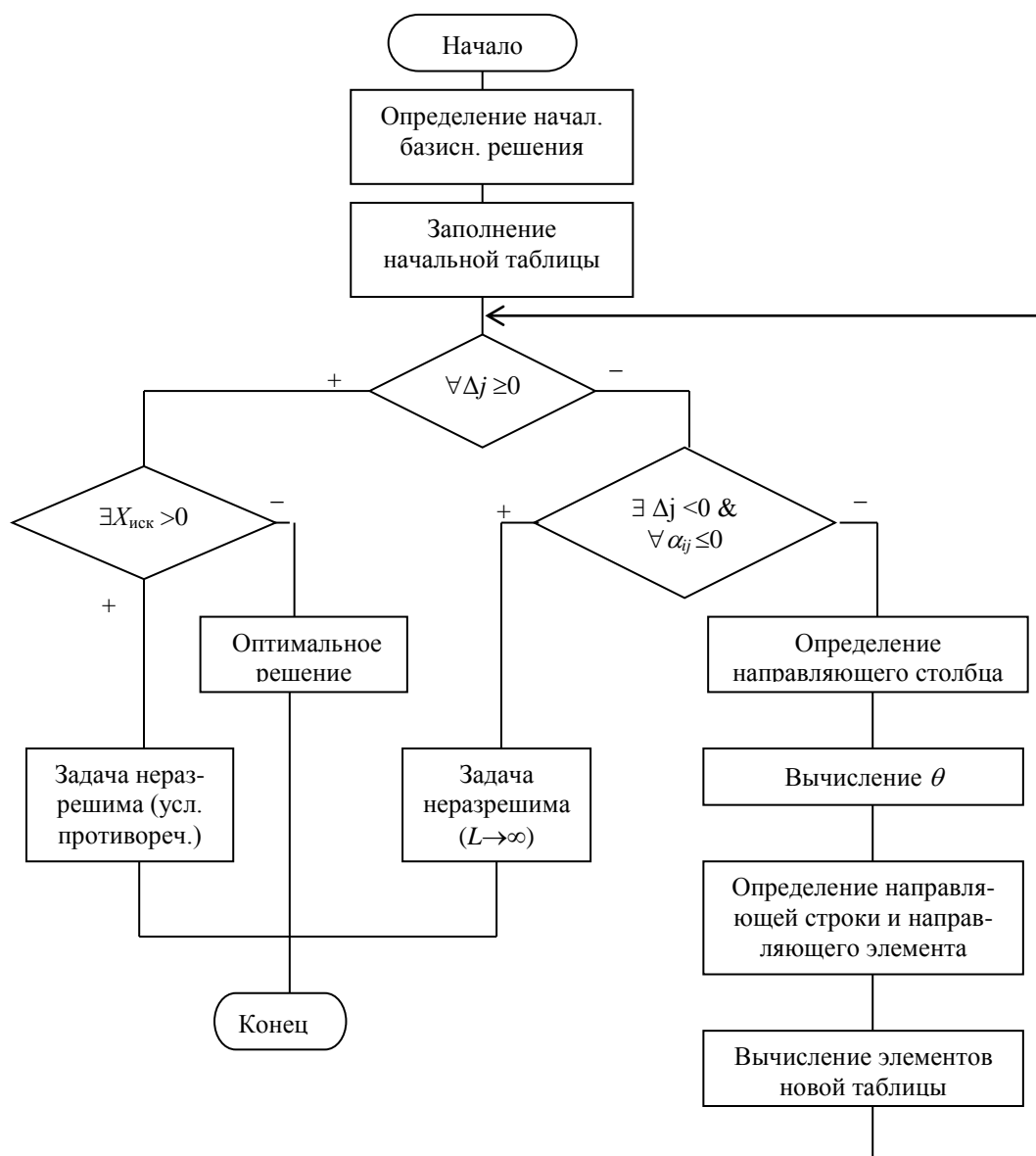


Рис. 4.8

4.9.7. Примеры

Пример 4.2. Для иллюстрации работы алгоритма применим его к задаче планирования, решенной графически в разд. 4.6.

Исходная модель

$$\begin{aligned}
 &L = 7x_1 + 5x_2 \rightarrow \max, \\
 &1) \ 2x_1 + 3x_2 \leq 19, \\
 &2) \ 2x_1 + x_2 \leq 13, \\
 &3) \ 3x_2 \leq 12, \\
 &4) \ 3x_1 \leq 17, \\
 &5) \ x_1 \geq 0, \\
 &6) \ x_2 \geq 0.
 \end{aligned}$$

Каноническая модель

$$\begin{aligned}
 &L = 7x_1 + 5x_2 \rightarrow \max, \\
 &2x_1 + 3x_2 + x_3 = 19, \\
 &2x_1 + x_2 + x_4 = 13, \\
 &3x_2 + x_5 = 12, \\
 &3x_1 + x_6 = 17, \\
 &\forall x_j \geq 0.
 \end{aligned}$$

Исходная модель соответствует первому случаю построения начального базисного решения (см. разд. 4.9.4). Следовательно, базисными переменными в начальном решении будут дополнительные переменные x_3 , x_4 , x_5 и x_6 , а базисными векторами – A_3 , A_4 , A_5 и A_6 . Очевидно, что такое базисное решение является допустимым (опорным планом), а в геометрическом представлении это вершина в начале координат.

Имея начальное решение, заполняем начальную симплекс-таблицу. В столбцы $A_1 - A_6$ заносятся коэффициенты при переменных x_1-x_6 в канонической модели, а в A_0 – свободные члены. Так как C_3 , C_4 , C_5 и C_6 равны нулю, то согласно (4.14) и все $z_j=0$. Следовательно, в этом случае $\Delta_j = z_j - C_j = -C_j$.

Таблица 0			0	7	5	0	0	0	0	θ
i	C_{si}	Баз.	A_0	A_1	A_2	A_3	A_4	A_5	A_6	
1	0	A_3	19	2	3	1	0	0	0	19/2
2	0	A_4	13	2	1	0	1	0	0	13/2
3	0	A_5	12	0	3	0	0	1	0	--
4	0	A_6	17	3	0	0	0	0	1	17/3
5	Δ_j		0	-7	-5	0	0	0	0	
6	z_j		0	0	0	0	0	0	0	

Анализируем таблицу. Признак оптимальности не выполняется и при этом в столбцах с отрицательными Δ_j есть положительные элементы, значит, решение можно улучшить. По минимальному значению Δ_j определяем направляющий столбец – A_1 . Вычисляем θ делением элементов столбца A_0 на положительные элементы направляющего столбца. По минимальному значению θ находим направляющую строку – строка 2. Направляющий столбец и направляющая строка выделены в таблице серым цветом. Таким образом, из базиса выходит вектор A_6 и на его место встает A_1 . Аналогичная смена происходит в базисном решении: x_1 заменяет x_6 .

Переходим к вычислению элементов новой таблицы. Новые значения в строке 4 получаем делением элементов этой строки в начальной таблице на направляющий элемент. Остальные элементы в небазисных столбцах, столбце A_0 и новые значения Δ_j вычисляются по правилу прямоугольника. Для примера в начальной таблице показаны 2 прямоугольника, которые построены на элементах α_{16} и Δ_2 . Они позволяют вычислить новые значения этих элементов:

$$\alpha_{16} = 0 - (2 \cdot 1)/3 = -2/3; \quad \Delta_2 = -5 - (-7 \cdot 0)/3 = -5.$$

В результате получаем симплекс-таблицу 1.

Таблица 1			0	7	5	0	0	0	0	θ
i	C_{si}	Баз.	A_0	A_1	A_2	A_3	A_4	A_5	A_6	
1	0	A_3	23/3	0	3	1	0	0	-2/3	23/9
2	0	A_4	5/3	0	1	0	1	0	-2/3	5/3
3	0	A_5	12	0	3	0	0	1	0	4
4	7	A_1	17/3	1	0	0	0	0	1/3	--
5	Δj		119/3	0	-5	0	0	0	7/3	
6	z_j		119/3	7	0	0	0	0	7/3	

В ней также показаны вспомогательные строки (верхняя и нижняя) и вспомогательный столбец C_{si} . Они предназначены для контроля вычислений оценок (непосредственно по формулам (4.14) и (4.15)), а значит, и всей таблицы. Легко проверить, что в нашем случае эти формулы дают те же значения оценок.

Следующая итерация начинается с проверок на оптимальность и разрешимость задачи, выбора направляющего элемента и заканчивается заполнением симплекс-таблицы 2.

Таблица 2			0	7	5	0	0	0	0	θ
i	C_{si}	Баз.	A_0	A_1	A_2	A_3	A_4	A_5	A_6	
1	0	A_3	8/3	0	0	1	-3	0	4/3	2
2	5	A_2	5/3	0	1	0	1	0	-2/3	--
3	0	A_5	7	0	0	0	-3	1	2	7/2
4	7	A_1	17/3	1	0	0	0	0	1/3	17
5	Δj		48	0	0	0	5	0	-1	
6	z_j		48	7	5	0	5	0	-1	

Так как оптимальное решение не достигнуто, проводим 3-ю итерацию, результаты которой представлены в табл. 3.

В этой таблице нет отрицательных оценок, что свидетельствует о достижении оптимального решения. Максимальная прибыль составляет $L^*=50$ при значениях переменных $x_1^*=5$, $x_2^*=3$, $x_3^*=x_4^*=0$, $x_5^*=3$, $x_6^*=2$, что полностью совпадает с результатами графического решения в разд. 4.6.

Таблица 3			0	7	5	0	0	0	0
i	C_{si}	Баз.	A_0	A_1	A_2	A_3	A_4	A_5	A_6
1	0	A_6	2	0	0	3/4	-9/4	0	1
2	5	A_2	3	0	1	1/2	-1/2	0	0
3	0	A_5	3	0	0	-3/2	3/2	1	0
4	7	A_1	5	1	0	-1/4	3/4	0	0
5	Δ_j		50	0	0	3/4	11/4	0	0
6	z_j		50	7	5	3/4	11/4	0	0

Пример 4.3. Рассмотрим задачу с разными видами ограничений и покажем только отличие от предыдущего примера.

Исходная модель

$$L = 4x_1 - x_2 \rightarrow \max,$$

$$1) \ 3x_1 + 2x_2 \leq 10,$$

$$2) \ 2x_1 + x_2 \geq 8,$$

$$3) \ x_1 - 3x_2 = 12,$$

$$4) \ x_1 \geq 0,$$

$$5) \ x_2 \geq 0.$$

Каноническая модель

$$L = 4x_1 - x_2 \rightarrow \max,$$

$$3x_1 + 2x_2 + x_3 = 10,$$

$$2x_1 + x_2 - x_4 = 8,$$

$$x_1 - 3x_2 = 12,$$

$$\forall x_j \geq 0.$$

Для построения начального базисного решения по 4-му варианту введем искусственные переменные x_5 и x_6 . Тогда модель примет вид

$$L = 4x_1 - x_2 \rightarrow \max,$$

$$3x_1 + 2x_2 + x_3 = 10,$$

$$2x_1 + x_2 - x_4 + x_5 = 8,$$

$$x_1 - 3x_2 + x_6 = 12,$$

$$\forall x_j \geq 0.$$

Из нее получаем искусственное (недопустимое) базисное решение: $x_3 = 10$, $x_5 = 8$, $x_6 = 12$ (остальные переменные равны нулю). Соответственно базис состоит из одноименных векторов условий.

Далее можно выбрать один из способов решения:

1. решение в один этап;

2. решение в два этапа.

В первом случае критерий модифицируется введением искусственных переменных с большим весом: $L' = 4x_1 - x_2 - M(x_5 + x_6)$. Вместо символа большого числа M можно взять конкретное значение, например положить $M = 100$, что много больше $C_1 = 4$. Далее решение проводится согласно описанному алгоритму.

При использовании второго варианта на первом этапе решается задача минимизации искусственного критерия: $L_u = x_5 + x_6 \rightarrow \min$. Если

оптимальное значение этого критерия окажется отличным от нуля, исходная задача неразрешима из-за противоречивости условий. Нулевое значение L_u^* будет свидетельствовать о достижении допустимого базисного решения, которое принимается за начальное для второго этапа. На нем решается задача по исходному критерию L . При этом в последней таблице первого этапа относительные оценки по критерию L_u заменяются оценками по критерию L . Они вычисляются так же, как в начальной таблице.

4.9.8. Учет двусторонних ограничений

В общем случае на переменные могут накладываться двусторонние ограничения $\alpha_j \leq x_j \leq \beta_j$. Каждое такое ограничение порождает 2 равенства в канонической модели и, следовательно, увеличивает размер симплекс-таблицы на 2 строки. Если сместить начало отсчета на α_j , ограничение примет вид $0 \leq x_j \leq d_j$, где $d_j = \beta_j - \alpha_j$, и таблица будет увеличиваться только на 1 строку. Однако, если такие ограничения накладываются на многие переменные, увеличение размеров симплекс-таблицы будет значительным.

Идея метода с двусторонними ограничениями состоит в учете ограничения сверху аналогично условию $x_j \geq 0$. Как было показано в разд. 4.9.2, выполнение этого условия обеспечивается выбором направляющей строки, т.е. значения вводимой переменной, равного θ_0 . Чтобы переменные в новом базисном решении помимо неотрицательности были не больше d_j , усложним выбор значения вводимой переменной. Предельное значение θ по условию неотрицательности, вычисляемое по формуле (4.9), обозначим θ'_0 , а предельное значение по ограничению сверху – θ''_0 . Из формулы (4.11) следует, что верхнего значения могут достигать только переменные с отрицательными коэффициентами α_{ir} . Приравнявая эти переменные значениям d_j , получаем формулу для вычисления θ''_0 :

$$\theta''_0 = \min \frac{x_i - d_i}{\alpha_{ir}} \quad \text{для } \alpha_{ir} < 0. \quad (4.24)$$

Новое базисное решение будет определяться по формуле (4.13), в которой θ_0 берется из соотношения

$$\theta_0 = \min(\theta'_0, \theta''_0). \quad (4.25)$$

Соответственно и направляющая строка выбирается по θ_0 . В симплекс-таблице вместо одного столбца для θ удобнее иметь два: для θ' и θ'' . Кроме того, добавляется одна строка (сверху), в которой записываются значения небазисных переменных: выводимая из базисного решения переменная x_k равна нулю, если в (4.25) $\theta'_0 < \theta''_0$, и равна d_k в противном случае.

Изменяется также признак оптимальности базисного решения. Условие $\Delta_j \geq 0$ остается в силе только для нулевых небазисных переменных.

К нему добавляется условие для небазисных переменных на верхнем уровне: $\Delta_j \leq 0$. Поэтому в случае неоптимальности текущего решения направляющий столбец выбирается по $\max |\Delta_j|$ из отрицательных для $x_k=0$ и положительных для $x_k=d_k$. Симплекс-преобразование (пересчет таблицы) не изменяется.

4.10. Модифицированный алгоритм

Этот алгоритм отличается от рассмотренного в разд. 4.9.6 тем, что основан на обратной матрице базиса. Для простоты рассмотрим случай с односторонними ограничениями на переменные. Тогда небазисные переменные равны нулю, а система условий задачи принимает вид

$$\mathbf{A}_b \mathbf{X}_b = \mathbf{B}, \quad (4.26)$$

где \mathbf{A}_b – базисная матрица $m \times m$, \mathbf{X}_b – вектор базисных переменных. Так как определитель базисной матрицы не равен нулю, существует обратная матрица \mathbf{A}_b^{-1} . Из (4.26) следует, что базисные переменные можно вычислять по формуле

$$\mathbf{X}_b = \mathbf{A}_b^{-1} \mathbf{B}. \quad (4.27)$$

Теперь покажем, что относительные оценки также можно определять по обратной матрице. Для этого выполним ряд преобразований:

$$\Delta_j = z_j - C_j = \sum_i C_{si} \alpha_{ij} - C_j = \mathbf{C}_b^T \mathbf{a}_j - C_j. \quad (4.28)$$

Вектор \mathbf{a}_j найдем из разложения вектора условий \mathbf{A}_j по базису

$$\begin{aligned} \mathbf{A}_j &= \sum_i A_i \alpha_{ij} = \mathbf{A}_b \mathbf{a}_j, \\ \mathbf{a}_j &= \mathbf{A}_b^{-1} \mathbf{A}_j. \end{aligned} \quad (4.29)$$

Подставляя (4.29) в (4.28), получаем

$$\Delta_j = \mathbf{C}_b^T \mathbf{A}_b^{-1} \mathbf{A}_j - C_j.$$

Произведение $\mathbf{C}_b^T \mathbf{A}_b^{-1}$ не зависит от индекса j , поэтому окончательно будем иметь

$$\Delta_j = \boldsymbol{\pi}^T \mathbf{A}_j - C_j, \quad (4.30)$$

где

$$\boldsymbol{\pi}^T = \mathbf{C}_b^T \mathbf{A}_b^{-1}. \quad (4.31)$$

Таким образом, для решения задачи модифицированным симплекс-методом достаточно вести не всю таблицу, а только обратную матрицу. При единичном начальном базисе обратную матрицу вычислять не надо – она также единичная. Имея обратную матрицу текущего решения, вычисляем сначала вектор $\boldsymbol{\pi}^T$ по формуле (4.31), а затем оценки небазисных переменных по формуле (4.30). Если признак оптимальности

не выполняется, находим минимальную оценку $\Delta_r = \min \Delta_j$. Коэффициенты разложения α_{ir} вектора \mathbf{A}_r по текущему базису находятся по формуле (4.29):

$$\alpha_r = \mathbf{A}_b^{-1} \mathbf{A}_r,$$

где \mathbf{A}_r – вектор условий вводимой переменной x_r , который берется из канонической модели. Столбец α_r добавляем к обратной матрице в качестве направляющего. Далее действуем, как в стандартном методе, то есть для положительных α_{ir} вычисляем θ , находим направляющую строку и направляющий элемент. Затем получаем новую обратную матрицу путем симплекс-преобразования текущей обратной матрицы. После выполнения признака оптимальности решение находится по формуле (4.27).

Очевидно, что преимущество этого метода перед стандартным тем выше, чем больше разница между общим числом переменных и числом базисных переменных канонической модели. Однако обнаружение неразрешимости задачи из-за неограниченности критерия может происходить на более поздних итерациях: только тогда, когда соответствующее условие имеет место в направляющем (добавляемом) столбце.

Новую обратную матрицу можно находить не только симплекс преобразованием старой, но и по формуле

$$\mathbf{A}_{b_нов}^{-1} = \mathbf{E}_k \mathbf{A}_{b_стар}^{-1},$$

где \mathbf{E}_k – почти единичная матрица (только k -й столбец отличается от единичного). Если эту формулу применять на всех итерациях, то для l -й обратной матрицы получим

$$\mathbf{A}_{b_l}^{-1} = \mathbf{E}_l \cdot \mathbf{E}_{l-1} \cdot \dots \cdot \mathbf{E}_1 \cdot \mathbf{E}.$$

Такое представление обратной матрицы называют *мультипликативным*. По сравнению с обычным симплекс-преобразованием оно уменьшает объем вычислений на каждой итерации и тем сильнее, чем меньше плотность матрицы условий.

4.11. Двойственность задач ЛП

Любой задаче ЛП можно поставить в соответствие другую задачу, называемую сопряженной или *двойственной*. При этом исходную задачу называют прямой.

Выделяют общий и симметричный случаи двойственности. Если в прямой задаче все условия представлены в виде неравенств и все переменные ограничены по знаку, то имеет место симметричная пара двойственных задач.

Когда в исходной задаче есть равенства и/или переменные, которые не ограничены по знаку, то говорят об общем случае двойственности (симметрия моделей отсутствует).

4.11.1. Запись двойственной задачи в симметричном случае

Рассмотрим пример. Пусть некоторая фирма способна выпускать 3 вида продукции, используя 4 вида ресурсов.

Известны произведенная стоимость единицы продукции C_j , норма расхода каждого вида ресурса на единицу продукции A_{ij} и количество ресурсов b_i .

Модель прямой задачи, отражающей стремление произвести максимум продукции в стоимостном выражении, очевидна:

$$L = C_1x_1 + C_2x_2 + C_3x_3 \rightarrow \max;$$

$$U_1: A_{11}x_1 + A_{12}x_2 + A_{13}x_3 \leq b_1;$$

$$U_2: A_{21}x_1 + A_{22}x_2 + A_{23}x_3 \leq b_2;$$

$$U_3: A_{31}x_1 + A_{32}x_2 + A_{33}x_3 \leq b_3;$$

$$U_4: A_{41}x_1 + A_{42}x_2 + A_{43}x_3 \leq b_4;$$

$$\forall x_j \geq 0.$$

Она отвечает условиям симметрии, и модель ее двойственной задачи запишется в виде

$$\tilde{L} = b_1U_1 + b_2U_2 + b_3U_3 + b_4U_4 \rightarrow \min;$$

$$A_{11}U_1 + A_{21}U_2 + A_{31}U_3 + A_{41}U_4 \geq C_1;$$

$$A_{12}U_1 + A_{22}U_2 + A_{32}U_3 + A_{42}U_4 \geq C_2;$$

$$A_{13}U_1 + A_{23}U_2 + A_{33}U_3 + A_{43}U_4 \geq C_3;$$

$$\forall U_i \geq 0.$$

Здесь \tilde{L} – критерий двойственной задачи, U_i – переменные двойственной задачи или, просто, двойственные переменные.

Правила, по которым составлена эта модель, включают 5 пунктов:

1. Если в прямой задаче целевая функция максимизируется, то в двойственной минимизируется, и наоборот.
2. Коэффициенты критерия двойственной задачи образуются из компонентов вектора ограничений прямой задачи.
3. Компоненты вектора ограничений двойственной задачи образуются из коэффициентов линейной формы (критерия) прямой задачи.
4. Матрица условий двойственной задачи образуется транспонированием матрицы условий прямой задачи.
5. Знаки неравенств двойственной задачи обратны знакам неравенств прямой.

Для однозначной записи двойственной модели в прямой задаче на максимум все неравенства следует привести к виду “меньше или равно”, а в задаче на минимум – к виду “больше или равно”.

Первые 4 правила действуют как в симметричном, так и в общем случае, а пятое правило – только в случае симметрии.

Как следует из приведенных правил, число условий двойственной задачи равно числу переменных прямой задачи, а число переменных

двойственной задачи равно числу условий прямой. Если для двойственной задачи построить двойственную, то получим прямую.

Характерным примером возникновения симметричной пары двойственных задач является игра двух лиц с нулевой суммой, рассмотренная в разд. 4.2.6.

4.11.2. Интерпретация двойственной задачи

Что отражает двойственная модель? Оказывается, она дает возможность оценить решение исходной (прямой) задачи. В рассматриваемом примере прямая задача состоит, фактически, в наилучшем использовании *всех* имеющихся ресурсов. Каждому варианту плана производства продукции соответствует свое использование ресурсов, а, следовательно, и их полезность или *значимость*. Под последним понимается степень влияния ресурса на результат. Так как каждому условию прямой задачи, отражающему использование ресурса, ставится в соответствие двойственная переменная, то именно она и является мерилем значимости этого ресурса.

Действительно, рассмотрим уравнение размерности условия двойственной задачи

$$[A][U]=[C].$$

Пусть, например, ресурс – фонд времени оборудования (сколько часов оборудование может быть загружено в течение планового периода). Тогда размерность двойственной переменной будет

$$[U] = \frac{[C]}{[a]} = \frac{\frac{\text{руб}}{\text{ед.продукции}}}{\text{час}} = \frac{\text{руб}}{\text{час}}.$$

Итак, U дает стоимость единицы ресурса в единицах критерия, то есть в нашем случае – прирост произведенной стоимости в рублях на каждый дополнительный час работы оборудования. Ниже, в теоремах двойственности, это будет показано строго математически. Поэтому двойственные переменные называют также *теневыми ценами*. Чтобы увидеть отличие теневой цены от рыночной, возьмем конкретные цифры. Пусть рыночная цена некоторого ресурса, полностью используемого в производстве, равна 500 руб/кг и 1 кг достаточно (при наличии других ресурсов) для выпуска дополнительной продукции на сумму 100000 руб. Тогда теневая цена этого ресурса равна 100000 руб. Если поставщик сорвал поставку данного ресурса, то он должен нести ответственность не в размере рыночной цены, а по теневой цене за каждую единицу недопоставленного ресурса. Такое предложение было высказано впервые Л. Канторовичем, который называл двойственные переменные *объективно обусловленными оценками*, сокращенно О.О.О. (объективные цены, складывающиеся в конкретной ситуации производства и потребления).

Таким образом, чем больше абсолютная величина двойственной переменной, тем выше значимость ресурса в полученном решении, и наоборот, более сильному влиянию ресурса на критерий соответствует большее значение двойственной переменной.

Теперь интерпретируем условия двойственной задачи. Если U_i – объективная цена за единицу ресурса, то левая часть неравенства двойственной модели представляет собой *полные затраты* на производство единицы продукции, а все неравенство отражает тот факт, что произведенная стоимость C_i не может превышать суммарных затрат.

Значимость ресурса эквивалентна его дефицитности. Поэтому критерий двойственной задачи можно интерпретировать как суммарную дефицитность ресурсов, которую следует минимизировать.

Другая трактовка заключается в том, что двойственная задача моделирует взаимодействия Покупателя и Продавца ресурсов. Продавец готов продать свои ресурсы, отказавшись от производства продукции, если цены на них (U_i) будут такими, что он получит за ресурсы, расходуемые им на единицу продукции, не меньше C_i , то есть не меньше того, что он имел бы от производства этой продукции. Эти требования выражаются неравенствами двойственной задачи. С другой стороны, Покупатель стремится к таким ценам, которые минимизируют плату за все ресурсы. Это стремление и выражает критерий двойственной задачи.

4.11.3. Запись двойственной задачи в общем случае

Дополнительные правила записи двойственной задачи получим, сводя несимметричные условия прямой задачи к симметричным.

1. Среди условий прямой задачи есть равенство. Пусть таким условием является k -е, а остальные условия записаны как неравенства. Заменив k -е условие-равенство двумя неравенствами

$$a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n = b_k \Leftrightarrow \begin{cases} a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n \leq b_k; \\ -a_{k1}x_1 - a_{k2}x_2 - \dots - a_{kn}x_n \leq -b_k, \end{cases}$$

приходим к симметричному случаю. Если новым неравенствам сопоставить неотрицательные двойственные переменные U'_k и U''_k , то в соответствии с вышеописанными правилами запишем критерий и неравенства двойственной задачи

$$\begin{aligned} \tilde{L} &= b_1U_1 + b_2U_2 + \dots + b_kU'_k - b_kU''_k + \dots + b_mU_m; \\ a_{1j}U_1 + a_{2j}U_2 + \dots + a_{kj}U'_k - a_{kj}U''_k + \dots + a_{mj}U_m &\geq C_j, \quad j = \overline{1, n}. \end{aligned}$$

После вынесения общих множителей за скобки получаем

$$\tilde{L} = b_1 U_1 + b_2 U_2 + \dots + b_k (U'_k - U''_k) + \dots + b_m U_m;$$

$$a_{1j} U_1 + a_{2j} U_2 + \dots + a_{kj} (U'_k - U''_k) + \dots + a_{mj} U_m \geq C_j, \quad j = \overline{1, n}.$$

Так как U'_k и U''_k входят в модель только в виде разности, то можно произвести замену $U_k = U'_k - U''_k$ и, таким образом, иметь одну двойственную переменную, соответствующую равенству прямой задачи, но при этом она не будет ограничена по знаку.

2. Переменная x_k в прямой задаче не ограничена по знаку. Заменяем эту переменную всюду в модели разностью неотрицательных переменных:

$$x_k = x'_k - x''_k.$$

Этим переменным в двойственной задаче будут соответствовать 2 неравенства

$$x'_k : \sum_i a_{ik} U_i \geq C_k;$$

$$x''_k : \sum_i -a_{ik} U_i \geq -C_k \Rightarrow \sum_i a_{ik} U_i \leq C_k,$$

которые эквивалентны равенству

$$\sum_i a_{ik} U_i = C_k.$$

Итак, в общем случае 5-е правило записи двойственной задачи включает 4 пункта, представленные в следующей таблице

Правило	Прямая задача	Двойственная задача
5.1	Переменная $x_j \geq 0$	j -е условие \geq
5.2	Переменная x_j не ограничена по знаку	j -е условие $=$
5.3	i -е условие \leq	Переменная $U_i \geq 0$
5.4	i -е условие $=$	Переменная U_i не ограничена по знаку

Эти правила предполагают, что прямая задача записана с критерием на максимум и неравенствами в виде “меньше или равно”. Очевидно, что в симметричном случае из 5-го правила применяются только пункты 5.1. и 5.3.

Пример 4.4. Прямая задача:

$$L = 2x_1 + x_2 - x_4 + 3x_5 \rightarrow \max;$$

$$5x_1 - 7x_2 + 4x_3 + 2x_5 \leq 8;$$

$$3x_2 + 6x_3 - 2x_4 \geq 10;$$

$$x_1 + 4x_2 + x_3 - 3x_4 = 5;$$

$$9x_1 - x_2 + 5x_4 - 4x_5 \geq 16;$$

$$x_1 \geq 0, x_3 \geq 0, x_4 \geq 0.$$

Перепишем эту модель, изменив знаки 2-го и 4-го неравенств и сопоставив условиям двойственные переменные:

$$\begin{aligned} L &= 2x_1 + x_2 - x_4 + 3x_5 \rightarrow \max; \\ U_1: \quad 5x_1 - 7x_2 + 4x_3 \quad + 2x_5 &\leq 8; \\ U_2: \quad \quad -3x_2 - 6x_3 + 2x_4 &\leq -10; \\ U_3: \quad x_1 + 4x_2 + x_3 - 3x_4 &= 5; \\ U_4: \quad -9x_1 + x_2 \quad - 5x_4 + 4x_5 &\leq -16; \\ x_1 \geq 0, x_3 \geq 0, x_4 \geq 0. \end{aligned}$$

В соответствии с правилами для общего случая записываем модель двойственной задачи

$$\begin{aligned} \tilde{L} &= 8U_1 - 10U_2 + 5U_3 - 16U_4 \rightarrow \min; \\ 5U_1 \quad + U_3 - 9U_4 &\geq 2; \\ -7U_1 - 3U_2 + 4U_3 + U_4 &= 1; \\ 4U_1 - 6U_2 + U_3 &\geq 0; \\ 2U_2 - 3U_3 - 5U_4 &\geq -1; \\ 2U_1 \quad + 4U_4 &= 3; \\ U_1 \geq 0, U_2 \geq 0, U_4 \geq 0. \end{aligned}$$

4.11.4. Теоремы двойственности

Между решениями прямой и двойственной задач существует тесная взаимосвязь, которая устанавливается теоремами двойственности. Эта связь позволяет по решению одной задачи двойственной пары получать решение другой. Основными являются две теоремы, первая из которых определяет связи критериев, а вторая – условий и переменных. Мы сначала рассмотрим составляющие второй теоремы как самостоятельные теоремы, а затем приведем сводную. Аналогично поступим и с первой основной теоремой двойственности.

Теорема 1. Если в оптимальном решении прямой задачи условие выполняется как строгое неравенство

$$\sum_{j=1}^n a_{ij} x_j^* < b_i, \quad (4.32)$$

то соответствующая двойственная переменная равна нулю, то есть

$$U_i^* = 0.$$

Обоснование следует из смысла двойственных переменных. Неравенство (4.32) означает, что i -й ресурс используется не полностью, следовательно, малое изменение этого ресурса не повлияет на результат деятельности (критерий) и поэтому значение двойственной переменной равно нулю.

Следствие. Если дополнительная переменная в i -м условии прямой задачи больше нуля, то соответствующая двойственная переменная равна нулю.

Действительно, в этом случае i -е условие без дополнительной переменной будет заведомо строгим неравенством, что и оговорено в теореме.

Теорема 2. Если в **единственном** оптимальном решении прямой задачи условие выполняется как равенство, то есть

$$\sum_{j=1}^n a_{ij}x_j^* = b_i, \quad (4.33)$$

то соответствующая двойственная переменная будет заведомо не равна нулю.

Равенство (4.33) означает, что i -й ресурс полностью исчерпан, следовательно, малые изменения этого ресурса обязательно приведут к изменению критерия и поэтому его значимость не равна нулю.

Следствие. Если дополнительная переменная в i -м условии равна нулю, то двойственная переменная этого условия не равна нулю.

На рис.4.9 приведена геометрическая интерпретация рассмотренных теорем для случая единственного оптимального решения (вершина A). Здесь допустимое множество D образовано четырьмя условиями-неравенствами с ресурсами b_1, b_2, b_3 и b_4 . В оптимальном решении по 1-му и 2-му ресурсам выполняется равенство и изменение любого из них (показано пунктиром для b_1) приводит к перемещению оптимальной вершины и, следовательно, критерия. Поэтому значимость этих ресурсов или их двойственные переменные отличны от нуля. В то же время по 3-му и 4-му ресурсам имеем строгие неравенства и их изменения не влияют на оптимальное значение критерия, что соответствует нулевым дополнительным переменным.

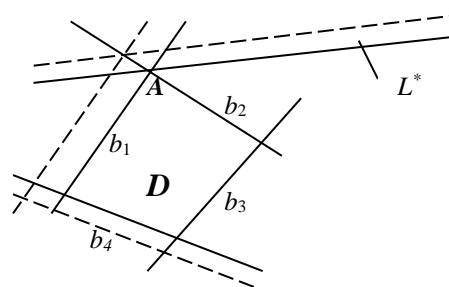


Рис.4.9

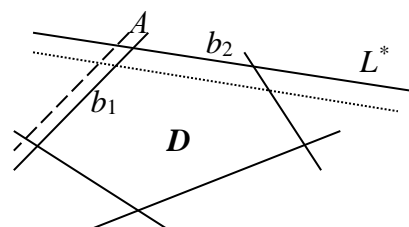


Рис.4.10

Случай с неединственным оптимальным решением показан на рис.4.10. Линия оптимального значения критерия L^* совпадает с границей по 2-му ресурсу. В оптимальном решении, соответствующем вершине A , первые два ресурса используются полностью. Однако изменение b_1 не

приводит к изменению критерия, тогда как любое изменение b_2 отражается на оптимальном значении критерия. Поэтому оценки этих ресурсов разные: $U_1=0, U_2 \neq 0$.

Теоремы 1 и 2 легко трансформируются для двойственной задачи.

Теорема 1'. Если в оптимальном решении двойственной задачи условие выполняется как строгое неравенство

$$\sum_{i=1}^m a_{ij} U_i^* > C_j, \quad (4.34)$$

то соответствующая переменная прямой задачи равна нулю: $x_j^* = 0$.

Интерпретация: если затраты превышают производимую стоимость, то производить такую продукцию невыгодно.

Теорема 2'. Если в **единственном** оптимальном решении двойственной задачи условие выполняется как равенство, то соответствующая переменная прямой задачи строго больше нуля:

$$\sum_{i=1}^m a_{ij} U_i^* = C_j \Rightarrow x_j^* > 0. \quad (4.35)$$

Так как производимая стоимость равна затратам, то производство такой продукции окупается.

Обобщением рассмотренных теорем является **вторая основная теорема двойственности:**

Для того чтобы векторы \mathbf{X}^* и \mathbf{U}^* являлись оптимальными решениями прямой и двойственной задач соответственно, необходимо и достаточно выполнение следующих условий:

$$\begin{cases} x_j^* \left(\sum_i a_{ij} U_i^* - C_j \right) = 0, & j = \overline{1, n}; \\ U_i^* \left(\sum_{j=1}^n a_{ij} x_j^* - b_i \right) = 0, & i = \overline{1, m}. \end{cases} \quad (4.36)$$

Эта теорема учитывает и случай множественности оптимальных решений, когда равенству в одной задаче может соответствовать нулевая переменная в другой.

Теперь покажем на конкретном примере, как приведенные теоремы позволяют находить решение одной из задач двойственной пары по известному решению другой.

Пример 4.5. Рассмотрим задачу, которая решалась ранее графически и симплекс-методом.

Прямая задача (ПЗ)

$$\begin{aligned} L &= 7x_1 + 5x_2 \rightarrow \max, \\ 2x_1 + 3x_2 &\leq 19, \\ 2x_1 + x_2 &\leq 13, \\ 3x_2 &\leq 12, \\ 3x_1 &\leq 17, \\ x_1 &\geq 0, x_2 \geq 0 \end{aligned}$$

Каноническая форма ПЗ

$$\begin{aligned} L &= 7x_1 + 5x_2 \rightarrow \max, \\ 2x_1 + 3x_2 + x_3 &= 19, \\ 2x_1 + x_2 + x_4 &= 13, \\ 3x_2 + x_5 &= 12, \\ 3x_1 + x_6 &= 17, \\ \forall x_j &\geq 0. \end{aligned}$$

Оптимальное решение этой задачи:

$$\begin{aligned} x_1^* &= 5; & x_3^* &= 0; & x_5^* &= 3; & L^* &= 50. \\ x_2^* &= 3; & x_4^* &= 0; & x_6^* &= 2; \end{aligned}$$

Запишем модель двойственной задачи (ДЗ):

$$\begin{aligned} \tilde{L} &= 19U_1 + 13U_2 + 12U_3 + 17U_4 \rightarrow \min; \\ 2U_1 + 2U_2 + 3U_4 &\geq 7; \\ 3U_1 + U_2 + 3U_3 &\geq 5; \\ \forall U_i &\geq 0. \end{aligned}$$

Получим ее решение на основе решения ПЗ и теорем двойственности.

Так как дополнительные переменные x_5 и x_6 , входящие в третье и четвертое условия ПЗ, в оптимальном решении не равны нулю, то согласно следствию теоремы 1

$$U_3^* = U_4^* = 0.$$

Из первой группы условий (4.36) следует, что если исходная переменная ПЗ не равна нулю, то ограничение ДЗ будет выполняться как равенство. Поэтому в нашем примере имеем:

$$\begin{aligned} x_1^* > 0 &\Rightarrow 2U_1^* + 2U_2^* = 7; \\ x_2^* > 0 &\Rightarrow 3U_1^* + U_2^* = 5; \end{aligned}$$

Получили систему 2-х уравнений с двумя неизвестными. Ее решение:

$$\begin{aligned} 4U_1^* &= 7 \Rightarrow U_1^* = \frac{7}{4}; \\ U_2^* &= \frac{11}{4}. \\ \tilde{L}^* &= \frac{57}{4} + \frac{143}{4} = \frac{200}{4} = 50. \end{aligned}$$

Таким образом, мы нашли решение ДЗ без применения симплекс-метода. Как увидим ниже, равенство оптимальных значений критериев ПЗ и ДЗ не случайно. Разумеется, таким способом решать ДЗ нецелесообразно, так как в реальных случаях пришлось бы решать систему уравнений большой размерности. Пример только демонстрирует связь решений двойственной пары задач, а значения двойственных переменных легко получить из оптимальной симплекс-таблицы ПЗ. Они расположены в вспомогательной строке Z в столбцах начального базиса. Обратившись к

симплекс-таблице 3 в разд.4.9.7, легко убедиться в справедливости этого способа нахождения двойственных переменных (см. в столбцах A_3, A_4, A_5 и A_6).

Следующая группа теорем определяет связь между критериями двойственной пары задач.

Теорема 3. Если \mathbf{X} и \mathbf{U} – допустимые решения прямой и двойственной задач соответственно, то

$$L(\mathbf{X}) \leq \tilde{L}(\mathbf{U}). \quad (4.37)$$

Доказательство. Так как допустимость решений означает выполнение неравенств $\sum_j a_{ij}x_j \leq b_i$ в ПЗ и $\sum_i a_{ij}U_i \geq c_j$ в ДЗ, то очевидна цепочка соотношений

$$L(x) = \sum_j c_j x_j \leq \sum_j \sum_i a_{ij} U_i x_j = \sum_i \sum_j a_{ij} x_j U_i \leq \sum_i b_i U_i = \tilde{L}(U),$$

из которой следует справедливость теоремы.

Таким образом, для **любых допустимых** решений значение критерия прямой задачи не может превышать значение критерия двойственной.

Теорема 4. Если \mathbf{X}^* и \mathbf{U}^* - допустимые решения прямой и двойственной задач и $L(\mathbf{X}^*) = \tilde{L}(\mathbf{U}^*)$, то они являются оптимальными решениями двойственной пары задач.

Доказательство. Согласно теореме 3 для любого допустимого \mathbf{X} справедливо неравенство

$$L(\mathbf{X}) \leq \tilde{L}(\mathbf{U}^*).$$

И так как $L(\mathbf{X}^*) = \tilde{L}(\mathbf{U}^*)$ по условию теоремы, то $L(\mathbf{X}) \leq L(\mathbf{X}^*)$. Следовательно, \mathbf{X}^* - оптимальное решение прямой задачи по определению.

Аналогично доказывается оптимальность \mathbf{U}^* для двойственной задачи.

Теорема 5. Для любых оптимальных \mathbf{X}^* и \mathbf{U}^* линейные формы прямой и двойственной задач равны:

$$L(\mathbf{X}^*) = \tilde{L}(\mathbf{U}^*). \quad (4.38)$$

Доказательство. В оптимальных решениях выполняются равенства (4.36). Суммируя первую группу по j , а вторую по i и сделав простые преобразования, получаем

$$\begin{aligned} \sum_j \sum_i a_{ij} U_i^* x_j^* &= \sum_j x_j^* C_j = L(\mathbf{X}^*); \\ \sum_i \sum_j a_{ij} x_j^* U_i^* &= \sum_i b_i U_i^* = \tilde{L}(\mathbf{U}^*). \end{aligned}$$

Из равенства левых частей следует равенство правых и, значит, справедливость теоремы.

Теперь ясно, что совпадение значений критериев в приведенном примере не является случайным.

Теорема позволяет строго объяснить смысл двойственных переменных. Действительно, правомерна запись $L^* = \tilde{L}^* = \sum_i b_i U_i^*$. Отсюда имеем

$$\frac{\partial L^*}{\partial b_i} = U_i^*. \quad (4.39)$$

Таким образом, в оптимальном решении двойственная переменная является производной оптимального значения критерия по правой части ограничения. Значит, как уже говорилось, оптимальная двойственная переменная показывает, как изменится оптимальное значение критерия при изменении ресурса на единицу (она равна этому изменению критерия).

Теорема 6. Если линейная форма одной из задач двойственной пары не ограничена, то условия другой противоречивы. (Обратное не всегда верно, возможна противоречивость в обеих задачах).

Доказательство проведем от противного. Допустим, что при неограниченности $L(\mathbf{x})$ сверху в прямой задаче условия двойственной задачи непротиворечивы. Тогда существует допустимое решение ДЗ, на котором значение ее критерия конечно. Но согласно теореме 3 для допустимых решений должно выполняться неравенство $L(\mathbf{x}) \leq \tilde{L}(\mathbf{U})$, что при принятом допущении невозможно (L бесконечно, а \tilde{L} конечно). Следовательно, ДЗ не может иметь допустимых решений, то есть ее условия противоречивы.

Аналогично доказывается 2-я часть теоремы для случая неограниченности снизу \tilde{L} .

Обобщением теорем 3-5 является ***первая основная теорема двойственности:***

Если одна из задач двойственной пары разрешима, то и другая задача разрешима, при этом оптимальные значения критериев равны; при неразрешимости одной из задач другая тоже неразрешима. ▲

Что дает двойственность для решения задач ЛП помимо анализа? Во-первых, вместо решения исходной задачи можно решать двойственную. Это выгодно, если в ПЗ число условий существенно больше числа переменных (тогда в ДЗ будет меньше ограничений и потребуется меньше итераций). Кроме того, переход к ДЗ может уменьшить число искусственных переменных или исключить их совсем.

Во-вторых, теория двойственности породила такие методы как двойственный симплекс-метод и метод сокращения невязок или венгерский метод. В последнем используются неотрицательные оптимальные решения, при которых не выполняются некоторые

ограничения-равенства (имеются невязки), но от итерации к итерации невязки уменьшаются. Нулевые невязки являются признаком достижения допустимого оптимального решения. Метод применяется в основном для решения транспортных задач.

4.11.5. Двойственный симплекс-метод

Метод был предложен Лемке в 1954 году. Первоначально он представлял собой обычный симплекс-метод, применяемый к двойственной задаче. Позднее метод приобрел самостоятельные черты, и необходимость в переходе к двойственной задаче отпала.

Можно сказать, что прямая задача решается двойственно: в начальном и последующих базисных решениях выполняются условия оптимальности (все оценки неотрицательны при максимизации), но вектор X неположителен, а значит, недопустим. В разрешимой задаче итерации метода приводят к допустимому X , который и будет оптимальным решением задачи.

Поэтому цикл начинается с анализа базисных переменных. Если все переменные неотрицательны, вычисления завершаются. Иначе выбирается направляющая строка k по минимальной базисной переменной. Затем вычисляются значения θ :

$$\theta = \left| \frac{\Delta_j}{\alpha_{kj}} \right| \text{ для } < 0. \quad (4.40)$$

Эта формула получается аналогично выводу в прямом методе (разд.4.9.2), но применительно к двойственной задаче. Если в прямом методе формула следует из требования получения *нового неотрицательного* решения, то здесь – из необходимости соблюдения в новом решении *условия оптимальности*.

При отсутствии в направляющей строке отрицательных α_{kj} констатируется неразрешимость задачи из-за противоречивости условий. Действительно, равенство с отрицательной правой частью и всеми неотрицательными коэффициентами при переменных в левой части не может быть удовлетворено неотрицательными переменными.

Направляющий столбец r определяется по минимальному θ . Далее текущая симплекс-таблица пересчитывается так же, как в прямом методе. В результате получается новое базисное решение, в котором, по крайней мере, x_k станет неотрицательной.

Очевидно, что в разрешимой задаче такой алгоритм приведет к оптимальному решению за конечное число итераций.

Таким образом, двойственный метод отличается от прямого свойствами начального решения и правилами выбора направляющего элемента.

Пример 4.6. Пусть заготовки вырезаются из прямоугольных листов размером 5×10 . Необходимо наилучшим образом выполнить заказ, включающий два вида прямоугольных заготовок: 650 штук размером 2×2.5 и 1300 – размером 3×4 .

В качестве критерия возьмем расход материала (листов), а за переменные x_j примем количество листов, раскраиваемых j -м способом.

Все возможные карты раскроя показаны на рис.4.11. Каждой карте соответствует своя переменная и количество получаемых заготовок (в скобках).

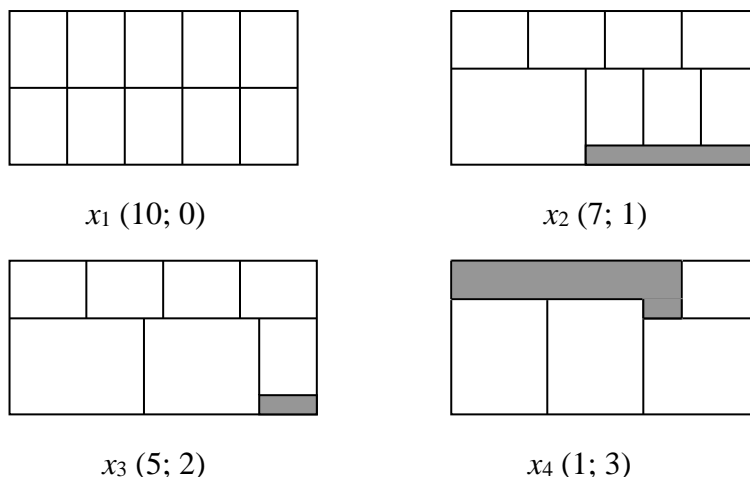


Рис.4.11

Теперь можно записать модель задачи:

$$\begin{aligned}
 L &= x_1 + x_2 + x_3 + x_4 \rightarrow \min \\
 10x_1 + 7x_2 + 5x_3 + x_4 &\geq 650; \\
 x_2 + 2x_3 + 3x_4 &\geq 1300; \\
 \forall x_j &\geq 0.
 \end{aligned}$$

Приведем условия к каноническому виду с помощью дополнительных переменных и затем умножим их на -1 :

$$\begin{aligned}
 -10x_1 - 7x_2 - 5x_3 - x_4 + x_5 &= -650; \\
 -x_2 - 2x_3 - 3x_4 + x_6 &= -1300.
 \end{aligned}$$

В качестве базисных возьмем переменные x_5 и x_6 и заполним начальную симплекс-таблицу:

Таблица 0		0	1	1	1	1	0	0
C_{si}	Базис	A_0	A_1	A_2	A_3	A_4	A_5	A_6
0	x_5	-650	-10	-7	-5	-1	1	0
0	x_6	-1300	0	-1	-2	-3	0	1
L, Δ_j		0	-1	-1	-1	-1	0	0
Z_j		0	0	0	0	0	0	0
θ		--	--	1	1/2	1/3	--	--

Как видно из таблицы, начальное базисное решение является недопустимым (отрицательным), но удовлетворяет условиям оптимальности ($\forall \Delta_j \leq 0$). Поэтому последующие действия будут направлены на достижение допустимого решения при сохранении условий оптимальности.

В качестве направляющей берем строку с минимальной базисной переменной ($x_6 = -1300$). По формуле (4.40) вычисляем значения θ , минимальное из которых определяет направляющий столбец. Тем самым определен и направляющий элемент. Выполнив симплекс-преобразование, получаем новое базисное решение (табл. 1).

Таблица 1		0	1	1	1	1	0	0
C_{si}	Базис	A_0	A_1	A_2	A_3	A_4	A_5	A_6
0	x_5	-650/3	-10	-20/3	-13/3	0	1	-1/3
1	x_4	1300/3	0	1/3	2/3	1	0	-1/3
L, Δ_j		1300/3	-1	-2/3	-1/3	0	0	-1/3
θ		--	1/10	1/10	1/13	--	--	1

Так как в этом решении есть отрицательная переменная, проводим следующую итерацию, результаты которой представлены в табл. 2.

Таблица 2		0	1	1	1	1	0	0
C_{si}	Базис	A_0	A_1	A_2	A_3	A_4	A_5	A_6
1	x_3	50	30/13	20/13	1	0	-3/13	1/13
1	x_4	400	-20/13	-9/13	0	1	2/13	-5/13
L, Δ_j		450	-3/13	-2/13	0	0	-1/13	-4/13

Здесь базисные переменные положительны, значит, решение допустимое. Условия оптимальности, как и в предыдущих решениях, выполняются. Таким образом, в табл. 2 имеем оптимальное решение задачи раскроя: $x_1^* = x_2^* = 0$, $x_3^* = 50$, $x_4^* = 400$, $L^* = 450$.

4.12. Параметрический анализ

Для практического внедрения результатов оптимизации полученное решение должно быть дополнено всесторонним анализом, который позволяет предсказать поведение оптимального решения при тех или иных возможных изменениях в модели.

Анализ чувствительности по переменным проводится по относительным оценкам. Оценка показывает скорость ухудшения значения

критерия при отклонении значения переменной от оптимального. Очевидно, что точность реализации оптимальных значений должна быть тем выше, чем больше по абсолютной величине относительная оценка.

Влияние изменения отдельного ресурса в окрестности исходного значения устанавливается по двойственным переменным в оптимальном решении.

Вариантный анализ заключается в исследовании поведения оптимального решения при изменении состава модели (смена критерия, снятие или добавление одного ограничения и т.п.).

Параметрический анализ (параметрическое программирование) применяется для определения изменения оптимального решения в общем случае при одновременном и непрерывном изменении нескольких коэффициентов модели (в частном случае может изменяться только один). При этом характер изменения коэффициентов задается параметрически как функция одного параметра, а интересующий диапазон изменения значительно шире окрестности исходных значений. В параметрическом программировании рассматривается параметрирование правой части, коэффициентов линейной формы, совместно коэффициентов критерия и правых частей, параметрирование отдельных столбцов или строк матрицы условий и другие более общие случаи совместного изменения коэффициентов.

Наибольший интерес представляют первые две задачи параметрирования, так как на практике нередко ситуации с изменением ресурсов (правых частей), цен или удельных затрат (коэффициентов критерия). Для них параметрическое решение можно найти в общем виде при линейной зависимости изменений от параметра, а в отдельных случаях – и при нелинейной.

4.12.1. Параметрирование вектора ограничений

Пусть оптимальное решение X^* получено для вектора $\mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix}$.

Поставим вопрос: как будет изменяться оптимальное решение при изменении правой части, заданном параметрически $\mathbf{B}(\lambda)$?

Рассмотрим только случай линейной зависимости

$$\mathbf{B}(\lambda) = \mathbf{B} + \lambda \mathbf{V}, \quad (4.41)$$

где $\lambda \geq 0$ – параметр, определяющий величину изменения вектора ограничений;

\mathbf{V} – вектор размерности m , определяющий направление и относительную скорость изменения компонентов вектора ограничений.

Этот вектор задается ЛПР исходя из прогноза возможных изменений ресурсов.

Например, для трехмерного вектора \mathbf{V} изменения могут быть заданы в виде

$$\mathbf{V} = \begin{bmatrix} -3 \\ 1,5 \\ -1 \end{bmatrix},$$

то есть ожидается одновременное уменьшение первого и третьего ресурсов и увеличение второго ресурса. При этом абсолютная величина изменения первого ресурса в три раза, а второго в полтора раза больше, чем третьего.

Для *любого* базисного решения условия задачи

$$\mathbf{AX}=\mathbf{V}$$

можно записать в виде

$$\mathbf{A}_B\mathbf{X}_B+\mathbf{A}_N\mathbf{X}_N=\mathbf{V},$$

где индексы “В” и “Н” обозначают базисные и небазисные векторы (матрицы). Так как небазисные переменные равны нулю, то отсюда следует

$$\mathbf{A}_B\mathbf{X}_B=\mathbf{V}$$

и, в частности, для оптимального решения

$$\mathbf{A}_B^*\mathbf{X}_B^*=\mathbf{V}. \quad (4.42)$$

Так как мы исходим из наличия решения \mathbf{X}^* , то базисная матрица \mathbf{A}_B^* - неособенная и существует обратная к ней матрица \mathbf{A}_B^{*-1} , умножая на которую слева равенство (4.42), получаем.

$$\mathbf{X}_B^* = \mathbf{A}_B^{*-1}\mathbf{V}. \quad (4.43)$$

Очевидно, что если заменить в (4.42) \mathbf{V} на $\mathbf{V}(\lambda)$ при $\lambda=0$, то ничего не изменится. При невырожденном оптимальном решении малое изменение \mathbf{V} ($\lambda>0$ мало) не изменяет базис: оптимальная вершина хотя и смещается, но образуется теми же ограничениями. Поэтому в данном случае изменяется только оптимальное решение. Оптимальное решение при $\lambda>0$ обозначим \mathbf{X}^{**} . Тогда для малых λ равенство (4.42) запишется в виде

$$\mathbf{A}_B^*\mathbf{X}_B^{**} = \mathbf{V} + \lambda\mathbf{V},$$

откуда находим изменяемое оптимальное решение

$$\mathbf{X}_B^{**} = \mathbf{A}_B^{*-1}\mathbf{V} + \lambda\mathbf{A}_B^{*-1}\mathbf{V}.$$

С учетом (4.43) окончательно имеем

$$\mathbf{X}_B^{**} = \mathbf{X}_B^* + \lambda\mathbf{P}, \quad (4.44)$$

где

$$\mathbf{P} = \mathbf{A}_B^{*-1}\mathbf{V}. \quad (4.45)$$

Таким образом, при линейном характере изменений ресурсов оптимальные значения переменных также меняются линейно. Однако это справедливо до тех пор, пока не происходит смена базиса. В невырожденном решении всегда найдется $\lambda>0$, при котором базис не

меняется. Из выражения (4.44) следует, что при неотрицательном векторе \mathbf{P} возрастание λ не может привести к уменьшению какой-либо базисной переменной и, значит, к смене базиса. В этом случае формула (4.44) справедлива для любых $\lambda > 0$. Такая ситуация показана на рис. 4.12, где изменение b_1 и b_2 в направлении стрелок не приводит к смене базиса (вершины, в которой достигается оптимальное решение).

Если же среди компонент вектора \mathbf{P} есть отрицательные, то соответствующие базисные переменные с увеличением λ будут уменьшаться. Если хотя бы одна из переменных обратится в нуль, то произойдет смена базиса и, следовательно, изменится обратная матрица. Формула (4.44) с исходными базисным решением и вектором \mathbf{P} становится несправедливой. Этот случай иллюстрируется рис. 4.13., где оптимальная вершина сначала образована ограничениями по b_1 и b_2 , а затем – ограничениями по b_1 и b_3 .

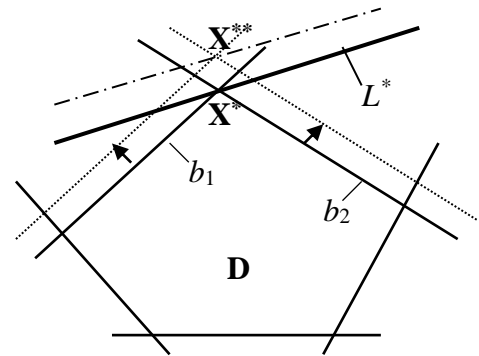


Рис. 4.12

Значение $\hat{\lambda}$, при котором происходит смена базиса (базисного решения), называется критическим. Оно определяется по очевидной формуле

$$\hat{\lambda} = \min_{p_i < 0} \frac{x_i^*}{|p_i|}, \quad (4.46)$$

где p_i – компоненты вектора \mathbf{P} .

Таким образом, исходное решение можно использовать для определения изменяемых решений по формуле (4.44) только в диапазоне

$$0 \leq \lambda < \hat{\lambda}.$$

Отсюда получаем максимальное изменение правой части

$$\Delta \mathbf{B}_{\max} = \hat{\lambda} \mathbf{V}.$$

Если диапазон изменения правой части недостаточен, то для его расширения необходимо заново решить задачу с вектором $\mathbf{B}_1 = \mathbf{B} + \Delta \mathbf{B}_{\max}$. Тогда получим новое оптимальное решение, новую обратную матрицу и на их основе снова проводится параметрирование для $\mathbf{B}(\lambda) = \mathbf{B}_1 + \lambda_1 \mathbf{V}$. Повторяя эти действия, можно охватить весь желаемый диапазон изменения ресурсов. При этом соотношение компонент (но не знаков!) в векторе \mathbf{V} может остаться исходным или

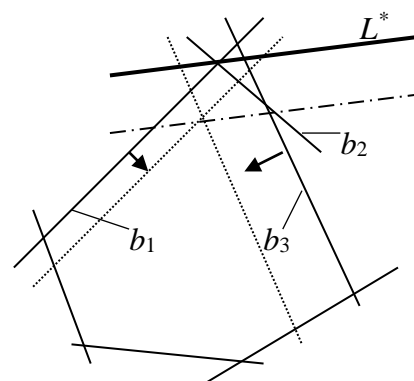


Рис. 4.13

измениться. В последнем случае зависимость от параметра λ на всем исследованном диапазоне будет кусочно-линейной.

Пример 4.7. Параметризуем задачу, решенную симплекс-методом в разд. 4.9.7 (пример 4.2) в предположении изменения 1-го и 4-го ресурсов.

Анализ поступления этих ресурсов показал, что первый может возрастать, а четвертый – уменьшаться, причем изменение четвертого может быть по абсолютной величине в два раза больше, чем первого. На этом основании записываем вектор изменений $\mathbf{V}=(1,0,0,-2)$.

Взяв обратную матрицу из оптимальной симплекс-таблицы (в столбцах начального базиса), по формуле (4.45) вычисляем вектор \mathbf{P} :

$$P = \underbrace{\begin{bmatrix} A_3 & A_4 & A_5 & A_6 \\ 3/4 & -9/4 & 0 & 1 \\ 1/2 & -1/2 & 0 & 0 \\ -3/2 & 3/2 & 1 & 0 \\ -1/4 & 3/4 & 0 & 0 \end{bmatrix}}_{A_B^{*-1}} \cdot \begin{bmatrix} V \\ 1 \\ 0 \\ 0 \\ -2 \end{bmatrix} = \begin{bmatrix} -5/4 \\ 1/2 \\ -3/2 \\ -1/4 \end{bmatrix}$$

Выписываем исходное оптимальное решение, соблюдая порядок базисных переменных в последней таблице:

$$x_6^* = 2;$$

$$x_2^* = 3;$$

$$x_5^* = 3;$$

$$x_1^* = 5.$$

Из данного порядка следует, что первый компонент вектора \mathbf{P} соответствует 6-й переменной, а последний – первой. Таким образом, параметрическое решение запишется в виде

$$x_1^{**} = 5 - 1/4\lambda;$$

$$x_2^{**} = 3 + 1/2\lambda;$$

$$x_3^{**} = x_4^{**} = 0;$$

$$x_5^{**} = 3 - 3/2\lambda;$$

$$x_6^{**} = 2 - 5/4\lambda;$$

$$L^{**} = 50 + 3/4\lambda.$$

Так как вектор \mathbf{P} имеет отрицательные компоненты, вычисляем критическое значение λ

$$\hat{\lambda} = \min \left[\frac{5}{1/4}; \frac{3}{3/2}; \frac{2}{5/4} \right] = \frac{8}{5} = 1,6.$$

Оно позволяет определить критические отклонения ресурсов от исходных значений:

$$\Delta \hat{b}_1 = v_1 \hat{\lambda} = 1 \cdot 1,6 = 1,6; \quad \Delta \hat{b}_4 = v_4 \hat{\lambda} = -2 \cdot 1,6 = -3,2.$$

Следовательно, полученное параметрическое решение будет справедливо при одновременном изменении ресурсов в диапазонах

$$19 \leq b_1 < 20,6;$$

$$13,8 < b_4 \leq 17.$$

Чтобы расширить эти диапазоны, в задаче нужно заменить вектор $\mathbf{B}=(19; 13; 12;17)$ вектором $\mathbf{B}_1=(20,6;13;12;13,8)$ и снова решить ее. Новое решение параметрируется аналогичным образом.

Замечания. 1) Вместо принятого в примере вектора \mathbf{V} можно брать $k\mathbf{V}$, где k – любое положительное число. При этом будет изменяться в k раз только $\hat{\lambda}$, а диапазоны изменения b_i останутся прежними. 2) Очевидно, что если изменяется правая часть только в одном условии и в векторе \mathbf{V} соответствующий компонент взят равным единице, то коэффициент при λ в параметрической записи L^{**} должен равняться двойственной переменной. При этом параметрический анализ позволяет определить диапазон изменения ресурса, в котором это значение двойственной переменной не меняется.

4.12.2. Параметрирование коэффициентов линейной формы

Здесь рассмотрим три варианта параметрирования, отличающихся своими возможностями.

1. Коэффициенты критерия изменяются линейно от параметра:

$$\mathbf{C}(\lambda)=\mathbf{C}+\lambda\mathbf{V},$$

а вектор \mathbf{V} задается аналогично случаю изменения ресурсов.

Тогда задача параметрирования имеет вид:

$$(\mathbf{C}+\lambda\mathbf{V})^T\mathbf{X} \rightarrow \max$$

$$\mathbf{A}\mathbf{X} \leq \mathbf{B}$$

$$\mathbf{X} \geq 0.$$

Запишем соответствующую двойственную задачу:

$$\mathbf{B}^T\mathbf{U} \rightarrow \min$$

$$\mathbf{A}^T\mathbf{U} \geq \mathbf{C}+\lambda\mathbf{V}$$

$$\mathbf{U} \geq 0$$

Очевидно, что она представляет собой задачу параметрирования вектора ограничений, решение которой может быть получено вышеописанным методом. В результате найдем диапазон изменения

параметра λ ($0 \leq \lambda < \hat{\lambda}$), в котором базис двойственной задачи остается неизменным. В строке Z оптимальной таблицы двойственной задачи находятся переменные прямой задачи (двойственные к двойственной). Но значения z_j зависят только от базиса, поэтому в найденном диапазоне λ оптимальное решение также не меняется. Изменяться будет только критерий. При достижении критического значения λ произойдет смена базиса (оптимальной вершины), а значит, и оптимального решения прямой задачи. Проследить дальнейшее изменение решения можно после повторного решения двойственной задачи с вектором $C + \hat{\lambda}V$.

Такое поведение следует и из геометрических представлений (рис. 4.14). Изменение коэффициентов линейной формы изменяет наклон

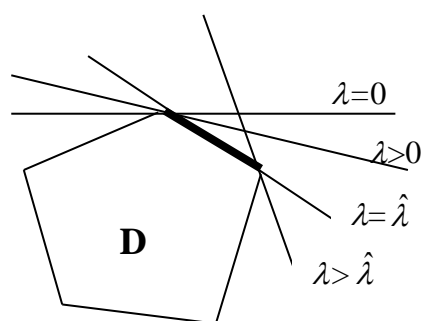


Рис. 4.14

линии уровня критерия, но не влияет на допустимое множество. При наличии критических значений λ изменение коэффициентов приводит к скачкообразному изменению оптимального решения – переходу из вершины в вершину (смежную).

2. Для небазисных переменных весьма просто можно определить диапазон изменения C_j , в котором

оптимальное решение остается неизменным.

Действительно, пока при изменения C_j все $\Delta_j \geq 0$ оптимальное решение исходной задачи сохраняет свой статус. Так как

$$\Delta_j = Z_j - C_j,$$

то уменьшение C_j не может изменить знак оценки. Поэтому интерес представляет увеличение C_j . Пусть $C'_j = C_j + \varepsilon_j$, $\varepsilon_j \geq 0$. Тогда

$$\Delta'_j = Z_j - C_j - \varepsilon_j = \Delta_j - \varepsilon_j \geq 0.$$

Отсюда следует, что при $\varepsilon_j \leq \Delta_j$ исходное решение остается оптимальным.

3. Этот вариант основан на формуле вычисления относительных оценок в модифицированном симплекс-методе:

$$\Delta_j = \underbrace{C_B^T A_B^{-1}}_{\Pi} A_j - C_j = \Pi^T A_j - C_j.$$

Она позволяет исследовать влияние изменения любых коэффициентов C_j . В общем случае эти коэффициенты являются некоторыми функциями параметра λ : $C_j(\lambda)$. Тогда условия оптимальности запишутся в виде

$$C_B^T(\lambda) A_B^{-1} A_j - C_j(\lambda) \geq 0, \quad j \in \text{небаз.}$$

Здесь обратная матрица соответствует оптимальному базису. Пока при изменении коэффициентов (т.е. λ) эти неравенства выполняются, оптимальное решение не изменяется. Значение λ , при котором хотя бы

одно из условий становится равенством, и будет критическим. Практически оно находится так: каждое условие записывается как равенство и определяются его корни; из всех корней выбирается наименьшее положительное. Это и будет $\hat{\lambda}$.

Очевидно, что данный вариант параметрирования пригоден как для линейных, так и нелинейных зависимостей от параметра. Однако в последнем случае его применение ограничено возможностью нахождения корней нелинейного уравнения.

Пример 4.8. Пусть ожидается изменение коэффициентов критерия в примере 4.2 (разд. 4.9.7) по закону: $C_1(\lambda)=7-2\lambda$, $C_2(\lambda)=5+\lambda$. Необходимо определить критическое значение λ , если таковое имеется.

В оптимальной симплекс-таблице базисные индексы расположены в следующем порядке: 6, 2, 5, 1. Значит, $C_B^T(\lambda)=[0; 5+\lambda; 0; 7-2\lambda]$. Вычисляем:

$$\Pi^T(\lambda) = C_B^T A_B^{-1} = [0, 5+\lambda, 0, 7-2\lambda] \begin{bmatrix} 3/4 & -9/4 & 0 & 1 \\ 1/2 & -1/2 & 0 & 0 \\ -3/2 & 3/2 & 0 & 0 \\ -1/4 & 3/4 & 1 & 0 \end{bmatrix} = [3/4 + \lambda; 11/4 - 2\lambda; 7 - 2\lambda; 0]$$

Из условий оптимальности $\Delta_3 \geq 0$, $\Delta_4 \geq 0$ записываем уравнения

$$\begin{aligned} \Pi^T(\lambda) A_3 - C_3 &= [3/4 + \lambda; 11/4 - 2\lambda; 7 - 2\lambda; 0] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 3/4 + \lambda = 0; \\ \Pi^T(\lambda) A_4 - C_4 &= [3/4 + \lambda; 11/4 - 2\lambda; 7 - 2\lambda; 0] \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = 11/4 - 2\lambda = 0. \end{aligned}$$

Первое уравнение имеет отрицательный корень, корень второго равен $11/8$. Таким образом, $\hat{\lambda}=11/8$. До этого значения λ оптимальное решение не изменяется, при $\lambda=11/8$ имеем альтернативные оптимальные решения (линии уровня $L(\hat{\lambda})=34/8x_1 + 51/8x_2=Const$ параллельны границе $2x_1 + 3x_2=19$), а при $\lambda>11/8$ оптимальное решение переместится в вершину В (рис. 4.3). ▲

Как отмечалось выше, параметрические решения могут быть получены также при одновременном изменении правых частей и коэффициентов критерия по линейной зависимости от одного параметра

$$\begin{cases} B + \lambda V_B \\ C + \lambda V_C, \end{cases}$$

при линейном изменении столбца условий $A_j + \lambda V_j$ или строки $a_i + \lambda V_i$.

В других случаях изменения модели поведение оптимального решения определяется решениями задачи одним из методов ЛП при разных значениях изменяемых параметров модели.

4.13. Задания для самостоятельной работы

1. Решить симплекс-методом и графически следующие задачи.

№1	$L = 2x_1 - 4x_2 \rightarrow \min$ $8x_1 - 5x_2 \leq 16$ $x_1 + 3x_2 \geq 2$ $2x_1 + 7x_2 \leq 9$ $\forall x_j \geq 0$	№2	$L = x_1 + x_2 \rightarrow \max$ $x_1 + 2x_2 \leq 14$ $4x_1 + 6x_2 \geq 24$ $-5x_1 + 3x_2 \leq 15$ $\forall x_j \geq 0$
№3	$L = -x_1 - x_2 \rightarrow \min$ $2x_1 + 3x_2 \geq 6$ $4x_1 + 2x_2 \leq 40$ $-3x_1 + 5x_2 \leq 30$ $x_1, x_2 \geq 0$	№4	$L = 8x_1 - 2x_2 \rightarrow \min$ $3x_1 - x_2 \geq 4$ $4x_1 - 2x_2 \geq 5$ $8x_1 - x_2 \leq 15$ $\forall x_j \geq 0$
№5	$L = 4x_1 + 6x_2 \rightarrow \max$ $2x_1 + 3x_2 \geq 6$ $4x_1 + 2x_2 \leq 40$ $-3x_1 + 5x_2 \leq 30$ $x_1, x_2 \geq 0$	№6	$L = 2x_1 - 4x_2 \rightarrow \max$ $8x_1 - 5x_2 \geq 16$ $2x_1 + x_2 \geq 2$ $2x_1 + 7x_2 \leq 9$ $\forall x_j \geq 0$
№7	$L = x_1 + 2x_2 \rightarrow \max$ $x_1 + x_2 \leq 6$ $3x_1 + 10x_2 \leq 26$ $4x_1 + 2x_2 \geq 7$ $\forall x_j \geq 0$	№8	$L = 2x_1 + 3x_2 \rightarrow \max$ $2x_1 + x_2 \leq 10$ $-2x_1 + 3x_2 \leq 6$ $2x_1 + 4x_2 \geq 8$ $\forall x_j \geq 0$
№9	$L = x_1 + 2x_2 \rightarrow \max$ $4x_1 - 2x_2 \leq 12$ $2x_1 + 4x_2 \geq 16$ $-x_1 + 3x_2 \leq 6$ $\forall x_j \geq 0$	№10	$L = 2x_1 + x_2 \rightarrow \max$ $20x_1 + 10x_2 \geq 75$ $12x_1 + 7x_2 \leq 55$ $25x_1 + 10x_2 \leq 90$ $\forall x_j \geq 0$
№11	$L = 5x_1 + 3x_2 \rightarrow \max$ $3x_1 + 5x_2 \leq 15$ $x_1 + x_2 \geq 2$ $5x_1 + 2x_2 \leq 10$ $\forall x_j \geq 0$	№12	$L = 2x_1 + 3x_2 \rightarrow \min$ $x_1 + x_2 \leq 4$ $6x_1 + 2x_2 \geq 8$ $x_1 + 5x_2 \geq 4$ $\forall x_j \geq 0$
№13	$L = -2x_1 + x_2 \rightarrow \min$ $3x_1 - 2x_2 \leq 12$ $-x_1 + 2x_2 \leq 8$ $2x_1 + 3x_2 \geq 6$ $\forall x_j \geq 0$	№14	$L = 2x_1 + 3x_2 \rightarrow \min$ $3x_1 + 2x_2 \geq 6$ $x_1 + 4x_2 \geq 4$ $x_1 + x_2 \leq 3$ $\forall x_j \geq 0$
№15	$L = x_1 + 2x_2 - x_3 \rightarrow \max$ $-x_1 + 4x_2 - 2x_3 \leq 12$ $x_1 + x_2 + 2x_3 \leq 17$ $2x_1 - x_2 + 2x_3 = 4$ $\forall x_j \geq 0$	№16	$L = x_1 + x_2 \rightarrow \max$ $2x_1 + 4x_2 \leq 16$ $-4x_1 + 2x_2 \leq 8$ $x_1 + 3x_2 \geq 9$ $\forall x_j \geq 0$

№17	$L = 2x_1 + 3x_2 \rightarrow \max$ $2x_1 + x_2 \leq 10$ $2x_1 + 4x_2 \geq 8$ $-2x_1 + 3x_2 \leq 6$ $\forall x_j \geq 0$	№18	$L = 3x_1 + x_2 \rightarrow \max$ $x_1 + x_2 \geq 5$ $2x_1 + 3x_2 \leq 21$ $7x_1 + x_2 \leq 35$ $\forall x_j \geq 0$
№19	$L = x_1 + x_2 \rightarrow \max$ $x_1 + 2x_2 \leq 14$ $2x_1 + 3x_2 \geq 12$ $-5x_1 + 3x_2 \leq 15$ $\forall x_j \geq 0$	№20	$L = 5x_1 - 2x_2 \rightarrow \min$ $3x_1 + x_2 \geq 1$ $-x_1 + x_2 \leq 25$ $7x_1 - 2x_2 \geq 8$ $\forall x_j \geq 0$
№21	$L = 2x_1 + 4x_2 \rightarrow \max$ $4x_1 - 2x_2 \leq 12$ $2x_1 + 4x_2 \geq 16$ $-2x_1 + 6x_2 \leq 12$ $\forall x_j \geq 0$	№22	$L = 2x_1 - 4x_2 \rightarrow \min$ $8x_1 - 5x_2 \leq 16$ $x_1 + 3x_2 \geq 2$ $2x_1 + 7x_2 \leq 8$ $\forall x_j \geq 0$
№23	$L = x_1 + 2x_2 - x_3 \rightarrow \max$ $-x_1 + 4x_2 - 2x_3 \leq 6$ $x_1 + x_2 + 2x_3 \geq 6$ $2x_1 - x_2 + 2x_3 = 4$ $\forall x_j \geq 0$	№24	$L = 9x_1 + 5x_2 \rightarrow \max$ $3x_1 - 6x_2 \geq 1$ $5x_1 + 2x_2 \leq 28$ $x_1 + 7x_2 \leq 42$ $\forall x_j \geq 0$
№25	$L = x_1 + 0,5x_2 \rightarrow \max$ $2x_1 + x_2 \geq 3$ $5x_1 + 2x_2 \leq 7$ $-3x_1 + 5x_2 \leq 10$ $\forall x_j \geq 0$	№26	$L = -x_1 - 0,5x_2 \rightarrow \max$ $2x_1 + x_2 \geq 1$ $x_1 + 2x_2 \geq 7$ $-3x_1 + 11x_2 \leq 30$ $\forall x_j \geq 0$

2. Задачи из п.1 решить модифицированным симплекс-методом.
 3. По решению прямой задачи (п.1) найти решение двойственной задачи с использованием теорем двойственности.
 4. Выполнить параметрический анализ задач из п.1 для случаев:
 - 4.1. увеличения b_1 ;
 - 4.2. уменьшения b_2 ;
 - 4.3. уменьшения b_3 ;
 - 4.4. одновременного уменьшения b_1 и увеличения b_2 и b_3 ; изменение b_1 в два, а b_3 в три раза больше изменения b_2 ;
 - 4.5. одновременного изменения коэффициентов критерия по закону $C_1(\lambda) = C_1 - 0.2C_1\lambda$, $C_2(\lambda) = C_2 + 0.1C_2\lambda$.
- Результаты пп. 3 и 4 сопоставить со значениями в строках Z_j и Δ_j оптимальной симплекс-таблицы, полученной при выполнении п.1.

Глава 5. ТРАНСПОРТНЫЕ ЗАДАЧИ

Задачи, называемые транспортными, составляют большой подкласс распределительных задач. С содержательной стороны они не обязательно связаны с доставкой или перевозкой грузов, а выделяются из других задач особой структурой математической модели. Поэтому правильнее говорить о моделях транспортного типа.

Если удельные затраты на перевозку не зависят от количества перевозимого груза, транспортная задача описывается линейной моделью. При этом ее особенности позволяют применять специальные методы линейного программирования, которые более эффективны, чем универсальные. Ниже рассматриваются только линейные задачи.

5.1. Основные модели транспортных задач

5.1.1. Простейшая транспортная задача (Т-задача)

Эта задача является основополагающей для всех транспортных задач. Пример такой задачи приведен в разд. 4.9.

В общем случае исходными данными являются:

m – число пунктов отправления (ПО) или производства;

n – число пунктов назначения (ПН) или потребления;

C_{ij} – затраты на перевозку единицы груза из пункта i в пункт j , $\forall ij$;

a_i – количество груза в пункте i , $\forall i$ (возможности ПО);

b_j – потребность в грузе в пункте j , $\forall j$.

Критерием задачи являются суммарные затраты на перевозку. Безотносительно к значениям a_i и b_j модель записывается в виде

$$\text{Критерий} \quad L = \sum_{i=1}^m \sum_{j=1}^n C_{ij} X_{ij} \rightarrow \min;$$

$$\text{Условия по ПО:} \quad \sum_{j=1}^n X_{ij} \leq a_i, \quad i = \overline{1, m};$$

$$\text{Условия по ПН:} \quad \sum_{i=1}^m X_{ij} \geq b_j, \quad j = \overline{1, n};$$

$$\forall X_{ij} \geq 0.$$

Однако такая запись модели корректна только тогда, когда $\sum_i a_i \geq \sum_j b_j$.

Напомним, что задача, в которой суммарные потребности равны суммарной возможности, то есть

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j, \quad (5.1)$$

называется сбалансированной или закрытой. Как будет показано в этой главе, любая несбалансированная задача легко приводится к закрытой. Поэтому здесь рассмотрим только сбалансированную задачу:

$$L = \sum_i \sum_j C_{ij} X_{ij} \rightarrow \min; \quad (5.2)$$

$$\sum_{j=1}^n X_{ij} = a_i, \quad i = \overline{1, m}; \quad (5.3)$$

$$\sum_{i=1}^m X_{ij} = b_j, \quad j = \overline{1, n}; \quad (5.4)$$

$$\forall X_{ij} \geq 0. \quad (5.5)$$

Элементы модели:

$$\mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1n} \\ X_{21} & X_{22} & \dots & X_{2n} \\ \dots & \dots & \dots & \dots \\ X_{m1} & X_{m2} & \dots & X_{mn} \end{bmatrix} \text{ – матрица перевозок;}$$

$$\mathbf{C} = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \dots & \dots & \dots & \dots \\ C_{m1} & C_{m2} & \dots & C_{mn} \end{bmatrix} \text{ – матрица транспортных затрат;}$$

$\mathbf{a} = (a_1, a_2, \dots, a_m)$ – вектор возможностей ПО;

$\mathbf{b} = (b_1, b_2, \dots, b_n)$ – вектор потребностей ПН.

Отметим особенности рассматриваемой задачи:

- ♦ Модель содержит две группы условий, размерность которых равна соответствующему числу ПО и ПН; число переменных равно произведению $m \times n$;
- ♦ Все коэффициенты при переменных в условиях (5.3), (5.4) равны единице;
- ♦ Каждая переменная входит в условия ровно 2 раза, один и только один раз в группу (5.3) и также один раз в группу (5.4);
- ♦ Задача имеет простые условия разрешимости, которые определяются следующей теоремой.

Теорема.

Для разрешимости Т-задачи необходимо и достаточно, чтобы она была сбалансированной.

Замечание. Теорема справедлива при конечных значениях C_{ij} .

Приведем доказательство.

Необходимость доказывается исходя из того, что задача (5.2)-(5.5) разрешима. В этом случае все условия задачи выполняются. Просуммируем условия (5.3) по i , а условия (5.4) по j :

$$\sum_i \sum_j X_{ij} = \sum_i a_i;$$

$$\sum_j \sum_i X_{ij} = \sum_j b_j.$$

Так как левые части равенств равны, то равны и правые. Таким образом, в разрешимой задаче всегда имеет место формальный баланс возможностей и потребностей.

Достаточность доказывается конструктивным способом.

Вспомним, что задача линейного программирования всегда разрешима, если допустимое множество – выпуклый многогранник, то есть непустое и ограниченное.

Ограниченность переменных снизу задана явно, а ограничение сверху следует из конечности всех a_i и b_j , больше которых переменные быть не могут. Следовательно, множество ограничено.

Теперь покажем, что оно непустое. Для этого достаточно найти хотя бы одно допустимое решение. Одно из таких решений всегда можно построить, если задача сбалансирована, следующим образом:

$$X_{ij} = \frac{a_i b_j}{\sum_i a_i} = \frac{a_i b_j}{\sum_j b_j}, \quad \forall ij. \quad (5.6)$$

Очевидно, что оно неотрицательно. Остается проверить выполнение основных условий задачи. Подставив (5.6) в левую часть (5.3), получим:

$$\sum_j \frac{a_i b_j}{\sum_j b_j} = \frac{a_i \sum_j b_j}{\sum_j b_j} = a_i \Rightarrow \text{решение удовлетворяет условиям (5.3).}$$

Подставив первый вариант (5.6) в (5.4), также убеждаемся в выполнении этих условий:

$$\sum_i \frac{a_i b_j}{\sum_i a_i} = \frac{b_j \sum_i a_i}{\sum_i a_i} = b_j.$$

Таким образом, допустимое множество сбалансированной задачи непустое и ограниченное, а, значит, задача всегда разрешима. ▲

Условия (5.3), (5.4) – линейно зависимы из-за сбалансированности задачи. Действительно, пусть известны все равенства (5.3) и $(n-1)$ равенство (5.4). Просуммируем отдельно первые и вторые и затем из первой суммы вычтем вторую. В результате получим недостающее равенство, описывающее пункт потребления, не включенный в исходную систему (5.4). Можно строго показать, что число линейно-независимых уравнений или, иначе, ранг системы (5.3), (5.4) равен $m+n-1$. Следовательно, такую размерность имеют базис и базисное решение Т-задачи.

5.1.2. Транспортная задача с ограниченными пропускными способностями (T_d -задача)

Отличается от предыдущей задачи учетом ограничений на пропускные возможности коммуникаций. В реальных условиях пропускные способности дорог, воздушных коридоров, линий связи и т.п. всегда ограничены сверху. Если известно, что фактическая загрузка будет заведомо меньше, задача рассматривается как простейшая. В противном случае учет этих ограничений приводит к более сложной транспортной задаче, называемой T_d -задачей. Ее модель имеет вид

$$L = \sum_i \sum_j X_{ij} C_{ij} \rightarrow \min; \quad (5.7)$$

$$\sum_{j=1}^n X_{ij} = a_i, \quad i = \overline{1, m}; \quad (5.8)$$

$$\sum_{i=1}^m X_{ij} = b_j, \quad j = \overline{1, n}; \quad (5.9)$$

$$0 \leq X_{ij} \leq d_{ij}, \quad \forall i, j, \quad (5.10)$$

где d_{ij} – пропускная способность коммуникации $i j$.

Ограничения (5.10) вносят существенные коррективы в свойства задачи. Из особенностей модели, присущих T -задаче, сохраняются все, кроме последней. В T_d -задаче условие сбалансированности не является достаточным для разрешимости задачи. Более того, в число необходимых условий существования решения помимо его входят еще две группы условий, отражающих физическую реализуемость решения:

$$\sum_{i=1}^m d_{ij} \geq b_j, \quad j = \overline{1, n}; \quad (5.11)$$

$$\sum_{j=1}^n d_{ij} \geq a_i, \quad i = \overline{1, m}. \quad (5.12)$$

Они требуют, чтобы суммарная пропускная способность коммуникаций, входящих в каждый ПН была не меньше объема поставок, а выходящих из ПО – не меньше количества вывозимого груза. Если хотя бы одно из них нарушается, задача заведомо неразрешима.

Однако и выполнение всех необходимых условий не гарантирует разрешимость T_d -задачи. Например, условия (5.1), (5.11) и (5.12) выполняются для

транспортной сети, показанной на рис. 5.1, что легко проверить. Но задача неразрешима, так как невозможно поставить во

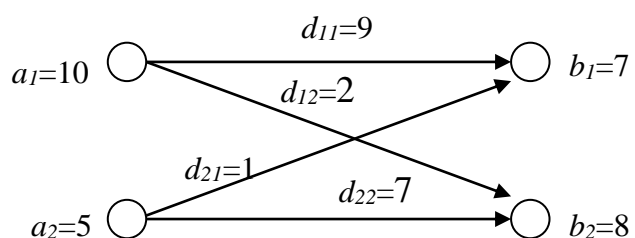


Рис. 5.1

второй пункт назначения 8 единиц груза.

5.1.3. Задачи с неоднородным грузом

В рассмотренных задачах по умолчанию предполагалось, что для отправителей и получателей грузы неразличимы – это задачи с однородным грузом. Если в перевозках участвуют несколько видов груза с одинаковыми или различными транспортными затратами, исходную многопродуктовую задачу можно разбить на задачи с однородным грузом (по числу видов).

Если же имеет место взаимозаменяемость грузов у получателей, то исходную задачу нельзя разделить на отдельные задачи. Например, получателю нужен каменный и бурый уголь. Известна потребность в том и другом и, кроме того, есть потребность, которая может быть удовлетворена любым из них. Последняя измеряется в единицах либо каменного, либо бурого угля. Такие задачи называют задачами с *неоднородным* грузом. В случае отсутствия ограничений на пропускные способности они легко преобразуются к задачам с однородным грузом.

Взаимозаменяемость грузов характеризуется коэффициентом взаимозаменяемости α . Например, если 1 т. каменного угля заменяет потребителю 2 т. бурого, то $\alpha = 2$. Зная α , все грузы можно привести к одному виду. Затем вместо одного исходного ПО вводится столько, сколько в нем видов груза. Аналогично каждый исходный ПН заменяется новыми, число которых равно числу видов потребностей. Наконец, определяются приведенные затраты на перевозки между всеми новыми пунктами. Если виды грузов в ПО и ПН совпадают, затраты на перевозку равны исходным C_{ij} ; если же они разные, то перевозка запрещается ($C_{ij} = M$). Между ПО с пересчитанным грузом (αa_i) и ПН с взаимозаменяемой потребностью затраты равны C_{ij} / α .

После таких преобразований модель задачи записывается аналогично случаю с однородным грузом, а ее размерность определяется числом пунктов, заменяющих исходные.

Для разрешимости задачи *необходимо* кроме сбалансированности, чтобы по каждому виду груза суммарные возможности были не меньше суммарной потребности (без учета взаимозаменяемой). Однако и при выполнении всех необходимых условий возможна неразрешимость задачи из-за присутствия запрещенных перевозок.

5.1.4. Многоиндексные задачи

Для учета дополнительных условий перевозки вводятся переменные с числом индексов более двух. В таких случаях говорят о *многоиндексных транспортных задачах*. Например, если существенное значение имеет вид транспорта, то в модели используются переменные X_{ijk} , означающие

количество груза, перевозимое из i -го пункта в j -й k -ым видом транспорта. Модель трехиндексной задачи зависит от конкретных условий. Если в исходных данных имеем производительность каждого вида транспорта p_k и не учитываются пропускные способности, то задача описывается *трипланарной* моделью:

$$\begin{aligned} L &= \sum_i \sum_j \sum_k C_{ijk} X_{ijk} \rightarrow \min \\ \sum_j \sum_k X_{ijk} &= a_i, \quad i = \overline{1, m}; \\ \sum_i \sum_k X_{ijk} &= b_j, \quad j = \overline{1, n}; \\ \sum_i \sum_j X_{ijk} &= p_k, \quad k = \overline{1, t}; \\ \forall X_{ijk} &\geq 0. \end{aligned}$$

Она идентична Т-задаче. Отличие лишь в числе переменных и групп условий. Поэтому каждая переменная входит в модель ровно три раза, а сбалансированность, как необходимое и достаточное условие разрешимости задачи, записывается в виде

$$\sum_i a_i = \sum_j b_j = \sum_k p_k.$$

Если транспортные средства принадлежат разным перевозчикам, то в модели будут фигурировать четырехиндексные переменные X_{ijkl} , где l – индекс перевозчика.

Дальнейшая детализация условий транспортировки может потребовать переменных с пятью и более индексами. В ряде случаев многоиндексные задачи удается свести к двухиндексным.

5.1.5. Транспортные задачи по критерию времени

При осуществлении перевозок определяющим показателем могут быть не затраты, а время доставки. Характерными примерами являются чрезвычайные ситуации, перевозка раненых, скоропортящихся продуктов и т. п. В таких задачах главное – как можно быстрее доставить все грузы. Тогда вместо матрицы транспортных затрат дается матрица времени $[t_{ij}]$, а критерий выражает время завершения всех перевозок:

$$T = \max t_{ij} \rightarrow \min,$$

где максимум берется по коммуникациям, на которых перевозки больше нуля. Предполагается, что перевозки между всеми пунктами начинаются одновременно и ведутся параллельно. Условия задачи записываются как и в случаях с критерием-затратами. Однако здесь критериальная функция нелинейна, что принципиально отличает эту задачу от ранее рассмотренных. В то же время она легко преобразуется к линейному виду, и решение задачи может быть получено любым универсальным методом

линейного программирования. Один приближенный метод рассмотрен в разд. 5.5. ▲

Для решения транспортных задач применяют специальные методы, которые учитывают их особенности и поэтому более эффективны, чем универсальные. К ним относятся распределительный метод, метод потенциалов, венгерский метод, метод Глейзала и др. Основными являются методы венгерский и потенциалов. Они применяются для решения задач как типа T , так и T_d . Ниже рассматривается второй из них.

5.2. Метод потенциалов

Концепция метода потенциалов та же, что и в симплекс-методе. Оптимальное решение ищется путем последовательных переходов от одного базисного решения (опорного плана) к другому с лучшим значением критерия. Но все шаги алгоритма выполняются проще, чем в симплекс-методе. В то же время метод потенциалов имеет много общего с распределительным методом и в связи с этим его иногда называют модифицированным распределительным методом.

Сначала рассмотрим метод применительно к T -задаче, а затем сделаем дополнения, позволяющие решать T_d -задачу.

5.2.1. Построение начального плана перевозок

Как было показано выше, размерность базисного решения или плана перевозок равна $m+n-1$, где m и n – число ПО и ПН *сбалансированной* задачи. Если задача открытая, то сначала ее необходимо сбалансировать.

Следует также иметь в виду, что в транспортных задачах вырожденность базисного решения встречается очень часто. В задаче заведомо будут вырожденные решения, если имеются такие неполные группы пунктов отправления и назначения, что суммарная возможность первых равна суммарной потребности вторых. Вырожденным может оказаться и начальное решение.

Для построения начального плана перевозок применяют правила северо-западного угла, минимального элемента и алгоритм Фогеля. Последний можно применять и как приближенный метод решения T -задачи.

Здесь мы рассмотрим только первые два способа. Хотя по аналогии легко предложить и другие правила. При этом важно соблюдать принцип: очередной переменной, включаемой в план, присваивать *максимально допустимое* значение. Этим обеспечится построение *базисного* решения.

Правило северо-западного угла

Все исходные данные и переменные сбалансированной T -задачи удобно представить в виде таблицы (табл. 5.1).

Построение плана начинается с северо-западной клетки таблицы, то есть первым определяется значение переменной X_{11} .

Таблица 5.1

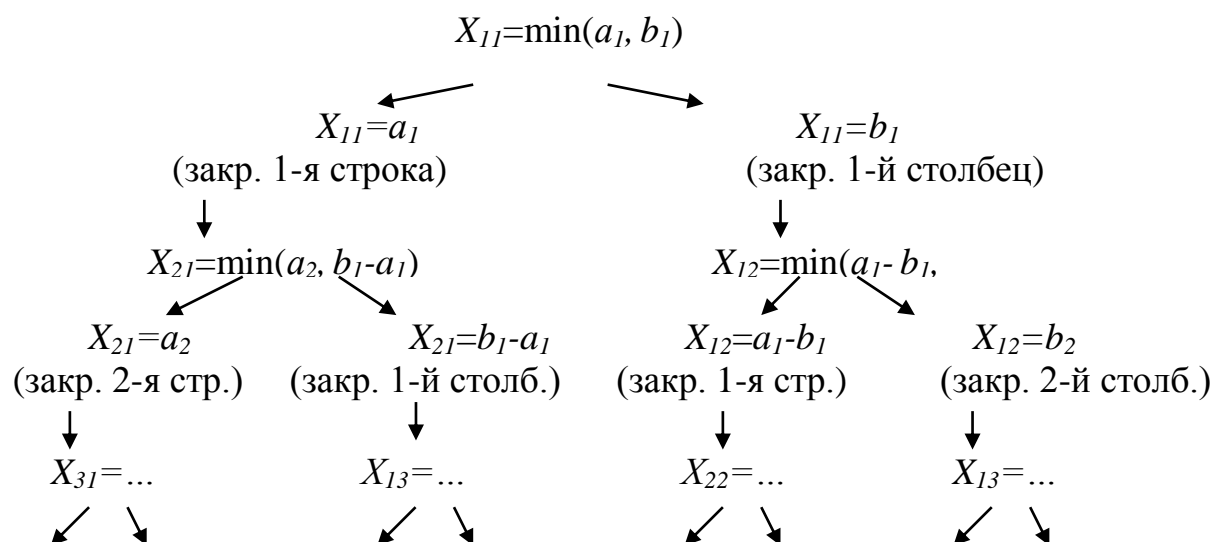
	b_1	b_2	...	b_n
a_1	X_{11} C_{11}	X_{12} C_{12}	...	X_{1n} C_{1n}
a_2	X_{21} C_{21}	X_{22} C_{22}	...	X_{2n} C_{2n}
...
a_m	X_{m1} C_{m1}	X_{m2} C_{m2}	...	X_{mn} C_{mn}

Так как оно должно быть максимально допустимым, то

$$X_{11} = \min(a_1, b_1).$$

При этом обязательно выполнится одно из равенств (5.3), (5.4), что соответствует закрытию строки или столбца: переменные в остальных клетках строки или столбца будут равны нулю. Конкретнее, если $X_{11}=a_1$, то закрывается первая строка и $X_{12}=X_{13}=\dots=X_{1n}=0$, а следующей базисной переменной будет X_{21} . Из указанного выше принципа следует $X_{21}=\min(a_2, b_1-a_1)$. Если же окажется, что $X_{11}=b_1$, то закроется первый столбец и следующей базисной переменной станет $X_{12}=\min(a_1-b_1, b_2)$.

Весь процесс построения начального плана можно представить в виде следующего дерева решений.



Общее правило определения значения очередной базисной переменной:

$$X_{ij}=\min(\text{остаток от } a_i, \text{остаток до } b_j). \quad (5.13)$$

Из него следует, что на каждом шаге закрывается или строка, или столбец, а на последнем шаге при назначении X_{mn} закрываются одновременно m -я строка и n -й столбец (так как задача сбалансированная). Таким образом, число базисных переменных равно $m + n - 1$. Построение начального плана завершено.

Пример 5.1. Исходные данные и построение начального плана показано в табл. 5.2. Значения базисных переменных выделены красным (серым) цветом, а порядок движения по клеткам отражен стрелками. Этому плану соответствуют суммарные затраты $L=1295$.

Таблица 5.2

Поставщик	Потребитель				Запасы груза
	B ₁	B ₂	B ₃	B ₄	
A ₁	6 75 →	7 25 ↓	3	5	100
A ₂	1	2 55 →	5 60 →	6 35 ↓	150
A ₃	3	10	20	1 50	50
Потребность в грузе	75	80	60	85	300

Правило минимального элемента.

В приведенном способе построения плана не участвовали затраты на перевозку. Следует ожидать, что учет затрат позволит получить начальный план, более близкий к оптимальному. Этим и отличается рассматриваемое правило.

Первой заполняется клетка с минимальными затратами. Пусть $\min C_{ij} = C_{kp}$. Тогда $X_{kp} = \min(a_k, b_p)$. Если при этом закрывается строка k , то в столбце p ищем клетку с минимальными затратами и определяем значение соответствующей переменной согласно (5.13). При закрытии столбца p действуем аналогично в строке k . В общем случае клетка, лежащая в закрытом столбце и/или закрытой строке является закрытой, иначе – открытой. На каждом шаге движение идет либо по столбцу, либо по строке и при этом отыскивается среди открытых клетка с минимальным значением C_{ij} .

Пример 5.2. Построим начальный план по правилу минимального элемента для задачи из примера 1. Результат представлен в табл. 5.3.

Таблица 5.3

Поставщик	Потребитель				Запасы груза
	B ₁	B ₂	B ₃	B ₄	
A ₁	6	7 5 →	3 60 →	5 35 ↓	100
A ₂	1	2	5	6	150

	75 →	75 ↑			
A_3	3	10	20	1	50
Потребность в грузе	75	80	60	50	85
				85	300

При таком начальном плане $L=665$, что меньше чем в примере 1. Однако нельзя утверждать, что для любых данных этот способ дает лучший план. Правильнее говорить, что правило минимального элемента эффективнее в среднем (на множестве задач). В то же время алгоритм реализации этого правила сложнее, чем правила северо-западного угла.

Применяется также вариант, в котором на каждом шаге ищется клетка с минимальными затратами среди *всех* открытых клеток. Такой способ еще сложнее, но в среднем дает планы, более близкие к оптимальным.

5.2.2. Переход от одного плана перевозок к другому

Как и в симплекс-методе, новый план можно получить из исходного заменой одной базисной переменной. Клетки с базисными переменными будем называть базисными или занятыми, остальные – небазисными или свободными. Для перехода к новому плану используется замкнутая цепь, которая строится в матрице перевозок по следующим правилам.

Построение начинается со свободной клетки, которую соединяют с базисной в строке (столбце). Последнюю соединяют с базисной в столбце (строке). Далее, чередуя движение по строкам и столбцам, продолжаем соединение занятых клеток так, чтобы вернуться в начальную. При этом не требуется, чтобы цепь включала все базисные клетки. Угловые клетки цепи назовем вершинами цепи. Тогда правило построения замкнутой цепи можно сформулировать проще: начальная вершина должна быть в свободной клетке, остальные – в занятых.

Такая цепь называется *циклом пересчета*. Он является геометрическим представлением разложения небазисного вектора условий при переменной в свободной клетке по векторам текущего базиса. Если базисная клетка не попала в цикл пересчета, то соответствующий базисный вектор имеет в этом разложении нулевой коэффициент. Так как любой небазисный вектор выражается через базис единственным образом, то для любой небазисной (свободной) клетки можно построить один и только один цикл пересчета. Примеры конфигурации циклов показаны на рис. 5.2.

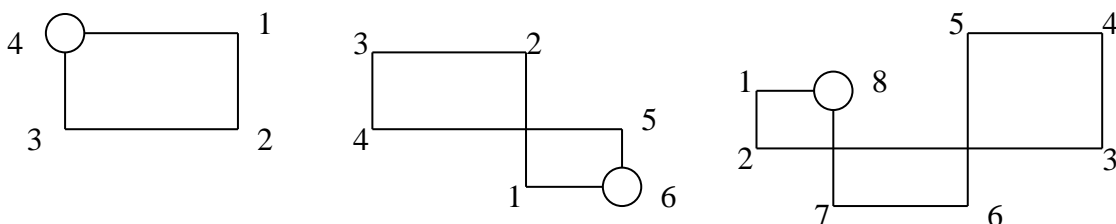


Рис. 5.2

Кружком выделена начальная (небазисная) клетка цикла. Нумеровать вершины можно в любом направлении. И начинать можно с любой вершины. На рисунке нумерация проведена с клетки, смежной начальной. В этом случае начальная клетка всегда будет четной.

Теперь становится очевидным, что в каждой строке и каждом столбце, по которым проходит цикл пересчета, будет две и только две вершины: одна четная и одна нечетная. Если бы оказалось вершин больше двух, то из базисных клеток образовался бы цикл, что невозможно. В этом легко убедиться на примере: допустим, в правом цикле на рис. 5.2 отрезки 1-8 и 5-4 лежат в одной строке, тогда вершины 1, 4, 3 и 2 образуют цикл.

В результате цикл пересчета, построенный в допустимой матрице перевозок, обладает замечательным свойством: если перемещать по нему некоторое количество груза $\theta > 0$, прибавляя его к X_{ij} в четных вершинах и вычитая из X_{ij} в нечетных, то условия задачи (5.3) и (5.4) не нарушатся. Чтобы новое решение было допустимым, то есть выполнялось и условие неотрицательности переменных, необходимо ограничить значение θ :

$$\theta \leq \theta_0 = \min X_{ij}, \quad ij \in \text{нечет}. \quad (5.14)$$

Здесь *нечет* – множество индексов переменных в нечетных вершинах цикла.

Для получения базисного решения (нового опорного плана) достаточно взять $\theta = \theta_0$. При этом переменная свободной клетки, на которой строился цикл, становится базисной со значением θ_0 , а переменная, доставляющая минимум в (5.14), обнуляется и переходит в небазисные.

Таким образом, переход от одного плана к другому в методе потенциалов заключается в построении цикла пересчета, определении θ_0 с последующим прибавлением к значениям переменных в четных вершинах и вычитанием в нечетных. Очевидно, что это значительно проще, чем в аналогичной процедуре симплекс-метода.

5.2.3. Признак оптимальности

При перемещении θ по циклу пересчета увеличиваются на эту величину значения переменных X_{ij} в четных вершинах, а следовательно, увеличиваются и затраты на перевозку на θC_{ij} . Одновременно уменьшаются на θ переменные в нечетных вершинах и на θC_{ij} соответствующие им затраты. Отсюда следует, что значение критерия в новом, $(k+1)$ -м решении можно определить по критерию в исходном решении и изменениям в клетках цикла:

$$L^{k+1} = L^k + \left(\sum_{ij \in \text{чет}} C_{ij} - \sum_{ij \in \text{нечет}} C_{ij} \right) \theta$$

или

$$L^{k+1} = L^k - \theta \Delta_{ij}, \quad (5.15)$$

где

$$\Delta_{ij} = \sum_{ij \in \text{нечет}} C_{ij} - \sum_{ij \in \text{чет}} C_{ij}. \quad (5.16)$$

Здесь, как и в симплекс-методе, Δ_{ij} – относительная оценка переменной X_{ij} , на которой построен цикл. Для базисных переменных оценка всегда равна нулю. Согласно (5.15) Δ_{ij} показывает, как изменится критерий (в какую сторону и насколько) при перемещении по циклу единицы груза ($\theta=1$).

Если $\Delta_{ij} > 0$, то введение X_{ij} в число базисных приведет к уменьшению суммарных затрат. Если же $\Delta_{ij} < 0$, критерий возрастет, что противоречит цели. Следовательно, решение нельзя улучшить, когда среди оценок нет положительных, и поэтому признак оптимальности имеет вид

$$\forall \Delta_{ij} \leq 0. \quad (5.17)$$

Если признак не выполняется, то новое решение целесообразно строить на основе клетки с максимальной оценкой (аналогично выбору в симплекс-методе при минимизации).

Вычисление оценок по формуле (5.16) требует построения цикла пересчета для каждой свободной клетки. Такой способ неэффективен для задач реальной размерности. Покажем, что возможен другой путь, исключающий построение циклов.

Поставим в соответствие каждому пункту отправления сбалансированной задачи некоторую величину U_i , $i=1, 2, \dots, m$, а каждому пункту назначения – V_j , $j=1, 2, \dots, n$ так, чтобы для базисных клеток выполнялись равенства

$$V_j - U_i = C_{ij}, \quad i, j \in \text{баз.} \quad (5.18)$$

Система (5.18) содержит $m+n-1$ уравнений с $m+n$ неизвестными. Присвоив одной из неизвестных некоторое произвольное значение, например, нуль, легко найти значения остальных. В таких случаях говорят о получении решения системы с точностью до постоянной величины. Дальше мы увидим, что произвольный выбор неизвестной и ее значения не влияет на конечный результат.

Зная U_i и V_j , можно вычислить относительную оценку для любого цикла в текущем плане перевозок. Покажем это на произвольно взятом цикле (рис. 5.3).

В скобках указаны индексы клеток (переменных), в которых расположены вершины цикла. Вычисляем относительную оценку свободной клетки i_0j_0 (небазисной переменной $X_{i_0j_0}$) по формуле (5.16):

$$\Delta_{i_0j_0} = C_{i_0j_1} - C_{i_1j_1} + C_{i_1j_2} - C_{i_2j_2} + C_{i_2j_0} - C_{i_0j_0}.$$

Заменим в этом выражении затраты в базисных клетках согласно (5.18). Тогда получим

$$\Delta_{i_0j_0} = V_{j_1} - U_{i_0} - V_{j_1} + U_{i_1} + V_{j_2} - U_{i_1} - V_{j_2} + U_{i_2} + V_{j_0} - U_{i_2} - C_{i_0j_0} = V_{j_0} - U_{i_0} - C_{i_0j_0}.$$

Выполненные сокращения не зависят от конфигурации цикла, так как все индексы кроме начальных входят в выражение два раза. Поэтому в итоге остаются только V_{j_0} , U_{i_0} и $C_{i_0j_0}$. Таким образом, для любой свободной клетки ij относительная оценка может быть вычислена без построения цикла пересчета по формуле

$$\Delta_{ij} = V_j - U_i - C_{ij}. \quad (5.19)$$

Из сравнения (5.18) и (5.19) видно, что для базисных клеток $\Delta_{ij} = 0$.

Новые переменные U_i и V_j называются потенциалами пунктов отправления и назначения соответственно, отсюда происходит название метода. Из формулы (5.19) следует, что значение постоянной величины при нахождении потенциалов из системы (5.18) не влияет на оценки.

Потенциалы можно интерпретировать как локальные цены. Если цена в пункте отправления i равна U_i и груз из него доставляется в пункт назначения j по коммуникации ij , то локальная цена в ПН возрастет по отношению к ПО на величину транспортных затрат:

$$V_j = U_i + C_{ij}. \quad (5.20)$$

Из этого соотношения также следует, что в оптимальном решении не может иметь место неравенство

$$V_j > U_i + C_{ij},$$

так как оно означает, что локальная цена в пункте j выше, чем в случае

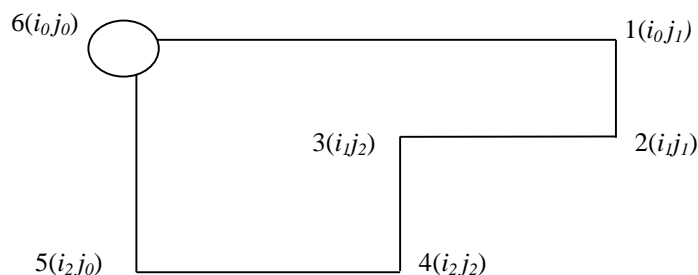


Рис. 5.3

прямой доставки из i в j . ▲

Приведенный способ определения оценок через потенциалы пригоден для любого опорного плана перевозок. Однако учитывая структуру матрицы оценок (нули в базисных клетках), можно оценки нового плана получить минуя вычисления потенциалов простым преобразованием матрицы оценок предшествующего плана.

Рассмотрим конкретно преобразование матрицы $\Delta^{(k)}$ в матрицу $\Delta^{(k+1)}$ на основе нового решения $\mathbf{X}^{(k+1)}$. Как отмечалось выше, новое решение получено вводом небазисной переменной с максимальной оценкой в $\Delta^{(k)}$. Пусть $\max \Delta_{ij} = \Delta_{kr}$. В матрице $\Delta^{(k)}$ отмечаем элементы, соответствующие базисным в новом решении $\mathbf{X}^{(k+1)}$ (на рис. 5.4 помечены символом *), максимальную оценку отмечаем особо. Далее строим цепочку выделения. Она строится с особо отмеченного элемента, который соединяют с отмеченными в этой строке. Затем отмеченные элементы, попавшие в цепочку, соединяют с отмеченными в их столбцах. Далее снова проводим соединение по строкам, и так до тех пор, пока не оборвутся все ветви.

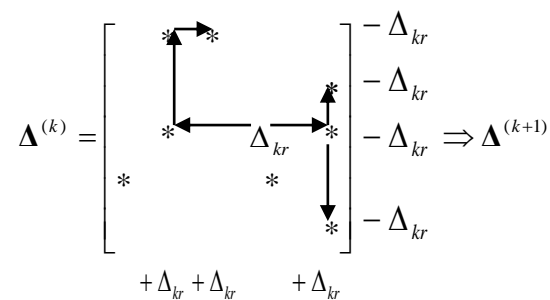


Рис.5.4

Элементы, попавшие в цепочку выделения, выделяют строку и столбец за исключением особо отмеченного элемента, который выделяет только строку. К выделенным столбцам прибавляем, а из выделенной строки вычитаем Δ_{kr} . Нетрудно увидеть, что при этом переменной X_{kr} будет соответствовать нулевая оценка, как и тем переменным из решения $\mathbf{X}^{(k)}$, которые сохранили статус базисных. Таким образом, преобразованная матрица соответствует новому опорному плану.

Провести выделение можно и иначе: сначала вычеркивать строку с максимальным элементом, затем вычеркивать столбцы, где есть элементы, отмеченные * в этой строке, и т.д. Вычеркнутые строки и столбцы являются выделенными.

5.2.4. Алгоритм метода потенциалов

Алгоритм включает предварительный и основной этапы.

Предварительный этап:

1. В матрице перевозок построить начальный план $\mathbf{X}^{(0)}$.

2. Решением системы (5.18) определить потенциалы всех пунктов в начальном плане.
3. Вычислить оценки небазисных переменных (свободных клеток) по формуле (5.19) и записать матрицу $\Delta^{(0)}$.

Основной этап (получены $X^{(k)}$ и $\Delta^{(k)}$):

1. Проверить оценки в $\Delta^{(k)}$. Если нет положительных, то перейти на п. 9.
2. Определить максимальную оценку $\Delta_{kr} = \max \Delta_{ij}$.
3. В матрице $X^{(k)}$ построить цикл пересчета на клетке kr .
4. В построенном цикле вычислить $\theta_0 = \min X_{ij}$, $ij \in \text{нечет}$.
5. Прибавить θ_0 в четных вершинах цикла и вычесть в нечетных, результат – матрица перевозок $X^{(k+1)}$.
6. В матрице $\Delta^{(k)}$ провести выделение строк и столбцов по решению $X^{(k+1)}$.
7. К выделенным столбцам прибавить θ_0 , а из выделенных строк вычесть Δ_{kr} , результат – матрица $\Delta^{(k+1)}$.
8. Перейти на п.1 основного этапа.
9. Конец.

Примечание. Если имелись запрещенные перевозки (некоторые $C_{ij}=M$), то соответствующие переменные в последнем решении должны равняться нулю. В противном случае задача неразрешима.

Пример 5.3. Решить методом потенциалов транспортную задачу, представленную в табл. 5.4.

Таблица 5.4

Поставщик (ПО)	Потребитель (ПН)				Количество груза
	B ₁	B ₂	B ₃	B ₄	
A ₁	3	8	2	1	10
A ₂	1	4	3	5	30
A ₃	7	2	1	6	40
Потребность в грузе	20	5	30	25	$\Sigma=80$

Решение. Задача сбалансированная. Начальный опорный план перевозок строим по правилу северо-западного угла. Полученный план невырожденный (табл. 5.5). Число базисных переменных (занятых клеток) $r=m+n-1=3+4-1=6$, они выделены цветом.

Таблица 5.5

Поставщик (ПО)	Потребитель (ПН)				Количество груза
	B ₁	B ₂	B ₃	B ₄	
A ₁	3 10 -	8	2	1 +	10
A ₂	1	4	3	5	30

	10	5	15		
	+		-		
A ₃	7	2	1	6	40
			15 +	25 -	
Потребность в грузе	20	5	30	25	Σ=80

Значение критерия в начальном плане

$$L^{(0)} = 10 * 3 + 10 * 1 + 5 * 4 + 15 * 3 + 15 * 1 + 25 * 6 = 270.$$

Вводим потенциалы u_i для ПО и v_j для ПН так, чтобы для базисных клеток выполнялись равенства:

$$\begin{aligned} v_1 - u_1 &= 3; & v_3 - u_2 &= 3; \\ v_1 - u_2 &= 1; & v_3 - u_3 &= 1; \\ v_2 - u_2 &= 4; & v_4 - u_3 &= 6. \end{aligned}$$

Полагая $u_1 = 0$, последовательно находим остальные потенциалы:

$$v_1 = 3; \quad u_2 = 2; \quad v_2 = 6; \quad v_3 = 5; \quad u_3 = 4 \quad v_4 = 10.$$

Вычисляем Δ_{ij} для свободных клеток:

$$\begin{aligned} \Delta_{12} &= v_2 - u_1 - c_{12} = 6 - 0 - 8 = -2; \\ \Delta_{13} &= v_3 - u_1 - c_{13} = 5 - 0 - 2 = 3; \\ \Delta_{14} &= v_4 - u_1 - c_{14} = 10 - 1 = 9; \\ \Delta_{24} &= v_4 - u_2 - c_{24} = 10 - 2 - 5 = 3; \\ \Delta_{31} &= v_1 - u_4 - c_{31} = 3 - 4 - 7 = -8; \\ \Delta_{32} &= v_2 - u_3 - c_{32} = 6 - 4 - 2 = 0. \end{aligned}$$

Записываем матрицу оценок для начального плана перевозок:

$$\Delta^{(0)} = \begin{bmatrix} 0 & -2 & 3 & \boxed{9^*} \\ 0^* & 0^* & 0^* & 3 \\ -8 & 0 & 0^* & 0^* \end{bmatrix} - 9$$

В начальном плане строим цикл на клетке с максимальной оценкой. Это клетка (1,4). Находим значение вводимой переменной:

$$\theta_0 = \min(10, 15, 25) = 10.$$

Переместив θ_0 по циклу, получаем новый план перевозок

$$\mathbf{X}^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 10 \\ 20 & 5 & \boxed{5} & \boxed{0} \\ 0 & 0 & \boxed{25} & \boxed{15} \end{bmatrix},$$

для которого $L^{(1)} = L^{(0)} - \theta_0 * \Delta_{14} = 270 - 10 * 9 = 180 \Rightarrow$ первая итерация улучшила критерий на 90 единиц.

Для выяснения статуса нового решения находим матрицу оценок. С этой целью в $\Delta^{(0)}$ отмечаем элементы, соответствующие базисным в $\mathbf{X}^{(1)}$, и строим цепочку выделения. Так как в строке с максимальной оценкой других отмеченных элементов

нет, выделенной оказывается только первая строка. Вычитая из нее Δ_{kr} , получаем матрицу

$$\Delta^{(I)} = \begin{bmatrix} -9 & -11 & -6 & 0 \\ 0 & 0 & 0 & 3 \\ -8 & 0 & 0 & 0 \end{bmatrix} - 3.$$

+3 +3

Как следует из анализа матрицы $\Delta^{(I)}$, решение $\mathbf{X}^{(I)}$ не является оптимальным. Следующее решение получаем с помощью построенного в $\mathbf{X}^{(I)}$ цикла, перемещая по нему $\theta_0 = \min(5, 15) = 5$:

$$\mathbf{X}^{(II)} = \begin{bmatrix} 0 & 0 & 0 & 10 \\ 20 & 5 & 0 & 5 \\ 0 & 0 & 30 & 10 \end{bmatrix}$$

Мы получили новый план перевозок с критерием

$$L^{(II)} = 180 - 5 \cdot 3 = 165.$$

Матрицу оценок этого плана находим преобразованием матрицы $\Delta^{(I)}$ аналогично описанному выше. В результате имеем

$$\Delta^{(II)} = \begin{bmatrix} -6 & -8 & -6 & 0 \\ 0 & 0 & -3 & 0 \\ -5 & 3 & 0 & 0 \end{bmatrix} - 3$$

+3 +3 +3

В матрице есть положительный элемент, поэтому на клетке (3,2) строим цикл пересчета. Определяем $\theta_0 = \min(5, 10) = 5$ и, перемещая 5 по циклу, находим очередной план перевозок

$$\mathbf{X}^{(III)} = \begin{bmatrix} 0 & 0 & 0 & 10 \\ 20 & 0 & 0 & 10 \\ 0 & 5 & 30 & 5 \end{bmatrix},$$

которому соответствует значение критерия $L^{(III)} = 165 - 5 \cdot 3 = 150$. Преобразуя матрицу $\Delta^{(II)}$, получаем

$$\Delta^{(III)} = \begin{bmatrix} -6 & -11 & -6 & 0 \\ 0 & -3 & -3 & 0 \\ -5 & -3 & 0 & 0 \end{bmatrix}.$$

Эта матрица не содержит положительных оценок, следовательно, план $\mathbf{X}^{(III)}$ является оптимальным. Согласно этому плану от 1-го поставщика надо поставить 10 ед. продукции 4-му потребителю, от 2-го поставщика - 20 ед. первому и 10 ед. четвертому потребителям, от 3-го поставщика - 5, 30 и 5 ед. соответственно 2, 3 и 4 потребителям. Такая схема перевозок обеспечивает минимум суммарных затрат, которые равны 150. ▲

Примечания. 1).Метод потенциалов применим и для решения трипланарных задач. Отличие лишь в том, что циклы пересчета и цепочки выделения строятся не на плоскости, а в трехмерном пространстве. 2).Если на каждой итерации оценки вычислять непосредственно по построенным циклам (на всех свободных клетках), то получим *распределительный* метод. Поэтому метод потенциалов называют также модифицированным распределительным методом. Однако метод потенциалов значительно эффективнее распределительного на задачах средней и большой размерности.

5.2.5. Двойственная пара транспортных задач

Построим двойственную задачу простейшей транспортной задачи (Т-задачи). Предварительно изменим знаки в выражении критерия и в условиях по пунктам назначения. Тогда модель прямой задачи примет вид:

$$\begin{aligned} L_1 &= -\sum_i \sum_j C_{ij} x_{ij} \rightarrow \max; \\ U_i : \quad &\sum_{j=1}^n x_{ij} = a_i, i = \overline{1, m}; \\ V_j : \quad &-\sum_{i=1}^m x_{ij} = -b_j, j = \overline{1, n}; \\ &\forall x_{ij} \geq 0. \end{aligned}$$

Здесь слева от равенств записаны сопоставленные им двойственные переменные. Модель двойственной задачи запишем по правилам, приведенным в разд. 4.11.3:

$$\tilde{L}_1 = \sum_i a_i U_i - \sum_j b_j V_j \rightarrow \min ;$$

$$U_i - V_j \geq -C_{ij} \Rightarrow V_j - U_i \leq C_{ij}, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Если C_{ij} перенести в левую часть, то согласно (5.19) условия двойственной задачи приобретают смысл признака оптимальности $\forall \Delta_{ij} \leq 0$. Итак, если выполняются условия прямой и двойственной задач, решение оптимально. Теперь понятно, что потенциалы представляют собой переменные двойственной задачи.

Из теорем двойственности известно, что в оптимальном решении критерии прямой и двойственной задач равны. Для рассматриваемой двойственной пары это означает, что

$$L^* = \tilde{L}^* = -\tilde{L}_1^* = \sum_j b_j V_j^* - \sum_i a_i U_i^*. \quad (5.21)$$

Отсюда находим:

$$\frac{\partial L^*}{\partial b_j} = V_j^*; \quad \frac{\partial L^*}{\partial a_i} = -U_i^*.$$

Учитывая линейность (5.21), полный дифференциал запишем в виде

$$\Delta L^* = V_j^* \Delta b_j - U_i^* \Delta a_i.$$

Изменения a_i и b_j могут быть только равными, иначе нарушится сбалансированность задачи. Если положить $\Delta a_i = \Delta b_j = 1$, то получим

$$\Delta L^* = V_j^* - U_i^*.$$

Следовательно, разность потенциалов показывает, как изменится оптимальное значение критерия при одновременном изменении соответствующих потребностей и возможностей на единицу.

5.3. Приведение открытой транспортной задачи к закрытой

В открытой или несбалансированной задаче имеет место неравенство

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

Прежде чем решать такую задачу, необходимо привести ее к сбалансированному виду. В зависимости от ситуации сбалансировать задачу можно формальным способом без обращения к ЛПР или с привлечением дополнительной информации от ЛПР.

Рассмотрим формальные приемы. Пусть в исходной задаче предложение превышает спрос:

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

Тогда условия задачи имеют вид

$$\sum_{j=1}^n x_{ij} \leq a_i, \quad i = \overline{1, m}; \quad (5.22)$$

$$\begin{aligned} \sum_{i=1}^m x_{ij} &= b_j, \quad j = \overline{1, n}; \\ \forall x_{ij} &\geq 0. \end{aligned} \quad (5.23)$$

В каждое неравенство (5.22) введем дополнительную переменную $x_{i, n+1}$. В сумме эти переменные должны равняться величине дебаланса:

$$\sum_{i=1}^m x_{i, n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j = b_{n+1}.$$

Добавляя это равенство к условиям (5.23), получаем закрытую задачу:

$$\begin{aligned} \sum_{j=1}^{n+1} x_{ij} &= a_i, \quad i = \overline{1, m}; \\ \sum_{i=1}^m x_{ij} &= b_j, \quad j = \overline{1, n+1}; \\ \forall x_{ij} &\geq 0. \end{aligned}$$

Потребность b_{n+1} называют фиктивной. Таким образом, чтобы сбалансировать задачу, достаточно ввести фиктивного потребителя с

потребностью, равной дебалансу. Практически это означает, что к исходной таблице добавляется один столбец с потребностью b_{n+1} и затратами $C_{i,n+1}=0$. Ненулевые дополнительные переменные в оптимальном решении будут показывать количество груза, остающееся в соответствующих ПО.

Второй случай несбалансированности задачи имеет место, когда спрос превышает предложение:

$$\sum_i a_i < \sum_j b_j.$$

При этом исходные условия записываются в виде:

$$\begin{aligned} \sum_j x_{ij} &= a_i, \quad i = \overline{1, m}; \\ \sum_i x_{ij} &\leq b_j, \quad j = \overline{1, n}; \\ \forall x_{ij} &\geq 0. \end{aligned}$$

Поступаем аналогично первому случаю. Введем в каждое неравенство дополнительную переменную $x_{m+1,j}$. Очевидно, что сумма этих переменных равна величине дебаланса:

$$\sum_j x_{m+1,j} = \sum_j b_j - \sum_i a_i = a_{m+1}.$$

С учетом этого равенства сбалансированная модель принимает вид:

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= a_i, \quad i = \overline{1, n+1}; \\ \sum_{i=1}^{m+1} x_{ij} &= b_j, \quad j = \overline{1, n}; \\ \forall x_{ij} &\geq 0. \end{aligned}$$

Такое преобразование соответствует введению фиктивного поставщика (дополнительной строки) с возможностью a_{m+1} и нулевыми затратами $C_{m+1,j}$. Дополнительная переменная $x_{m+1,j}$ имеет смысл количества груза, недопоставленного j -му ПН.

Рассмотренный формальный способ будет неприемлем, если потребители по-разному реагируют на недопоставки. Тогда возможны два варианта решения задачи:

1. ЛПР корректирует потребности, обеспечивая баланс.

2. Выявляется и учитывается влияние недопоставок для каждого потребителя. Если зависимость потерь от величины недопоставки линейная, то задача остается в классе линейных. В этом случае задача балансируется как при формальном подходе, но в дополнительной строке в качестве затрат берутся удельные потери от недопоставки.

Если ожидается, что спрос будет длительное время превышать существующие возможности на величину a_{m+1} , то встает вопрос о расширении производства. Он может решаться в рамках транспортной модели следующим образом. Проектируются варианты увеличения

производства, каждый на величину a_{m+1} . В исходную таблицу добавляется столько строк, сколько предлагается вариантов. При k вариантах это приведет к противоположному дебалансу, равному $(k-1) \cdot a_{m+1}$. Поэтому для сбалансированности модели добавляется фиктивный потребитель с такой потребностью. А в качестве затрат во всех клетках таблицы принимаются суммарные затраты на перевозку и производство $C'_{ij} = C_{ij} + C_i$, где C_i – себестоимость в i -м ПО. Исключение составляет фиктивный столбец: в первых m клетках затраты равны M , а в остальных – нулю. Те варианты, которые в оптимальном решении закрепятся за фиктивным потребителем, должны быть отброшены.

При прогнозировании длительного превышения возможностей над спросом может возникнуть вопрос о сокращении производства. Он также может быть представлен в виде транспортной задачи. Достаточно в затраты включить себестоимость, как в предыдущем случае, и добавить фиктивного потребителя с потребностью b_{n+1} и нулевыми затратами. Оптимальные значения дополнительных переменных в фиктивном столбце дадут величину сокращения производства в соответствующих ПО с учетом полных затрат.

5.4. Метод потенциалов для Та-задачи

Транспортная задача с ограниченными пропускными способностями (5.7)-(5.10) отличается от Т-задачи наличием ограничений сверху на перевозки:

$$0 \leq x_{ij} \leq d_{ij}.$$

Они существенно усложняют решение задачи. В плане перевозок не все положительные переменные являются базисными. В невырожденном решении базисные переменные могут принимать только значения больше нуля и меньше пропускной способности:

$$0 < x_{ij} < d_{ij}. \quad (5.24)$$

В вырожденном решении некоторые базисные переменные равны граничным значениям (0 или d_{ij}).

Если задача не сбалансирована, то, как и в Т-задаче, добавляют столбец или строку с нулевыми затратами, но с бесконечной пропускной способностью.

Правило северо-западного угла непригодно для построения начального решения. Его строят по одному из вариантов правила минимального элемента. При этом принцип определения значений переменных сохраняется: на каждом шаге присваивается максимальное допустимое значение согласно формуле

$$X_{ij} = \min(\text{остаток от } a_i, \text{остаток до } b_j, d_{ij}).$$

Если минимум достигается на d_{ij} , то не закрывается ни строка, ни столбец, и следует продолжать движение по строке или столбцу (если двигались по столбцу). Однако может оказаться, что в строке (столбце)

больше нет открытых клеток, а она не закрыта. Это значит, что начальный план получается *недопустимым*. Если часть строк (столбцов) не закрыта, то обязательно не закроются и некоторые столбцы (строки). Так как задача сбалансированная, то суммарная величина незакрытия строк

$$\sum_i \left(a_i - \sum_j x_{ij} \right) \text{ будет равна суммарному незакрытию столбцов } \sum_j \left(b_j - \sum_i x_{ij} \right).$$

Обозначим эту величину γ . Чтобы в этом случае завершить построение начального плана, добавляют фиктивного потребителя (столбец) и фиктивного поставщика (строку) с одинаковой потребностью и возможностью γ . Так как их клетки соответствуют искусственным переменным, которые в разрешимой задаче должны стать равными нулю, затраты в них полагают бесконечно большими (M), а в клетке на пересечении фиктивного столбца с фиктивной строкой – равными нулю. Пропускные способности в фиктивных клетках не лимитируются. В процессе работы алгоритма план станет допустимым, когда все искусственные переменные обнулятся, то есть в юго-восточной клетке перевозка станет равна γ (фиктивный поставщик замкнется на фиктивного потребителя).

После построения начального плана определяются базисные клетки по условию (5.24). Если таких клеток окажется меньше $m+n-1$ (вырожденный план), то в число базисных включают переменные (клетки), равные нулю или d_{ij} . При этом на базисных клетках не должен строиться замкнутый цикл, иначе клетки добавлены неверно.

Пример 5.4. Исходные данные задачи приведены в табл. 5.6. В клетках слева даны пропускные способности (красным цветом), справа – затраты на перевозки (синим цветом). Задача сбалансированная. Начальный план строим по правилу минимального элемента, порядок построения показан стрелками (табл. 5.7). Строка 4 и столбцы 2 и 3 не закрылись: $\gamma=3$.

Таблица 5.6

$a_i \backslash b_j$	15	33	25
12	8 3	5 1	10 5
6	5 2	9 4	4 7
20	12 6	11 2	10 3
35	20 4	10 5	7 9

Таблица 5.7

$a_i \backslash b_j$	15	33	25
12	8 3	5 1	10 5
6	5 2	9 4	4 7
20	12 6	11 2	10 3
35	20 4	10 5	7 9

Дополнительные данные из таблицы 5.7:

- В строке 1: стрелка от (1,2) к (1,3) с цифрой 5; стрелка от (1,3) к (1,4) с цифрой 7.
- В строке 2: стрелка от (2,3) к (2,4) с цифрой 4.
- В строке 3: стрелка от (3,3) к (3,4) с цифрой 2.
- В строке 4: стрелка от (4,3) к (4,4) с цифрой 5.
- В строке 4: стрелка от (4,1) к (4,2) с цифрой 4.
- В строке 4: стрелка от (4,2) к (4,3) с цифрой 10.
- В строке 4: стрелка от (4,3) к (4,4) с цифрой 7.
- В строке 4: стрелка от (4,4) к (4,3) с цифрой 9.
- В строке 4: стрелка от (4,3) к (4,2) с цифрой 15.
- В строке 4: стрелка от (4,2) к (4,1) с цифрой 10.
- В строке 4: стрелка от (4,1) к (4,2) с цифрой 7.

Поэтому добавляем фиктивные строку и столбец, и в клетки незакрытых столбцов и строки записываем недостающие перевозки. В результате выполняется баланс по всем строкам и столбцам. Построенный *искусственный* (недопустимый) начальный план приведен в табл. 5.8

Так как общее число пунктов равно 9, то базисных переменных должно быть 8. Из сравнения значений x_{ij} и d_{ij} находим только 7 базисных переменных (базисные клетки закрашены), то есть план вырожденный. В качестве недостающей базисной клетки возьмем клетку 4,2 (закрашена более темным цветом), в которой значение переменной находится на верхней границе.

Нетрудно убедиться, что ни из каких базисных клеток нельзя построить замкнутый цикл. (Чтобы соблюдалось это требование в случае вырожденности плана, можно рекомендовать рассматривать в качестве кандидатов на включение в число базисных в первую очередь те клетки, в которых происходит поворот на 90 градусов при построении плана. Так, в данном примере такими являются клетки 4,2 и 4,3.) ▲

Таблица 5.8

$a_i \backslash b_j$	15	33	25	$\gamma=3$
12	8 3	5 1 5	10 5 7	M
6	5 2	9 4 6	4 7	M
20	12 6	11 2 11	10 3 9	M
35	20 4 15	10 5 10	7 9 7	M 3
$\gamma=3$	M	M 1	M 2	0

Теперь рассмотрим изменения на основном этапе алгоритма. Признак оптимальности в T_d -задаче расширяется. Согласно (5.15) при введении в решение небазисной переменной $x_{ij}=0$, то есть ее увеличении, критерий

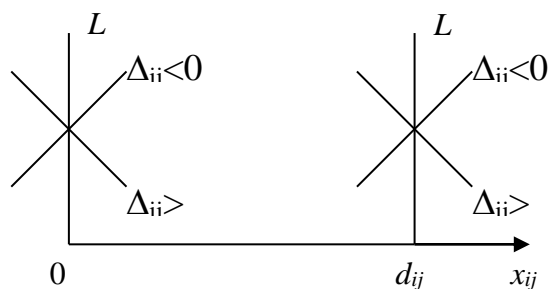


Рис. 5.5

уменьшается при $\Delta_{ij}>0$ и увеличивается при $\Delta_{ij}<0$. Если же вводить переменную $x_{ij}=d_{ij}$, а это означает ее уменьшение, то критерий уменьшится при $\Delta_{ij}<0$ и возрастет при $\Delta_{ij}>0$. Этот характер изменения критерия показан на рис. 5.5. Поэтому решение не может быть улучшено, если выполняются

условия, составляющие признак оптимальности задачи с ограниченными пропускными способностями:

$$\begin{cases} \Delta_{ij} \leq 0, & \text{если } x_{ij} = 0; \\ \Delta_{ij} = 0, & \text{если } 0 < x_{ij} < d_{ij}; \\ \Delta_{ij} \geq 0, & \text{если } x_{ij} = d_{ij}. \end{cases} \quad (5.25)$$

Если условия не выполняются хотя бы для одной небазисной клетки, решение может быть улучшено. В связи с этим введем два множества:

- множество индексов переменных на нижней границе

$$G_0 = \{ij \mid \Delta_{ij} > 0 \text{ \& } x_{ij} = 0\};$$

- множество индексов переменных на верхней границе

$$G_d = \{ij \mid \Delta_{ij} < 0 \text{ \& } x_{ij} = d_{ij}\}.$$

Очевидно, что объединенное множество $G = G_0 \cup G_d$ является множеством индексов перспективных переменных (клеток): введение любой из них приведет к улучшению критерия. В Т-задаче имеется только множество G_0 , и выбор производится из него. Здесь же выбор переменной, вводимой в базис, осуществляется на множестве G :

$$\max_{ij \in G} |\Delta_{ij}| = \Delta_{kr}. \quad (5.26)$$

Так как переменная вводится по-разному, увеличивается с нижней границы и уменьшается с верхней, вычисление θ_0 и перемещение по циклу выполняется иначе, чем в Т-задаче. При определении θ_0 теперь необходимо учитывать обе границы: при вычитании нельзя вычесть больше, чем имеется, а при добавлении недопустимо превысить пропускную способность.

Из (5.26) следует, что возможны 2 варианта при выборе вводимой переменной и соответственно 2 варианта перехода к новому плану:

1. Если $kr \in G_0$, то цикл строится на клетке, в которой перевозка равна нулю. Новый план получается прибавлением θ_0 в четных вершинах цикла и вычитанием в нечетных. Поэтому

$$\theta_0 = \min \left[\min_{ij \in \text{неч}} x_{ij}; \min_{ij \in \text{четн}} (d_{ij} - x_{ij}) \right]. \quad (5.27)$$

2. Если $kr \in G_d$, цикл строится на клетке, в которой перевозка равна d_{ij} . В этом случае вводимая переменная должна уменьшаться. Поэтому перемещение по циклу состоит в вычитании θ_0 в четных вершинах и прибавлении в нечетных. Отсюда следует, что

$$\theta_d = \min \left[\min_{ij \in \text{чет}} x_{ij}; \min_{ij \in \text{нечет}} (d_{ij} - x_{ij}) \right]. \quad (5.28)$$

В обоих вариантах значение критерия улучшается на величину $\theta_0 |\Delta_{kr}|$.

Таким образом, алгоритм решения сбалансированной T_d -задачи включает следующие шаги:

1. Построение начального плана перевозок. План может получиться как допустимый, так и искусственный (недопустимый).
2. Выделение базисных клеток. Если их меньше $m+n-1$, то добавляются клетки на границе.
3. Нахождение потенциалов из системы (5.18).
4. Вычисление оценок по формуле (5.19)
5. Начало цикла. Определение множества G по матрицам плана и оценок.
6. Проверка признака оптимальности: если $G=\emptyset$ (эквивалент (5.25)), переход на шаг 10.
7. Определение вводимой переменной (клетки kr) по (5.26) и построение цикла пересчета.
8. Построение нового плана: вычисление θ_0 в зависимости от принадлежности kr по (5.27) или (5.28) и соответствующее перемещение по циклу.
9. Получение матрицы оценок нового плана с помощью преобразования матрицы оценок старого плана (как в T -задаче). Переход на шаг 5.
10. Конец. Полученный план является оптимальным, если не содержит запрещенных перевозок (с затратами M).

Когда решение начинается с искусственного плана, то после достижения допустимого решения можно сократить матрицы перевозок и оценок за счет отбрасывания фиктивных столбца и строки. Если в них было ровно две базисные клетки, то пересчитывать матрицу оценок не надо. Иначе она рассчитывается через потенциалы для сокращенного плана. Однако сокращение матриц не является обязательным.

5.5. Решение задачи по критерию времени

Как было показано в разд. 5.1.5, такая задача исходно является нелинейной, но может быть легко преобразована к линейной. Однако получаемая линейная модель громоздка и для нахождения решения требует применения универсальных методов линейного программирования.

Поэтому может оказаться целесообразным обратиться к приближенным, но более простым методам решения. Один из них рассмотрен ниже. Он использует идеи методов решения обычных транспортных задач и заключается в следующем.

Любым из способов строится начальный план перевозок. Затем определяется текущее значение критерия $T^{(0)}$ как максимальное время в занятых клетках ($x_{ij}>0$). Далее рассмотрим действия в цикле.

Пусть на k -й итерации получен план со значением критерия $T^{(k)}$. Оно может быть уменьшено, если освободить клетку с $t_{ij} = T^{(k)}$. С этой целью на клетке строится разгрузочный цикл так, чтобы в нечетных вершинах выполнялось неравенство $t_{ij} < T^{(k)}$, а в четных – $x_{ij} > 0$ (исходная вершина – четная). Очевидно, что такие правила позволяют в общем случае строить более одного цикла на выбранной клетке. В цикле вычисляется θ_0 как минимальная перевозка в четных вершинах. Вычитая θ_0 в четных вершинах и прибавляя в нечетных, получаем новый план. Если клетка с максимальным временем была единственной и перевозка в ней стала равна нулю, новый план улучшил значение критерия. Если клетка не обнулилась, то на ней строится другой разгрузочный цикл. В случае нескольких клеток со временем $T^{(k)}$ для улучшения критерия необходимо разгрузить все. Решение завершается, когда нельзя разгрузить клетку (клетки), определяющую значение критерия.

Естественно, что рассмотренный метод не гарантирует получение оптимального решения.

5.6. Транспортные задачи в сетевой постановке (транспортные сети)

Транспортную задачу можно представить в виде ориентированного графа с одним истоком (в него не входит ни одна дуга) и с одним стоком (из него не выходят дуги), который называют в этом случае сетью. Вершинам графа ставятся в соответствие пункты отправления, назначения и промежуточные пункты. Основным параметр вершины – количество груза. Дуги отображают коммуникации. Им могут быть приписаны такие параметры как количество груза, затраты на перевозку, пропускная способность.

Исходный граф транспортной задачи легко сводится к сети с одним истоком и одним истоком путем введения фиктивных пунктов t (исток) и s (сток). Фиктивным дугам приписываются значения параметров: $d_{ti} = a_i$,

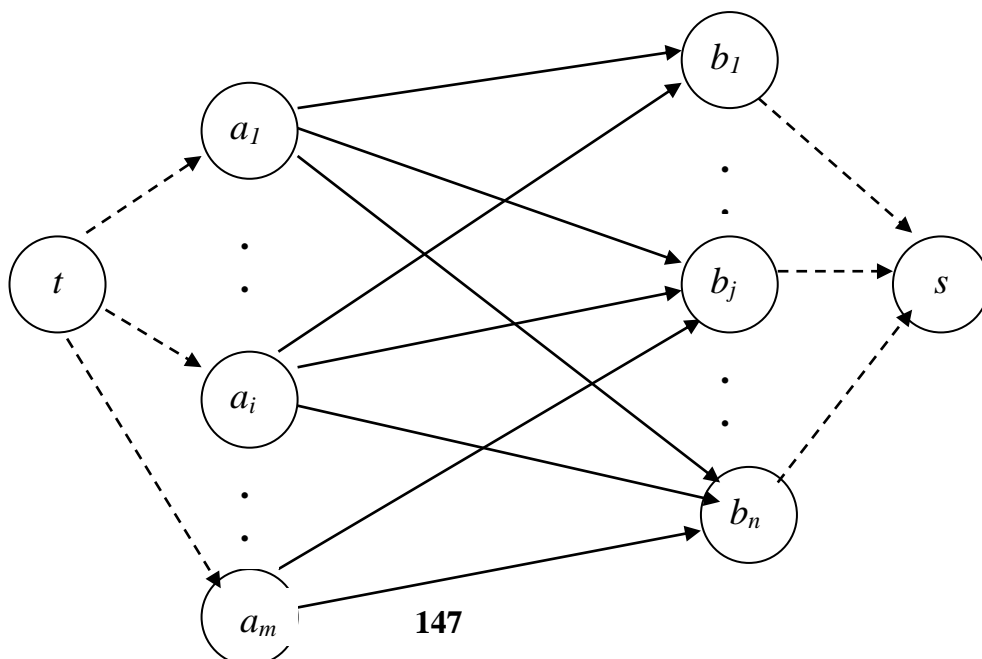


Рис. 5.6

$d_{js}=b_j$, $C_{ti}=C_{js}=0$. Пример сети транспортной задачи без промежуточных пунктов приведен на рис. 5.6.

Модель T_d -задачи в сетевой постановке имеет вид:

$$\sum \sum C_{ij}x_{ij} \rightarrow \min; \quad (5.29)$$

$$\sum_{lk \in U_k^+} x_{lk} = \sum_{kp \in U_k^-} x_{kp}, \quad k \neq t, k \neq s; \quad (5.30)$$

$$\sum_{ti \in U_t^-} x_{ti} = \sum_{js \in U_s^+} x_{js} = Z. \quad (5.31)$$

В сбалансированной транспортной задаче

$$Z = \sum a_i = \sum b_j; \quad (5.32)$$

$$0 \leq x_{ij} \leq d_{ij}. \quad (5.33)$$

Равенства (5.30) отражают условия баланса для всех пунктов кроме источника и стока. Баланс для последних представлен уравнением (5.31). В модели использованы обозначения: U_k^+, U_k^- – множество дуг, входящих в вершину k и выходящих из нее, Z – новая величина, называемая **потоком сети**.

Наибольший интерес представляет постановка задачи, в которой критерием является поток сети:

$$Z \rightarrow \max; \quad (5.34)$$

$$\sum_{lk \in U_k^+} x_{lk} = \sum_{kp \in U_k^-} x_{kp}, \quad k \neq t, k \neq s; \quad (5.35)$$

$$\sum_{ti \in U_t^-} x_{ti} = \sum_{js \in U_s^+} x_{js} = Z. \quad (5.36)$$

$$0 \leq x_{ij} \leq d_{ij}. \quad (5.37)$$

Задача (5.34) – (5.37) называется задачей о **максимальном потоке**. Она имеет большое практическое значение. Для нее разработаны алгоритмы, которые эффективнее методов решения транспортных задач. Они работают непосредственно с сетью как разновидностью графов.

В связи с этим напомним понятие разреза графа (сети), которое используется в основополагающей теореме Форда-Фалкерсона.

Пусть дан ориентированный граф $G=(V, U)$, где V и U – множества вершин и дуг соответственно. *Разрезом сети* на подмножестве вершин $A \subset V$, $A \neq \emptyset$, $A \neq V$, $t \in A$, $s \in V \setminus A$ называется множество дуг $ij \in U$ таких, что $i \in A$ & $j \in V \setminus A$. Обозначим его $P(A)$. Сумма пропускных способностей дуг разреза называется *величиной (пропускной способностью) разреза*:

$$d(A) = \sum_{ij \in P(A)} d_{ij}.$$

Пример 5.5. Построим один из разрезов сети, приведенной на рис.5.7.

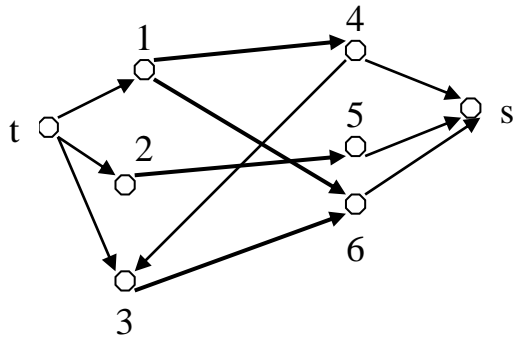


Рис. 5.7

Если $A = \{t, 1, 2, 3\}$, то разрезом будет множество дуг $P(A) = \{1, 4; 2, 5; 3, 6\}$, а его величина определяется как $d(A) = d_{14} + d_{25} + d_{36}$. Дуги, составляющие этот разрез, выделены жирными линиями. ▲

Разрез сети, имеющий минимальную пропускную способность, называется

минимальным разрезом.

Можно показать, что задачи максимизации потока и минимизации величины разреза являются двойственной парой. Из этого факта следует Теорема Форда и Фалкерсона:

Величина потока сети (от истока к стоку) не превосходит пропускной способности минимального разреза и существует максимальный поток, величина которого равна пропускной способности минимального разреза.

Методы решения задачи о максимальном потоке основаны на последовательном увеличении потока при соблюдении условий (5.35)-(5.37). При этом легко увидеть аналогию с перемещением по циклу в методах решения транспортных задач.

Аналогом цикла пересчета является *увеличивающая цепь*. Это цепь, соединяющая исток и сток, все дуги которой *допустимые*. Дуга является *допустимой увеличивающей*, если ее направление совпадает с направлением потока и поток на ней меньше пропускной способности, то есть $x_{ij} < d_{ij}$. Дуга считается *допустимой уменьшающей*, если направление дуги противоположно потоку и $x_{ij} > 0$.

На увеличивающей дуге поток может возрасти на величину $\theta_{ij} = d_{ij} - x_{ij}$, а на уменьшающей дуге возможно снижение потока, равное $\theta_{ij} = x_{ij}$. Следовательно, максимальное допустимое изменение величины потока по увеличивающей цепи определяется как минимальное из возможных:

$$\theta_0 = \min_{ij \in \text{ув.цепи}} \theta_{ij}. \quad (5.38)$$

Таким образом, максимальный поток сети может быть определен по следующему алгоритму.

1. Задать начальную величину потока, обеспечиваемую потоками дуг при выполнении условий (5.35)-(5.37).
Примечание. Очевидно, что в качестве начального всегда можно взять нулевой поток.
2. Построить увеличивающую цепь. Если построить невозможно, то решение завершено.
3. По формуле (5.38) вычислить θ_0 .

4. Переместить вдоль цепи θ_0 , прибавляя к потокам на увеличивающих дугах и вычитая из потоков уменьшающих дуг. В результате поток сети увеличивается на θ_0 . Перейти на шаг 2.

Пример 5.6. Определить максимальный поток сети на рис. 5.8.

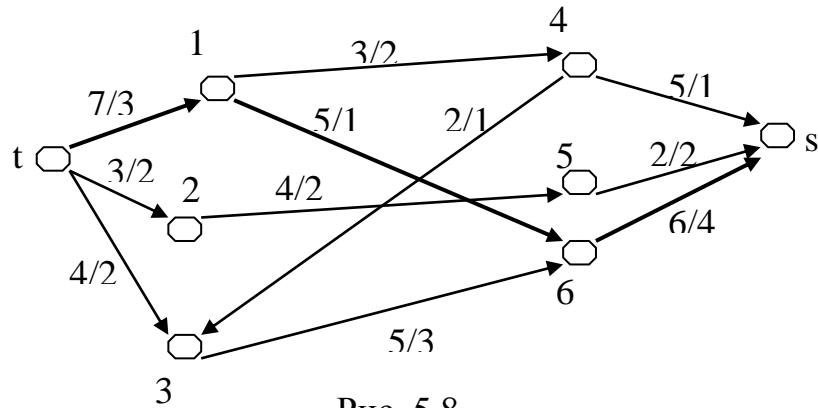


Рис. 5.8

Пропускные способности дуг показаны у стрелок перед косой чертой.

Задаем начальный поток. Значения начальных потоков дуг даны за косой чертой, они удовлетворяют условиям задачи. Величина потока сети $Z^{(0)}=7$.

Первая итерация.

Строим увеличивающую цепь. Она показана на рис. 5.8 утолщенными линиями. Определяем приращение потока: $\theta_0 = \min(7-3, 5-1, 6-4)=2$. Увеличиваем потоки дуг цепи на 2 (рис. 5.9). $Z^{(1)} = Z^{(0)} + \theta_0 = 7+2=9$.

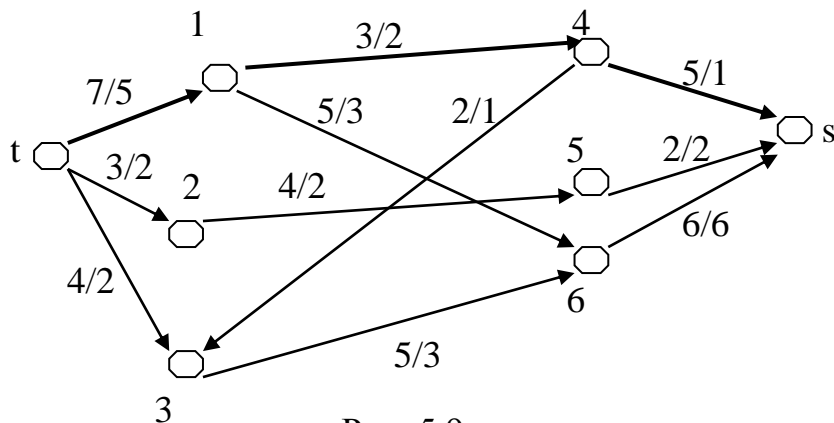


Рис. 5.9

Вторая итерация.

Строим увеличивающую цепь $\{t, 1; 1, 4; 4, s\}$ (рис. 5.9). $\theta_0 = \min(7-5, 3-2, 5-1)=1$. Увеличиваем потоки по дугам цепи на θ_0 (рис. 5.10). $Z^{(2)} = Z^{(1)} + \theta_0 = 9+1=10$.

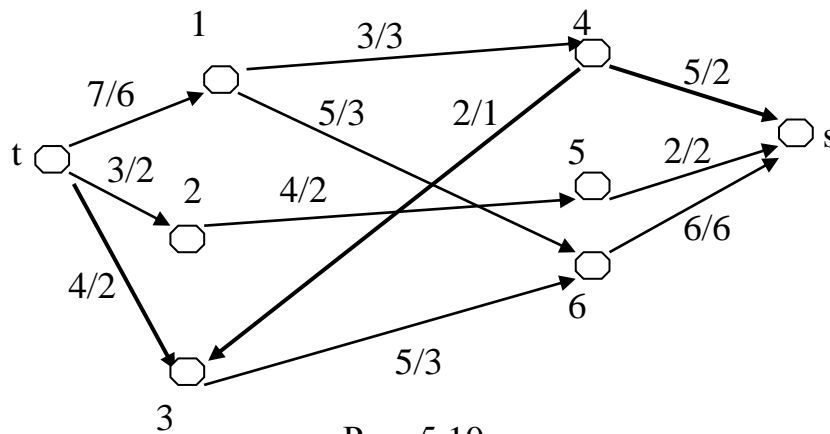


Рис. 5.10

Третья итерация.

Новая цепь состоит из увеличивающих дуг $t, 3$ и $4, s$ и уменьшающей дуги $4, 3$ (рис. 5.10). $\theta_0 = \min(4-2, 1, 5-2)=1$. Изменяем потоки: на дугах $t, 3$ и $4, s$ увеличиваем, а на дуге $4, 3$ уменьшаем на величину θ_0 . Тогда получаем $Z^{(3)} = Z^{(2)} + \theta_0 = 10+1=11$ (рис. 5.11).

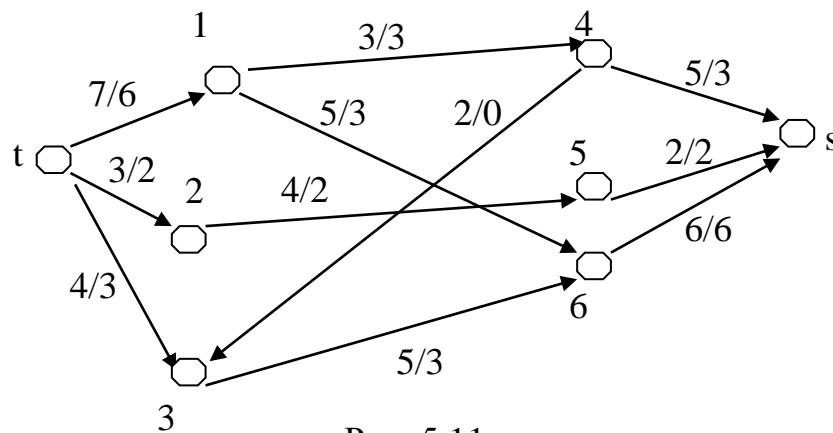


Рис. 5.11

Так как увеличивающую цепь построить нельзя, последнее решение является оптимальным. Максимальный поток сети равен 11.

Минимальный разрез рассмотренной сети соответствует множеству вершин $A = \{t, 1, 2, 3, 5, 6\}$, то есть $P(A) = \{1, 4; 5, s; 6, s\}$. Его пропускная способность $d(A) = d_{14} + d_{5s} + d_{6s} = 3 + 2 + 6 = 11$ равна величине максимального потока, что согласно теореме Форда-Фалкерсона также является признаком оптимальности найденного решения.

5.7. Задания для самостоятельной работы

I. Следующие Т-задачи решить методом потенциалов.

№1.

b_j a_i	10	20	20	10
10	10	15	15	8
20	40	10	30	5
10	35	25	40	10

№2.

b_j a_i	80	30	50	40
70	8	7	4	7
40	3	5	6	4
20	9	2	5	3

№3.

b_j a_i	30	25	18	20
40	1	2	6	4
30	3	1	3	2
20	5	7	5	1

№4.

b_j a_i	20	60	55	45
45	2	5	3	4
35	6	1	2	5
70	3	4	3	8

№5.

b_j a_i	120	40	60	80
180	2	3	4	3
60	5	3	1	2
80	2	1	4	2

№6.

b_j a_i	20	18	44	75
40	1	7	2	5
30	3	8	4	1
50	6	3	5	3

№7.

b_j a_i	10	40	20	60
30	2	7	3	6
70	9	4	5	7
50	5	7	6	2

№8.

b_j a_i	15	40	20	50
48	4	7	2	5
25	3	5	8	6
30	6	10	9	7

№9.

b_j a_i	2	3	3	16
68	18	2	9	7
55	30	4	1	55
40	6	4	8	3

№10.

b_j a_i	10	2	30	40
45	5	4	0	5
20	3	5	3	0
35	0	6	7	6

№11.

b_j a_i	75	80	60	85
100	6	7	3	5
150	1	2	5	6
50	8	10	20	1

№12.

b_j a_i	45	45	100	160
180	6	7	3	2
90	5	1	4	3
170	3	2	6	2

№13.

b_j a_i	25	30	40	15
20	1	3	3	8
20	8	6	2	6
40	7	7	3	8
45	5	2	4	5

№14.

b_j a_i	35	80	25	70
30	1	9	7	2
40	3	1	5	5
70	6	8	3	4
60	2	3	1	3

№15.

b_j a_i	5	15	15	10
15	10	0	20	11
25	12	7	9	20
5	0	14	6	18

№16.

b_j a_i	35	40	40	30
40	3	2	4	1
50	2	3	1	5
30	3	2	4	4

№17.

b_j a_i	30	60	45	25
50	4	7	1	3
70	5	9	6	2
40	8	2	9	11

№18.

b_j a_i	400	800	200	500
500	3	5	6	1
700	5	1	3	3
600	4	5	8	1

№19.

b_j a_i	80	80	60	80
160	5	4	3	4
140	3	2	5	5
60	1	6	3	2

№20.

b_j a_i	20	34	16	10	25
40	2	6	3	4	8
30	1	5	6	9	7
35	3	4	1	6	10

№21.

b_j a_i	80	60	170	80
110	8	1	9	7
190	4	6	2	12
90	3	5	8	9

№22.

b_j a_i	180	110	60	40
175	9	7	5	3
125	1	2	4	6
140	8	10	12	1

№23.

b_j a_i	70	90	60
20	1	5	3
40	1	2	4
50	5	5	1

№24.

b_j a_i	25	30	40	15
10	1	3	3	8
20	8	6	2	6
35	4	7	7	3

Транспортные задачи

40	3	5	2
----	---	---	---

45	5	2	4	5
----	---	---	---	---

№25.

b_j	27	10	30
a_i			
15	8	5	10
35	6	7	9
20	10	12	8
10	5	3	4

№26.

b_j	18	37	50
a_i			
60	4	7	3
42	9	5	13
20	11	8	9
75	7	12	20

№27.

b_j	28	37	50
a_i			
35	4	7	3
42	9	5	13
20	11	8	9
75	7	12	20

II. Методом потенциалов решить T_d -задачи.

№1										
C_{ij}						d_{ij}				
b	5	15	35	15	40					
a_i										
30	18	3	8	5	14	15	7	21	22	6
50	5	19	12	5	9	2	6	18	5	25
20	17	20	11	10	9	3	4	1	7	8
10	2	15	12	13	6	9	5	3	3	7

№2											№3										
C_{ij}						d_{ij}					C_{ij}						d_{ij}				
b a_i	40	40	20	10	30						b a_i	11	10	14	16	17					
14	10	16	3	8	15	8	2	3	2	2	10	17	18	7	20	2	7	10	4	5	8
25	3	14	12	9	1	15	3	3	4	5	13	5	16	3	9	19	5	4	6	5	9
56	2	20	4	11	5	10	20	15	5	10	28	15	14	17	7	5	11	6	12	8	7
45	7	17	13	8	15	11	20	7	4	21	17	11	15	9	8	16	3	2	1	6	10

№4											№5										
C_{ij}						d_{ij}					C_{ij}						d_{ij}				
b a_i	13	37	26	91	24						b a_i	24	30	27	16	33					
20	1	14	13	6	12	3	7	6	5	5	15	10	16	12	3	9	10	5	3	5	4
30	3	18	20	4	2	4	10	8	11	2	17	10	4	16	1	11	4	7	5	5	10
70	8	17	5	15	10	7	20	15	38	9	23	19	19	10	18	20	3	12	7	6	4
71	16	4	12	18	13	14	11	13	41	20	75	19	12	4	11	18	20	10	17	15	35

№6											№7										
C_{ij}						d_{ij}					C_{ij}						d_{ij}				
b a_i	30	50	13	70	17						b a_i	11	10	14	16	17					
62	5	8	13	14	9	20	25	10	20	2	17	11	15	9	8	16	3	2	1	6	10
26	11	20	18	19	17	4	7	6	10	15	10	17	18	7	20	2	7	10	4	5	8
27	1	15	2	8	3	5	6	12	20	30	28	15	14	17	7	5	11	6	12	8	7
65	9	16	15	11	17	10	30	15	25	5	13	5	16	3	9	19	5	4	6	5	9

№8											№9										
C_{ij}						d_{ij}					C_{ij}						d_{ij}				
b a_i	40	10	50	50	20						b a_i	30	35	35	45	25					
60	20	5	4	5	14	14	5	20	25	13	30	9	10	12	7	1	4	10	15	8	10
50	2	19	20	4	5	15	19	22	4	12	70	4	16	9	19	10	21	20	23	16	15
40	8	9	6	19	1	15	6	8	5	10	50	17	20	11	1	12	10	5	8	25	12
20	20	7	15	6	11	2	14	7	23	20	20	14	13	15	7	8	5	4	6	7	10

№10											№11										
C_{ij}						d_{ij}					C_{ij}						d_{ij}				
b a_i	75	10	20	40	30						b a_i	5	15	35	15	40					
80	7	19	7	12	18	20	6	15	22	25	50	5	19	12	5	9	2	6	18	5	25
12	17	11	7	13	11	2	5	2	3	4	20	17	20	11	10	9	3	4	1	7	8
38	1	13	19	18	12	20	1	3	15	8	10	2	15	12	13	6	9	5	3	3	7
45	18	4	11	3	11	40	5	6	2	10	30	18	3	8	5	14	15	7	21	22	6

Транспортные задачи

№12											№13										
C_{ij}						d_{ij}					C_{ij}						d_{ij}				
b a_i	37	26	91	24	13						b a_i	45	38	40	28	34					
20	14	13	6	12	1	7	6	5	5	3	20	3	17	6	19	2	5	7	6	5	4
30	18	20	4	2	3	10	8	11	2	4	40	1	15	7	6	1	20	3	20	4	2
70	17	5	15	10	8	20	15	38	9	7	52	5	13	8	11	17	24	13	7	11	8
71	4	12	18	13	16	11	13	41	20	14	73	18	13	17	1	8	9	21	16	13	25

№14											№15										
C_{ij}						d_{ij}					C_{ij}						d_{ij}				
b a_i	10	50	40	50	20						b a_i	30	27	16	33	24					
60	5	4	20	5	14	5	20	14	25	13	15	16	12	3	9	10	5	3	5	4	10
50	19	20	2	4	5	19	22	15	4	12	17	4	16	1	11	10	7	5	5	10	4
40	9	6	8	19	1	6	8	15	5	10	23	19	10	18	20	19	12	7	6	4	3
20	7	15	20	6	11	14	7	2	23	20	75	12	4	11	18	19	10	17	15	35	20

№16											№17										
C_{ij}						d_{ij}					C_{ij}						d_{ij}				
b a_i	35	30	35	45	25						b a_i	17	33	38	21	22					
30	10	9	12	7	1	10	4	15	8	10	48	8	12	6	12	5	20	19	17	3	6
70	16	4	9	19	10	20	21	23	16	15	27	3	1	12	2	19	10	14	30	18	7
50	20	17	11	1	12	5	10	8	25	12	46	6	3	3	1	18	18	16	11	13	17
20	13	14	15	7	8	4	5	6	7	10	10	5	3	20	7	25	9	1	16	5	11

№18											№19										
C_{ij}						d_{ij}					C_{ij}						d_{ij}				
b a_i	13	26	91	24	37						b a_i	75	10	20	40	30					
20	1	13	6	12	14	3	6	5	5	7	80	7	19	7	12	18	20	6	15	22	25
30	3	20	4	2	18	4	8	11	2	10	38	1	13	19	18	12	20	1	3	15	8
70	8	5	15	10	17	7	15	38	9	20	12	17	11	7	13	11	2	5	2	3	4
71	16	12	18	13	4	14	13	41	20	11	45	18	4	11	3	11	40	5	6	2	10

№20											№21										
C_{ij}						d_{ij}					C_{ij}						d_{ij}				
b a_i	30	65	20	10	15						b a_i	15	30	65	20	10					
50	10	2	5	10	14	35	14	10	5	15	50	14	10	2	5	10	15	35	14	10	5
20	5	4	11	3	11	4	20	12	18	8	20	11	5	4	11	3	8	4	20	12	18
30	8	12	1	18	9	7	32	20	14	10	30	9	8	12	1	18	10	7	32	20	14
40	4	9	17	18	1	4	16	20	17	8	40	1	4	9	17	18	8	4	16	20	17

Глава 6. БЛОЧНОЕ ПРОГРАММИРОВАНИЕ

При решении задач большой размерности значительная часть времени тратится на обращения к внешней памяти и это является главным препятствием на пути увеличения размерности задач. Уменьшить число обращений к внешней памяти можно, если удастся большую задачу заменить рядом задач существенно меньшей размерности. Приемы и методы, позволяющие выполнять такие преобразования, составляют предмет блочного программирования.

Одним из эффективных методов блочного программирования применительно к линейным задачам является метод декомпозиции Данцига – Вулфа. По данным авторов метод позволяет решать задачи с размерностью $n \sim 10^6$, $m \sim 10^5$.

6.1. Метод декомпозиции Данцига - Вулфа

Сначала рассмотрим математические преобразования, приводящие к разбиению исходной задачи, а затем покажем, в каких случаях это дает эффект по сравнению с непосредственным решением большой задачи.

Пусть имеется следующая модель задачи:

$$L = C^T X \rightarrow \max; \quad (6.1)$$

$$AX = B; \quad (6.2)$$

$$X \geq 0, \quad (6.3)$$

где вектор X имеет размерность n , а вектор B – m .

Условия (6.2), (6.3) определяют допустимое множество задачи D . Представим матрицу A и вектор B в виде двух подматриц:

$$A = \begin{bmatrix} A^{(0)} \\ A^{(1)} \end{bmatrix};$$

$$B = \begin{bmatrix} B^{(0)} \\ B^{(1)} \end{bmatrix}.$$

Тогда условия задачи (6.2)-(6.3) записываются следующим образом:

$$A^{(0)} X = B^{(0)}; \quad (6.4)$$

$$A^{(1)} X = B^{(1)}; \quad (6.5)$$

$$X \geq 0 \quad (6.6)$$

Условия (6.4), включающие m_0 равенств, порождают допустимое множество D_0 , а система (6.5) содержит m_1 равенств и вместе с (6.6) задает множество D_1 . Очевидно, что $m = m_0 + m_1$, $D = D_0 \cap D_1$. При этом выделение подматриц выполняется так, что $m_1 \gg m_0$.

Далее будем полагать, что множество D_I ограниченное и, значит, является выпуклым многогранником. В противном случае его легко сделать ограниченным добавлением ограничений сверху на переменные так, что они не повлияют на исходное множество D .

Предположим, что нам известны вершины множества D_I . Обозначим их координаты через X^1, X^2, \dots, X^N , где N – число вершин. Поскольку D_I – выпуклый многогранник, то любую его точку можно представить в виде линейной комбинации вершин:

$$X = \sum_{v=1}^N z_v X^v; \quad (6.7)$$

$$\sum z_v = 1; \quad (6.8)$$

$$z_v \geq 0, \forall v. \quad (6.9)$$

Так как все решения X , определяемые по (6.7)-(6.9), принадлежат D_I , то описание (6.7)-(6.9) эквивалентно (6.5), (6.6).

Подставим X в виде (6.7) в (6.1) и (6.4):

$$L = \sum_{v=1}^N C^T z_v X^v;$$

$$\sum A^{(0)} X^v z_v = B^{(0)}.$$

Считая X^v известными, введем обозначения:

$$C^T X^v = \sigma_v; \quad (6.10)$$

$$A^{(0)} X^v = P_v. \quad (6.11)$$

Тогда преобразованная модель задачи запишется в виде

$$L = \sum_v \sigma_v z_v \rightarrow \max;$$

$$\sum_v P_v z_v = B^{(0)};$$

$$\sum_v z_v = 1;$$

$$\forall z_v \geq 0.$$

В этой модели неизвестными являются z_v , число которых равно числу вершин многогранника D_I . Последнее равенство модели можно объединить со всеми остальными, используя обозначения расширенных векторов

$$\bar{P}_v = \begin{bmatrix} P_v \\ 1 \end{bmatrix}; \quad \bar{B}^{(0)} = \begin{bmatrix} B^{(0)} \\ 1 \end{bmatrix}. \quad (6.12)$$

Тогда окончательно получим:

$$L = \sum_v \sigma_v z_v \rightarrow \max; \quad (6.13)$$

$$\sum_v \bar{P}_v z_v = \bar{B}^{(0)}; \quad (6.14)$$

$$\forall z_v \geq 0. \quad (6.15)$$

Задача в виде (6.13) – (6.15) называется *координирующей* или *основной* задачей. Главное отличие этой задачи от исходной в несравнимо меньшем числе условий ($m_0+1 \ll m$).

Если мы сможем ее решить, то есть найти \mathbf{Z}^* , то получим решение и исходной задачи, воспользовавшись (6.7):

$$\mathbf{X}^* = \sum_{v=1}^N z_v^* \mathbf{X}^v. \quad (6.16)$$

Для решения основной задачи применим модифицированный симплекс-метод. Начальное решение можно построить, не зная ни одной вершины, с помощью искусственных переменных z_{N+i} .

Согласно модифицированному методу после получения очередного базисного решения вычисляются относительные оценки. В разд. 4.10 получены формулы:

$$\Delta_j = \boldsymbol{\pi}^T \mathbf{A}_j - C_j, \quad \boldsymbol{\pi}^T = \mathbf{C}_b^T \mathbf{A}_b^{-1}.$$

Перепишем их в обозначениях координирующей задачи:

$$\bar{\boldsymbol{\pi}}^1 = \boldsymbol{\sigma}_b^T \bar{\mathbf{P}}_b^{-1}; \quad (6.17)$$

$$\Delta_v = \bar{\boldsymbol{\pi}}^T \bar{\mathbf{P}}_v - \sigma_v = \bar{\boldsymbol{\pi}}^T \begin{bmatrix} \mathbf{A}^{(0)} \mathbf{X}^v \\ \mathbf{1} \end{bmatrix} - \mathbf{C}^T \mathbf{X}^v = \boldsymbol{\pi}^T \mathbf{A}^{(0)} \mathbf{X}^v + \pi_{m_0+1} - \mathbf{C}^T \mathbf{X}^v$$

или окончательно

$$\Delta_v = (\boldsymbol{\pi}^T \mathbf{A}^{(0)} - \mathbf{C}^T) \mathbf{X}^v + \pi_{m_0+1}. \quad (6.18)$$

Мы не можем вычислить все оценки, так как нам не известно даже их число. Но этого и не требуется, достаточно только определить: есть или нет среди них отрицательные. Для ответа на этот вопрос будем искать наименьшую оценку. Если она отрицательная, текущее решение координирующей задачи может быть улучшено введением переменной с этой оценкой. В противном случае констатируется выполнение признака оптимальности.

Итак, задача состоит в следующем:

$$\Delta_v \rightarrow \min.$$

Отбросив в (6.18) константу, запишем ее в виде

$$(\boldsymbol{\pi}^T \mathbf{A}^{(0)} - \mathbf{C}^T) \mathbf{X}^v \rightarrow \min_{\mathbf{X}^v}. \quad (6.19)$$

Решение задачи (6.19) проблематично, так как минимум ищется на дискретном множестве вершин многогранника \mathbf{D}_I . Учитывая, что минимизируемая функция линейная, будем искать решение не на вершинах, а на всем многограннике. Известно, что если решение существует, то оно будет достигаться в вершине. Поэтому решение на всем (непрерывном) множестве \mathbf{D}_I совпадет с решением задачи (6.19).

Таким образом, задачу (6.19) заменяем эквивалентной:

$$L_{всп} = (\pi^T A^{(0)} - C^T)X \rightarrow \min_x; \quad (6.20)$$

$$A^{(1)}X = B^{(1)}; \quad (6.21)$$

$$X \geq 0. \quad (6.22)$$

Эта задача называется *вспомогательной*. Если она неразрешима, то и исходная задача не имеет решения. Пусть оптимальное решение вспомогательной задачи (6.20)-(6.22) достигается в вершине r . Это означает, что нам становятся известны координаты вершины X^r и оптимальное значение критерия $L_{всп}^*$. Тогда согласно формуле (6.18) вычисляем минимальную оценку

$$\Delta_r = \min \Delta_v = (\pi^T A^{(0)} - C^T)X^r + \pi_{m_0+1} = L_{всп}^* + \pi_{m_0+1}. \quad (6.23)$$

Очевидно, что если $\Delta_r \geq 0$, то и все оценки неотрицательны, и решение координирующей задачи завершено. При отрицательной Δ_r решение продолжается. В базис основной задачи вводится вектор \bar{P}_r , определяемый по формуле

$$\bar{P}_r = \begin{bmatrix} A^{(0)}X^r \\ 1 \end{bmatrix}. \quad (6.30)$$

Направляющий столбец находится разложением этого вектора по текущему базису:

$$\alpha_r = \bar{P}_B^{-1} \bar{P}_r. \quad (6.31)$$

После определения направляющего элемента и симплекс-преобразования получаем новое решение основной задачи. Коэффициент критерия (6.13) при переменной, введенной в базисное решение, вычисляется согласно (6.10):

$$\sigma_r = C^T X^r. \quad (6.32)$$

Теперь по формуле (6.17) находим новый вектор $\bar{\pi}^T$, снова решаем вспомогательную задачу и по полученной минимальной оценке делаем вывод о дальнейших действиях.

Таким образом, решение исходной задачи заменяется многократным решением основной и вспомогательной задач. При этом порядок размерности вспомогательной задачи такой же, как у исходной. Поэтому естественен вопрос: в каких случаях такой метод эффективен?

Ответ очевиден: в тех случаях, когда сложность решения вспомогательной задачи намного ниже, чем исходной. Такие случаи имеют место, когда матрица условий задачи (после упорядочения строк и столбцов) оказывается почти-блочно-диагональной, как показано на рис. 6.1. Примером может служить задача планирования производства продукции в крупной фирме или холдинге, когда у каждого предприятия своя номенклатура продукции, а некоторые ресурсы являются общими.

Подматрица $\mathbf{A}^{(0)}$, входящая в параметры координирующей задачи, соответствует ограничениям по общим ресурсам. Такие условия называют *связующими*. Их относят к основной задаче.

$$\mathbf{A} = \left(\begin{array}{c} \boxed{\text{Связующий блок } \mathbf{A}^{(0)}} \\ \boxed{\mathbf{A}_1^{(1)}} \quad \dots \quad \boxed{\mathbf{A}_p^{(1)}} \end{array} \right) \left\{ \begin{array}{l} \text{общие ресурсы} \\ \mathbf{A}^{(1)} - \text{матрица} \\ \text{вспомогательной задачи} \end{array} \right.$$

Рис. 6.1

Остальные условия образуют вспомогательную задачу. При этом подматрица $\mathbf{A}^{(1)}$ имеет блочно-диагональную структуру, что позволяет разбить вспомогательную задачу на p независимых задач:

$$\begin{aligned} L_i &= \mathbf{C}_i^T \mathbf{X}_i; \\ \mathbf{A}_i^{(1)} \mathbf{X}_i &= \mathbf{B}_i^{(1)}; \\ \mathbf{X}_i &\geq 0, \quad i = \overline{1, p}. \end{aligned}$$

После решения этих задач определяется критерий вспомогательной задачи по очевидной формуле

$$L_{всп}^* = \sum L_i^*.$$

Таким образом, решение вспомогательной задачи существенно упрощается, если структура матрица условий может быть приведена к блочно-диагональной.

В следующем разделе декомпозиция вспомогательной задачи будет показана на примере решения транспортной задачи.

Применение рассмотренного метода может быть целесообразно и тогда, когда вспомогательная задача имеет особенности, позволяющие решать ее специальными методами.

6.2. Решение транспортной задачи методом Данцига-Вулфа

Применим метод декомпозиции к Т-задаче:

$$L = \sum_i \sum_j C_{ij} X_{ij} \rightarrow \min; \quad (6.33)$$

$$\sum_j X_{ij} = a_i, \quad i = \overline{1, m}; \quad (6.34)$$

$$\sum_i X_{ij} = b_j, \quad j = \overline{1, n}; \quad (6.35)$$

$$\forall X_{ij} \geq 0. \quad (6.36)$$

Использование этого метода целесообразно, если $m \ll n$ или $m \gg n$. Оба варианта решаются идентично. Они отличаются только распределением условий между основной и вспомогательной задачами.

Рассмотрим случай, когда $m \ll n$. Тогда основная задача формируется по условиям пунктов отправления. Следовательно, множество D_0 описывается ограничениями (6.34), а D_I – условиями (6.35) и (6.36).

Очевидно, что множество D_I представляет собой выпуклый многогранник (ограниченность вытекает из условий). Поэтому, как и в общем случае, любую точку в D_I можно представить в виде линейной комбинации его вершин:

$$X_{ij} = \sum_{v=1}^N X_{ij}^v Z_v; \quad (6.37)$$

$$\sum Z_v = 1; \quad (6.38)$$

$$\forall Z_v \geq 0, \quad (6.39)$$

где X_{ij}^v – координаты v -ой вершины.

Подставим (6.37) в (6.33) и (6.34):

$$L = \sum_i \sum_j \sum_v C_{ij} X_{ij}^v Z_v;$$

$$\sum_j \sum_v X_{ij}^v Z_v = a_i.$$

Введем обозначения:

$$\sigma_v = \sum_i \sum_j C_{ij} X_{ij}^v; \quad (6.40)$$

$$P_{iv} = \sum_j X_{ij}^v. \quad (6.41)$$

Тогда основная задача запишется в виде

$$L = \sum_v \sigma_v Z_v \rightarrow \min; \quad (6.42)$$

$$\sum_v P_{iv} Z_v = a_i, \quad i = \overline{1, m}; \quad (6.43)$$

$$\sum_v Z_v = 1; \quad (6.44)$$

$$\forall Z_v \geq 0. \quad (6.45)$$

Для сбалансированной задачи условие (10) выполняется автоматически. Действительно, суммируя (6.43) и используя подстановки (6.41) и (6.35), получаем

в левой части

$$\sum_i \sum_v P_{iv} Z_v = \sum_i \sum_v \sum_j X_{ij}^v Z_v = \sum_v \sum_j \sum_i X_{ij}^v Z_v = \sum_v Z_v \sum_j b_j,$$

в правой части $\sum_i a_i$. Таким образом,

$$\sum_v Z_v \sum_j b_j = \sum_i a_i,$$

откуда для сбалансированной задачи следует

$$\sum_v Z_v = 1.$$

Поэтому при решении основной задачи условие (6.44) из модели исключается.

Для определения статуса текущего базисного решения основной задачи необходимы относительные оценки. Как и в предыдущем разделе, нахождение оценок связано с решением вспомогательной задачи. Для построения вспомогательной задачи сделаем ряд преобразований:

$$\begin{aligned}\Delta_v &= \pi^T P_v - \sigma_v = \\ &= \sum_i \pi_i P_{iv} - \sum_i \sum_j C_{ij} X_{ij}^v = \\ &= \sum_j \sum_i \pi_i X_{ij}^v - \sum_j \sum_i C_{ij} X_{ij}^v = \\ &= \sum_j \sum_i (\pi_i - C_{ij}) X_{ij}^v.\end{aligned}$$

Так как основная задача решается на минимум, то оптимальному статусу соответствуют неположительные оценки. Поэтому нужно искать максимальную оценку. Если она окажется не больше нуля, то все оценки неположительны и признак оптимальности выполнен. В противном случае необходимо продолжить решение основной задачи.

Значит, задача ставится так:

$$\Delta_v = \sum_j \sum_i (\pi_i - C_{ij}) X_{ij}^v \rightarrow \max.$$

Вместо поиска максимума на дискретном множестве вершин X^v перейдем к эквивалентной задаче поиска на всем непрерывном множестве D_I :

$$L_{всп} = \sum_j \sum_i (\pi_i - C_{ij}) X_{ij} \rightarrow \max; \quad (6.46)$$

$$\sum_i X_{ij} = b_j, \quad j = \overline{1, n}; \quad (6.47)$$

$$\forall X_{ij} \geq 0. \quad (6.48)$$

Эта задача и является вспомогательной. Очевидно, что в оптимальном решении этой задачи $L_{всп}^* = \max \Delta_v$. Теперь остается выяснить, как найти его.

Вспомогательная задача включает одну группу условий (6.47). Раньше было показано, что каждая переменная входит в такие условия только один раз. Поэтому равенства (6.47) оказываются независимыми и, следовательно, вспомогательная задача распадается на n простейших независимых задач, каждая из которых имеет всего одно условие:

$$L_j = \sum_i (\pi_i - C_{ij}) X_{ij} \rightarrow \max; \quad (6.49)$$

$$\sum_{i=1}^m X_{ij} = b_j; \quad (j = \text{const}) \quad (6.50)$$

$$\forall X_{ij} \geq 0. \quad (6.51)$$

Критерий вспомогательной задачи равен сумме критериев этих задач:

$$L_{всп} = \sum_j L_j. \quad (6.52)$$

Оптимальное решение задачи (6.49)-(6.51), как линейной, находится на границе. При этом только одна переменная не равна нулю (базис имеет размерность 1). Поэтому ее решение состоит в определении максимального коэффициента в критерии (6.49). Пусть максимум достигается на индексе i^* , то есть

$$\max_i (\pi_i - C_{ij}) = \pi_{i^*} - C_{i^*j}.$$

Тогда имеем следующее решение задачи (6.49)-(6.51):

$$X^v_{i^*j} = b_j, \quad X^v_{ij} = 0, \quad \forall i, i \neq i^*, \quad (6.53)$$

и максимальная оценка определится как

$$L^*_{всп} = \sum_{j=1}^n L^*_j.$$

Если $L^*_{всп} \leq 0$, то положительных оценок нет и текущее решение основной задачи будет оптимальным.

При $L^*_{всп} > 0$ начинается новая итерация:

1. по (6.41) и (6.40) находим \mathbf{P}_v и σ_v ;
2. вычисляем элементы направляющего столбца как коэффициенты разложения вектора \mathbf{P}_v по текущему базису:

$$\alpha_v = \mathbf{P}^{-1}_B \mathbf{P}_v;$$

3. проводим симплекс-преобразование основной задачи, в результате которого получаем новое решение и новую обратную матрицу;

4. вычисляем $\boldsymbol{\pi}^T = \boldsymbol{\sigma}^T_B \mathbf{P}^{-1}_B$;

5. решаем вспомогательную задачу: вычисляем разности $\pi_i - C_{ij}$, находим оптимальные решения n задач (6.49)-(6.51) и максимальную оценку основной задачи.

Из рассмотренной вычислительной схемы следует, что эффективность метода тем выше, чем сильнее неравенство $m \ll n$ или $m \gg n$.

Пример.

Решим транспортную задачу с двумя пунктами отправления и четырьмя пунктами назначения:

b_i	8	4	10	8
a_i				
10	2	5	1	4
20	1	3	4	2

Числа в ячейках таблицы - затраты на перевозки C_{ij} .

Исходная модель задачи:

$$L = \sum \sum C_{ij} X_{ij} \rightarrow \min$$

$$\left. \begin{array}{l} \sum_j X_{1j} = 10 \\ \sum_j X_{2j} = 20 \end{array} \right\} \quad (6.54)$$

$$\left. \begin{array}{l} \sum_i X_{i1} = 8 \\ \sum_i X_{i2} = 4 \\ \sum_i X_{i3} = 10 \\ \sum_i X_{i4} = 8 \\ \forall X_{ij} \geq 0 \end{array} \right\} \quad (6.55)$$

Координирующая задача формируется по условиям (6.54):

$$L = \sum_v \sigma_v Z_v \rightarrow \min;$$

$$\sum_{v=1}^n P_{1v} Z_v = 10;$$

$$\sum_{v=1}^n P_{2v} Z_v = 20;$$

$$\forall Z_v \geq 0.$$

Для построения начального решения вводим искусственные переменные:

$$\sum_{v=1}^n P_{1v} Z_v + Z_{n+1} = 10;$$

$$\sum_{v=1}^n P_{2v} Z_v + Z_{n+2} = 20,$$

и модифицируем критерий

$$L = \sum_v \sigma_v Z_v + M(Z_{n+1} + Z_{n+2}) \rightarrow \min.$$

Составим начальную таблицу координирующей задачи:

σ_v	Базисные перемен.	P_0	P_{n+1}	P_{n+2}
M	Z_{n+1}	10	1	0
M	Z_{n+2}	20	0	1
π^T			M	M

В последней строке значения π_i получены умножением первого столбца на столбцы P_{n+i} .

Решение вспомогательной задачи представляем в таблице:

b_j	8	4	10	8
$\pi_1 - C_{1j}$	$M-2$	$M-5$	$M-1$	$M-4$
$\pi_2 - C_{2j}$	$M-1$	$M-3$	$M-4$	$M-2$
$v=1$	$X^1_{21}=\mathbf{8}$	$X^1_{22}=\mathbf{4}$	$X^1_{13}=\mathbf{10}$	$X^1_{24}=\mathbf{8}$

Значения переменных в последней строке таблицы получены согласно (6.53). Например, при $j=1$ максимальная разность равна $M-1$, поэтому $X^1_{21}=b_1=8$. Клетки с максимальными разностями выделены цветом фона. Вычисляем значение критерия по формуле (6.46):

$$L^1_{\text{сн}} = \sum_{j=1}^4 L_j = [(M-1)*8 + (M-3)*4 + (M-1)*10 + (M-2)*8] > 0.$$

Так как признак оптимальности не выполняется, переходим к итерациям. Находим σ_l согласно (6.40):

$$\sigma_1 = 1*8 + 3*4 + 1*10 + 2*8 = 46.$$

Вычисляем компоненты вектора \mathbf{P}_1 :

$$P_{11} = X^1_{13} = 10;$$

$$P_{21} = X^1_{21} + X^1_{22} + X^1_{24} = 8 + 4 + 8 = 20.$$

Следовательно, $\mathbf{P}_1 = \begin{bmatrix} 10 \\ 20 \end{bmatrix}$. Находим его разложение по начальному базису:

$$\alpha_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 10 \\ 20 \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \end{bmatrix}.$$

Добавляем столбец \mathbf{P}_1 с элементами α_l в начальную таблицу в качестве направляющего столбца:

σ_v	Базисные перемен.	\mathbf{P}_0	\mathbf{P}_{n+1}	\mathbf{P}_{n+2}	\mathbf{P}_1	θ
M	Z_{n+1}	10	1	0	10	1
M	Z_{n+2}	20	0	1	20	1
π^T			M	M		

Взяв 1-ю строку за направляющую и выполнив симплекс-преобразование, получаем новое решение основной задачи:

σ_v	Баз.	\mathbf{P}_0	\mathbf{P}_{n+1}	\mathbf{P}_{n+2}
46	Z_1	1	0,1	0
M	Z_{n+2}	0	-2	1
π^T			$-2M+4,6$	M

Для выяснения статуса этого решения снова находим максимальную оценку основной задачи, решая вспомогательную задачу:

b_j	8	4	10	8
$\pi_1 - C_{1j}$	$2,6-2M$	$-0,4-2M$	$3,6-2M$	$0,6-2M$
$\pi_2 - C_{2j}$	$M-1$	$M-3$	$M-4$	$M-2$
$\nu=2$	$X^2_{21}=8$	$X^2_{22}=4$	$X^2_{23}=10$	$X^2_{24}=8$

Очевидно, что $L^2_{\text{всп}} > 0$, то есть решение основной задачи не является оптимальным.

Вычисляем коэффициент критерия при Z_2 :

$$\sigma_2 = 1 \cdot 8 + 3 \cdot 4 + 4 \cdot 10 + 2 \cdot 8 = 8 + 12 + 40 + 16 = 76.$$

Определяем компоненты вектора P_2 :

$$P_{12}=0, P_{22}=8+4+10+8=30$$

Имея $P_2 = \begin{bmatrix} 0 \\ 30 \end{bmatrix}$, находим элементы направляющего столбца

$$a_2 = \begin{bmatrix} 0,1 & 0 \\ -2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 30 \end{bmatrix} = \begin{bmatrix} 0 \\ 30 \end{bmatrix}$$

и добавляем его к последней таблице основной задачи:

σ_v	Баз.	P_0	P_{n+1}	P_{n+2}	P_2	θ
46	Z_1	1	0,1	0	0	-
M	Z_{n+2}	0	-2	1	30	0
π^T			$-2M+4,6$	M		

В результате симплекс-преобразования получаем:

σ_v	Баз	P_0	P_{n+1}	P_{n+2}
46	Z_1	1	0,1	0
76	Z_2	0	-1/15	1/30
π^T			-7/15	38/15

Соответствующая вспомогательная задача:

b_j	8	4	10	8
$\pi_1 - C_{1j}$	-37/15	-82/15	-22/15	-67/15
$\pi_2 - C_{2j}$	23/15	-7/15	-22/15	8/15
$\nu=3$	$X^3_{21}=8$	$X^3_{22}=4$	$X^3_{13}=10$	$X^3_{24}=8$

Критерий этой задачи $L^3_{\text{всп}} = (23/15) \cdot 8 - (7/15) \cdot 4 - (22/15) \cdot 10 + (8/15) \cdot 8 = 0$. Следовательно, получено оптимальное решение основной задачи: $Z^*_1=1$, $Z^*_2=0$, $L^* = 46 \cdot 1 + 76 \cdot 0 = 46$.

Находим значения исходных переменных по формуле (6.37), которая для нашей задачи принимает вид:

$$X^*_{ij} = X^1_{ij} Z^*_1 + X^2_{ij} Z^*_2 = X^1_{ij}.$$

Таким образом, получено следующее оптимальное решение исходной задачи: $X_{21}^* = 8$, $X_{22}^* = 4$, $X_{13}^* = 10$, $X_{24}^* = 8$.

Проверка: $L = \sum \sum C_{ij} X_{ij} = 1 \cdot 8 + 3 \cdot 4 + 1 \cdot 10 + 2 \cdot 8 = 46$. Это значение совпадает с вычисленным через переменные Z_i .

6.3. Задания для самостоятельной работы

Транспортные задачи решить методом декомпозиции Данцига-Вулфа.

1

b_j	20	10	15	5
a_i				
12	7	4	8	2
30	5	3	6	1
8	4	9	5	3

2

b_j	10	8	15	20
a_i				
22	3	1	4	5
13	2	5	3	4
18	6	2	5	7

3

b_j	7	9	9	8
a_i				
13	2	4	3	2
11	5	1	6	4
9	3	2	4	3

4

b_j	90	40	60	20
a_i				
100	2	3	4	3
50	5	3	1	2
60	2	1	4	2

5

b_j	9	11	10	5
a_i				
6	4	1	5	2
14	2	3	6	1
15	3	5	7	2

6

b_j	120	40	50	30
a_i				
160	2	3	4	3
60	5	3	1	2
80	2	1	4	2

7

b_j	10	8	15	20
a_i				
16	3	1	4	5
19	2	5	3	4
18	6	2	5	3

8

b_j	9	11	10	5
a_i				
16	4	8	5	2
4	2	3	6	1
15	3	5	7	4

9

b_j	65	40	50	30
a_i				
80	4	2	3	1
35	6	3	5	6
70	3	2	6	3

10

b_j	80	50	50	70
a_i				
80	4	2	3	1
100	6	3	5	6
70	3	2	6	3

11

b_j	9	11	10	5
a_i				
16	4	1	5	2
4	2	3	6	1
15	3	5	7	2

12

b_j	6	10	5	7
a_i				
4	3	1	2	4
11	5	4	6	2
13	1	7	5	3

13

b_j	20	10	15	5
a_i				
25	7	4	8	2
17	9	3	6	1
8	4	6	5	3

14

b_j	9	12	10	7
a_i				
16	4	8	5	2
7	2	3	6	1
15	3	5	7	1

15

b_j	4	10	5	7
a_i				
8	2	6	3	5
15	4	7	4	2
3	1	3	2	5

16

b_j	5	15	15	10
a_i				
15	10	0	20	11
25	12	7	9	20
5	0	14	16	18

Глава 7. ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ

Целочисленное программирование (ЦП) – это наиболее важный раздел дискретного программирования. Задачи дискретного программирования отличаются тем, что на переменные накладывается требование дискретности, в частном случае – целочисленности. В качестве примеров можно привести задачи о раскрое (разд. 4.11.5), о ранце, коммивояжера и др.

Характерные источники целочисленности (дискретности):

1. неделимость объектов, представляемых переменными (например, x – число рабочих или отправляемых вагонов);
2. вариантность типа “да-нет” (например, включать или нет данный пакет в портфель ценных бумаг);
3. заданность возможных значений нормативными документами (например, сечения проводов, диаметров труб, размеров профилей и т.п.);
4. комбинаторность (например, размещение объектов, порядок обхода объектов, упорядочение);
5. логические условия (например, фиксированные затраты имеют место только при производстве продукции).

Различают задачи полностью целочисленные/дискретные и частично целочисленные (смешанные). В последних требование целочисленности накладывается не на все переменные.

Любой ряд дискретных значений может быть представлен линейной комбинацией целочисленных переменных. Поэтому дискретная задача легко сводится к целочисленной за счет значительного увеличения числа переменных.

В этой главе рассматриваются только линейные целочисленные задачи.

7.1. Проблема целочисленности

На первый взгляд может показаться, что целочисленная задача решается проще непрерывной. При малой размерности и узких диапазонах переменных задачу можно решить простым перебором. В других случаях необходимы соответствующие методы.

Несмотря на линейность модели допустимое множество целочисленной задачи не является выпуклым. Так, в полностью целочисленной задаче оно представляет собой множество отдельных

точек, имеющих целочисленные координаты. Методы линейного
программирования базируются на

выпуклости допустимого множества и поэтому непосредственно не могут быть применимы к целочисленным задачам.

Можно, конечно, пренебречь требованием целочисленности и использовать один из методов ЛП, но тогда, за редким исключением, результат не будет целочисленным. Округление дробных значений переменных проблематично. Действительно, так как оптимальное решение непрерывной задачи лежит в вершине допустимого множества, округление может привести к недопустимости. При двух дробных переменных имеется 4, а при n переменных – 2^n вариантов округления! Какие из них дают допустимые решения, можно определить только после проверки всех ограничений. При этом следует иметь в виду, что, во-первых, целочисленная задача может оказаться неразрешимой несмотря на разрешимость непрерывной задачи; во-вторых, допустимость округленного решения еще не означает его оптимальность. Проиллюстрируем последний тезис известной задачей о садовнике.

По расчетам садовника требуется внести в почву комплексные удобрения в количестве 107 кг. Удобрения продаются только в расфасованном виде: 1) мешок весом 35 кг стоит 140 руб.; 2) мешок весом 24 кг стоит 120 руб. Необходимо определить вариант закупки удобрений с минимальными затратами.

Запишем модель задачи:

$$\begin{aligned} L &= 140x_1 + 120x_2 \rightarrow \min; \\ 35x_1 + 24x_2 &\geq 107; \\ x_1, x_2 &\geq 0, \text{ int (целые)}. \end{aligned}$$

Если пренебречь целочисленностью, то легко увидеть, что оптимальным будет решение

$$x_1 = 3\frac{2}{35}, x_2 = 0.$$

Если округлить его по правилам арифметики, то получим недопустимое решение. Округление в большую сторону ($x_1=4, x_2=0$) приводит к допустимости, но является ли такое решение оптимальным?

Чтобы ответить на этот вопрос, рассмотрим все возможные целочисленные решения (табл.7.1). Прочерк в графе критерия означает недопустимость.

Таблица 7.1

x_1	x_2	L
3	0	-
4	0	560
3	1	540
2	2	520
1	3	500
0	4	-
0	5	600

Как видно из таблицы, оптимальным является решение $x_1^*=1, x_2^*=3$, которое принципиально отличается от непрерывного: закупать надо в основном мешки 2-го типа, тогда как по непрерывному решению – только 1-го типа. Кроме того, округленное допустимое решение оказалось

далеки́м от оптимального: затраты в нем выше минимальных на 12%. ▲

Другую особенность свойств целочисленных задач покажем также на конкретном примере:

$$L = 21x_1 + 11x_2 \rightarrow \max;$$

$$7x_1 + 4x_2 \leq 13;$$

$$x_1, x_2 \geq 0, \text{ int.}$$

Как и в задаче о садовнике, рассмотрим все возможные целочисленные решения. При этом сначала возьмем решение $x_1=1, x_2=1$ и решения, окружающие его (табл. 7.2). Целочисленные точки и ограничение показаны на рис. 7.1.

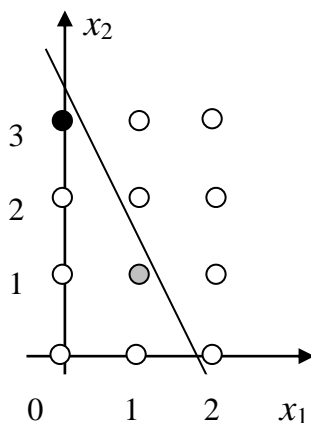


Рис. 7.1

Таблица 7.2

x_1	x_2	L
1	1	32
0	0	0
0	1	11
1	0	21
0	2	22
1	2	-
2	2	-
2	1	-
2	0	-
0	3	33

Из таблицы следует, что в точке $(1, 1)$ имеет место локальный экстремум, а глобальный максимум достигается в точке $(0, 3)$. В непрерывной линейной задаче любой локальный экстремум является глобальным. То что целочисленная задача может иметь локальные экстремумы, необходимо учитывать при использовании методов частичного перебора. ▲

В ряде случаев решение целочисленной задачи находят, решая ее как непрерывную. Так, если в оптимальном решении непрерывной задачи нецелочисленные значения переменных велики (их порядок $>10^2$), округление до целых оправдано: возможные нарушения условий и отклонение от оптимальности пренебрежимо малы.

При особых свойствах целочисленной задачи решение ее как непрерывной всегда дает целочисленный результат. Такой исход имеет место, если все вершины допустимого множества целочисленные. Многогранное множество, обладающее этим свойством, называется *целочисленным*. Определим условия, при которых множество оказывается целочисленным.

Возьмем многогранное множество $M(\mathbf{B})$:

$$\sum_{j=1}^n a_{ij}x_j = b_i \quad i = \overline{1, m}; \quad (7.1)$$

$$x_j \geq 0, \quad j = \overline{1, n}, \quad (7.2)$$

где все a_{ij} – фиксированные целые числа, $\mathbf{B} = (b_1, b_2, \dots, b_m)^T$ и $m < n$ (ранг матрицы $[a_{ij}]$ равен m .) Для него справедлива следующая теорема.

Теорема. Для того чтобы все вершины многогранного множества $M(\mathbf{B})$ при любом целочисленном векторе \mathbf{B} были целочисленными, необходимо и достаточно, чтобы каждый минор порядка m матрицы условий $[a_{ij}]$ был равен либо 0, либо +1 или –1.

Если вместо равенств (7.1) множество задается неравенствами, указанные в теореме значения относятся ко всем минорам матрицы $[a_{ij}]$.

Класс задач, удовлетворяющих теореме, очень узок (это транспортные задачи, задачи о назначениях и др.). Такие задачи относят к *леггоразрешимым* (по Дж. Эдмондсу), так как для них существуют *полиномиальные* алгоритмы (время или число итераций растет полиномиально с увеличением размерности задачи). Остальные целочисленные задачи входят в класс *трудноразрешимых* задач (класс *NP* по Карпу и Куку).

Для решения таких задач применяются различные подходы. Из точных методов можно назвать следующие:

1. методы отсечений;
2. метод ветвей и границ;
3. метод построения последовательности планов
4. модификации динамического программирования;
5. методы последовательного анализа вариантов.

Последние 4 метода входят в группу *комбинаторных* методов.

Кроме точных методов имеется также большое число приближенных методов.

Мы остановимся подробнее на первых двух методах как наиболее широко применяемых в пакетах программ, предназначенных для решения задач математического программирования. В ряде из них используются комбинация точных методов, добавление эвристических оценок и другие приемы, позволяющие повысить эффективность алгоритмов.

7.2. Метод отсечений

Идею этого метода высказал Г. Данциг. Она заключается в преобразовании невыпуклого множества целочисленной задачи в выпуклое целочисленное путем отсечения от выпуклого множества непрерывной задачи частей, не содержащих целочисленных точек. Тогда

использование методов ЛП гарантирует получение оптимального целочисленного решения (при разрешимости задачи).

С этой целью строится выпуклая оболочка допустимого множества целочисленной задачи. Выпуклой оболочкой невыпуклого множества Q называется наименьшее выпуклое множество, содержащее Q . В целочисленной задаче она может быть построена соединением крайних целочисленных точек допустимого множества гиперплоскостями. Пример построения выпуклой оболочки для задачи с двумя переменными показан на рис. 7.2. Здесь соединение крайних точек прямыми позволило получить целочисленное многогранное множество, содержащее все допустимые решения целочисленной задачи. Без требования целочисленности допустимое множество данной задачи представляет собой выпуклый четырехугольник. Как видно, разность этих множеств не содержит целочисленных решений.

Геометрически все выглядит достаточно просто. Но формализовать процедуру построения целочисленного множества долгое время не удавалось. Первым, кто смог это сделать, был Р. Гомори (1958 г.).

Он предложил итерационную процедуру, по которой на каждой итерации отсекается часть множества непрерывной задачи (НЗ), не содержащая целочисленных решений, но включающая оптимальное решение НЗ, и на сокращенном таким способом непрерывном множестве отыскивается новое оптимальное решение одним из методов ЛП. Итерации заканчиваются, когда оптимальное решение очередной НЗ окажется целочисленным или обнаружится неразрешимость НЗ, а значит, и целочисленной задачи. При этом выпуклая оболочка может быть построена только частично.

Рассмотрим пример (рис. 7.3). Оптимальное решение НЗ как по критерию L_1 , так и по L_2 находится в вершине А. После первого отсечения нецелочисленной части множества, содержащей точку А, появляется целочисленная вершина В. При решении задачи по критерию L_1 в ней будет оптимум НЗ, а значит, и исходной целочисленной задачи. Если же взять критерий L_2 , то оптимум НЗ окажется в вершине С, которая не является целочисленной. Поэтому потребуется еще одно отсечение, после которого будет получено оптимальное целочисленное решение в точке F. В

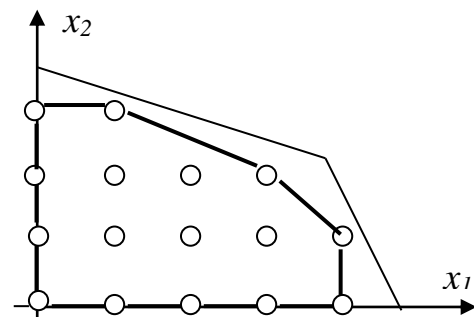


Рис. 7.2

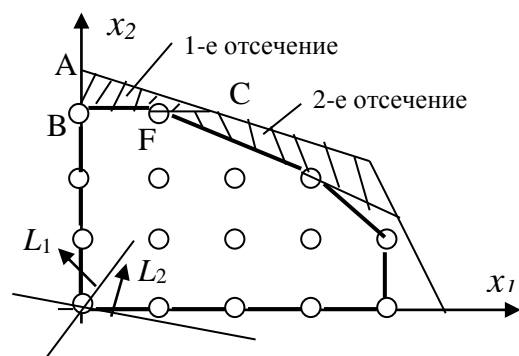


Рис. 7.3

обоих случаях выпуклая оболочка строится только частично.

Проблема состояла в получении регулярного условия, присоединение которого к ограничениям НЗ приводит к необходимому отсечению. Это условие должно удовлетворять двум требованиям: 1) не выполняться в текущем оптимальном решении НЗ; 2) выполняться во всех допустимых целочисленных решениях. Первое требование обеспечит отсечение части непрерывного множества, второе – неизменность целочисленного множества.

Гомори предложил несколько вариантов получения условий отсечения. Мы покажем вывод условия отсечения, которое применяется в 1-м алгоритме Гомори.

Пусть получено оптимальное решение НЗ. Уравнение, соответствующее строке оптимальной симплекс-таблицы с i -й базисной переменной, записывается следующим образом:

$$x_i^* + \sum_{j \in \text{небаз}} \alpha_{ij} \cdot x_j^* = \alpha_{i0}, \quad (7.3)$$

где α_{i0} – значение базисной переменной x_i^* (из столбца \mathbf{A}_0), α_{ij} – коэффициенты при небазисных переменных (из столбцов \mathbf{A}_j).

Нас интересуют переменные, которые имеют нецелые значения в полученном оптимальном решении. В этих случаях коэффициент α_{i0} в (7.3), естественно, не целый, а коэффициенты α_{ij} могут быть любыми действительными числами.

Нецелое значение представим в виде целой и дробной частей. Целая часть числа – наибольшее целое, не превосходящее само число. Будем обозначать ее взятием исходной величины в символы $\lfloor \cdot \rfloor$. Например, $\lfloor 2.1 \rfloor = 2$, $\lfloor 1.9 \rfloor = 1$, $\lfloor -2.1 \rfloor = -3$.

Применим такое представление к коэффициентам в (7.3). Тогда для дробной части имеем

$$\hat{\alpha}_{ij} = \alpha_{ij} - \lfloor \alpha_{ij} \rfloor,$$

следовательно, для нецелого α_{ij} всегда

$$0 < \hat{\alpha}_{ij} < 1.$$

Перепишем (7.3) с выделенными целыми и дробными частями коэффициентов:

$$x_i^* + \sum_{j \in \text{небаз}} (\lfloor \alpha_{ij} \rfloor + \hat{\alpha}_{ij}) x_j^* = \lfloor \alpha_{i0} \rfloor + \hat{\alpha}_{i0}. \quad (7.4)$$

Оставим в левой части (7.4) только целые части коэффициентов. Тогда, учитывая неотрицательность $\hat{\alpha}_{ij}$ и x_j^* , получаем неравенство

$$x_i^* + \sum_{j \in \text{небаз}} \lfloor \alpha_{ij} \rfloor x_j^* \leq \lfloor \alpha_{i0} \rfloor + \hat{\alpha}_{i0}. \quad (7.5)$$

Теперь воспользуемся требованием целочисленности. При целых переменных левая часть неравенства (7.5) может принимать только целые значения. Поэтому, если отбросить $\hat{\alpha}_{i0}$, нестрогое неравенство левой и правой частей сохранится:

$$x_i^* + \sum_{j \in \text{небаз}} \lfloor \alpha_{ij} \rfloor x_j^* \leq \lfloor \alpha_{i0} \rfloor. \quad (7.6)$$

(В этом легко убедиться на простых примерах: $9 \leq 10.3$; $9 \leq 9.75$).

Вычитая (7.6) из равенства (7.3), получаем:

$$\sum_{j \in \text{небаз}} \hat{\alpha}_{ij} x_j^* \geq \hat{\alpha}_{i0}. \quad (7.7)$$

Это и есть искомое условие отсечения. Действительно, в оптимальном решении НЗ (как и в любом базисном) небазисные переменные равны нулю, а $\hat{\alpha}_{i0} > 0$, следовательно, неравенство (7.7) в нем не выполняется. Поэтому добавление (7.7) к исходным условиям НЗ приведет к сужению допустимого множества за счет отсечения его части с оптимальной вершиной.

Пример 7.1. Выведем условие отсечения для задачи

$$\begin{aligned} L &= 2x_1 + x_2 \rightarrow \max; \\ 15x_1 + 30x_2 &\leq 96; \\ \forall x_j &\geq 0, \text{ int.} \end{aligned}$$

Приводим неравенство к каноническому виду

$$15x_1 + 30x_2 + x_3 = 96$$

и решаем непрерывную задачу симплекс-методом. Получаем оптимальную симплекс-таблицу

Базис	A_0	A_1	A_2	A_3
A_1	$\frac{96}{15}$	1	2	$\frac{1}{15}$
Δ	$\frac{192}{15}$	0	3	$\frac{2}{15}$

Графическое решение показано на рис. 7.4.

Записываем уравнение (7.3) по переменной x_1 :

$$x_1 + 2x_2 + \frac{1}{15}x_3 = \frac{96}{15}.$$

Дробную часть кроме свободного члена имеет только коэффициент при x_3 . Следуя (7.7), получаем условие отсечения

$$\frac{1}{15}x_3 \geq \frac{6}{15} \text{ или } x_3 \geq 6.$$

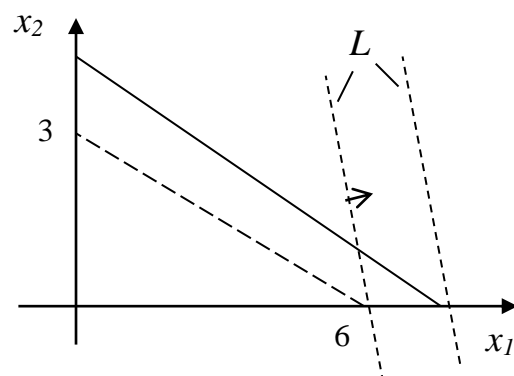


Рис. 7.4

Очевидно, что в оптимальном решении НЗ оно не выполняется ($x_3^* = 0$). Условие отсечения можно записать и через основные переменные. Так как $x_3 = 96 - 15x_1 - 30x_2$, то $96 - 15x_1 - 30x_2 \geq 6$ и окончательно имеем $x_1 + 2x_2 \leq 6$. Граница по этому ограничению показана на рис. 7.4 пунктирной линией. Легко убедиться, что все вершины усеченного множества целочисленные.



При многих нецелых переменных возникает вопрос выбора переменной, по которой следует строить отсечение. Строгого обоснования выбора не имеется. На основе экспериментальных данных рекомендуется брать переменную с наибольшей дробной частью. Второй вопрос относится к способу учета очередного условия отсечения: его можно добавить к условиям исходной задачи и решать задачу заново или после добавления продолжить симплекс-преобразования с полученного оптимального решения, которое стало недопустимым. В алгоритме Гомори применяется второй вариант как более экономичный.

Перед добавлением условие отсечения приводится к равенству:

$$\sum_{j \in \text{небаз}} \hat{\alpha}_{ij} \cdot x_j^* - x_{\bar{n}+1} = \hat{\alpha}_{i0}. \quad (7.8)$$

Так как небазисные переменные равны нулю, то новая дополнительная переменная $x_{\bar{n}+1} = -\alpha_{i0} < 0$. Поэтому рекомендуется для последующего решения применять двойственный симплекс-метод.

Таким образом, согласно 1-му алгоритму Гомори необходимо выполнить следующие действия.

1. Преобразовать условия задачи так, чтобы все коэффициенты стали целыми.
2. Решить исходную задачу без учета целочисленности (НЗ) одним из методов линейного программирования. Если непрерывная задача неразрешима, то зафиксировать неразрешимость исходной задачи и перейти на 9.
3. Проверить решение на целочисленность: если решение целочисленное, то зафиксировать получение оптимального решения и перейти на 9.
4. Если не все переменные целые, то из оптимальной таблицы выбрать переменную с наибольшей дробной частью.
5. Выписать из симплекс-таблицы строку с выбранной базисной переменной.
6. Выделить дробные части коэффициентов в полученном уравнении и записать условие отсечения согласно (7.7).
7. Привести условие отсечения к равенству (7.8), умножить его на -1 и добавить полученную строку к оптимальной симплекс-таблице. При этом размерность базиса увеличивается на единицу. В качестве недостающей базисной переменной принимается дополнительная переменная из новой строки.

8. Решить расширенную задачу двойственным симплекс-методом. Если задача разрешима, перейти на 3. Иначе зафиксировать неразрешимость целочисленной задачи.
9. Конец.

Гомори доказал сходимость этого алгоритма.

Примечание. Как следует из алгоритма, на каждой итерации увеличивается размер таблицы (число условий) на единицу. Для ограничения размера используется такой прием. Как только дополнительная переменная какого-либо условия отсечения снова становится базисной (но уже положительной!), строка с ней удаляется из таблицы. Если велся столбец этой переменной, то он тоже удаляется. Подобное исключение не отразится на решении, так как условие удаляется после того, как становится неактивным. В результате число строк не превысит \bar{n} .

Пример 7.2. Применим приведенный алгоритм к задаче

$$\begin{aligned} L &= 2x_1 + x_2 \rightarrow \max; \\ 2x_1 + x_2 &\geq 7,5; \\ 12x_1 + 7x_2 &\leq 55; \\ 5x_1 + 2x_2 + x_5 &\leq 18; \\ x_1, x_2 &\geq 0, \text{ целые.} \end{aligned}$$

Умножаем 1-е неравенство на 2 и приводим все условия к равенствам:

$$\begin{aligned} 4x_1 + 2x_2 - x_3 &\geq 15; \\ 12x_1 + 7x_2 + x_4 &\leq 55; \\ 25x_1 + 10x_2 + x_5 &\leq 90. \end{aligned}$$

Решаем непрерывную задачу симплекс-методом. В табл. 7.3 представлено решение НЗ (оптимальная таблица выделена двойной рамкой, строка оценок записана первой для удобства расширения таблицы).

Таблица 7.3

Базис	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆
Δ_j	8, 27	0	0	0	$\frac{5}{55}$	$\frac{2}{55}$	0
A ₂	$\frac{295}{55}$	0	1	0	$\frac{25}{55}$	$-\frac{12}{55}$	0
A ₃	$\frac{425}{55}$	0	0	1	$\frac{50}{55}$	$\frac{20}{55}$	0
A ₁	$\frac{80}{55}$	1	0	0	$-\frac{10}{55}$	$\frac{7}{55}$	0
A ₆	$-\frac{40}{55}$	0	0	0	$-\frac{50}{55}$	$-\frac{20}{55}$	1
θ	-	-	-	-	$\frac{5}{50}$	$\frac{2}{20}$	-

Решение содержит нецелые переменные. Наибольшую дробную часть имеет переменная x_3 . Выписываем соответствующую ей строку:

$$x_3 + \frac{50}{55} x_4 + \frac{20}{55} x_5 = \frac{425}{55}.$$

Из нее получаем 1-е условие отсечения

$$\frac{50}{55} x_4 + \frac{20}{55} x_5 \geq \frac{40}{55},$$

которое приводим к равенству

$$-\frac{50}{55} x_4 - \frac{20}{55} x_5 + x_6 = -\frac{40}{55}.$$

Это равенство добавляем к оптимальной симплекс-таблице решенной НЗ с включением в число базисных вектора A_6 (табл.7.3). К расширенной таблице применяем двойственный метод (направляющей является новая строка, направляющий столбец находится по θ). Полученное решение, которое называют оптимальным относительно первого отсечения, приведено в табл. 7.4 (см. часть в двойной рамке).

Таблица 7.4

Базис	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Δ_j	8, 2	0	0	0	0	0	$\frac{2}{20}$	0
A_2	$\frac{116}{20}$	0	1	0	1	0	$-\frac{12}{20}$	0
A_3	$\frac{140}{20}$	0	0	1	0	0	1	0
A_1	$\frac{24}{20}$	1	0	0	$-\frac{10}{20}$	0	$\frac{7}{20}$	0
A_5	$\frac{40}{20}$	0	0	0	$\frac{50}{20}$	1	$-\frac{55}{20}$	0
A_7	$-\frac{16}{20}$	0	0	0	0	0	$-\frac{8}{20}$	1

Оно также является нецелочисленным. Новое отсечение строим по переменной x_2 , имеющей наибольшую дробную часть. Выписываем уравнение

$$x_2 + x_4 - \frac{12}{20} x_6 = \frac{116}{20},$$

из которого следует условие отсечения

$$\frac{8}{20} x_6 \geq \frac{16}{20}$$

и затем равенство

$$\frac{8}{20}x_6 - x_7 = \frac{16}{20},$$

добавляемое к последней симплекс-таблице (см. табл.7.4). Применяем двойственный симплекс-метод.

Полученное решение, оптимальное относительно второго отсечения, дано в табл. 7.5.

Таблица 7.5

Базис	A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈
Δ _j	8,0	0	0	0	0	0	0	$\frac{2}{8}$	0
A ₂	$\frac{56}{8}$	0	1	0	1	0	0	$-\frac{12}{8}$	0
A ₃	$\frac{40}{8}$	0	0	1	0	0	0	$\frac{20}{8}$	0
A ₁	$\frac{4}{8}$	1	0	0	$-\frac{10}{20}$	0	0	$\frac{7}{8}$	0
A ₅	$\frac{60}{8}$	0	0	0	$\frac{50}{20}$	1	0	$-\frac{55}{8}$	0
A ₆	$\frac{16}{8}$	0	0	0	0	0	1	$-\frac{20}{8}$	0
A ₈	$-\frac{4}{8}$	0	0	0	$-\frac{4}{8}$	0	0	$-\frac{7}{8}$	1

Очевидно, что оно не удовлетворяет требованию целочисленности. Так как в базисное решение вернулась переменная x_6 , что означает строгое выполнение 1-го условия отсечения, соответствующие ей строка и столбец удаляются (отмечены выделением пунктирной полосой).

Теперь отсечение строим по переменной x_1 :

$$x_1 - \frac{10}{20}x_4 + \frac{7}{8}x_7 = \frac{4}{8} \Rightarrow \frac{4}{8}x_4 + \frac{7}{8}x_7 \geq \frac{4}{8} \Rightarrow \frac{4}{8}x_4 + \frac{7}{8}x_7 - x_8 = \frac{4}{8}.$$

Добавляем полученное уравнение отсечения к последней таблице (табл. 7.5) и, используя снова двойственный метод, находим оптимальное решение расширенной НЗ (табл. 7.6). Это решение целочисленное и, следовательно, оно является оптимальным для исходной целочисленной задачи. ▲

Таблица 7.6

Базис	A ₀
L	8, 0
A ₂	6
A ₃	5
A ₁	1
A ₅	5
A ₄	1

переменными, решение которых требовало более тысячи итераций.

Попытка вводить отсечение сразу по нескольким переменным не дает ощутимого эффекта.

Отмечается зависимость скорости сходимости от формы и порядка записи условий задачи, существенное влияние ошибок округления. Неприятной стороной алгоритма является постоянное увеличение числа строк вплоть до числа переменных в канонической форме. Если задача разрешима, алгоритм находит целочисленное решение только на последней итерации. Поэтому в случае прерывания расчета не будет получено ни одного даже приемлемого решения.

Эффективность алгоритма повышается с уменьшением значений a_{ij} и b_i и заполненности матрицы условий A . Можно рекомендовать применение метода для задач небольшой размерности (до десятков переменных), когда значения a_{ij} и b_i невелики и в оптимальном решении непрерывной задачи большая часть переменных имеет целые значения. Алгоритм мало пригоден для решения комбинаторных задач в целочисленной постановке.

Для других вариантов построения условий отсечения сходимость построенных на них алгоритмов не доказана, а отмеченные недостатки в основном не устраняются. Эти замечания относятся и к алгоритму отсечения для частично целочисленных задач.

В ряде пакетов прикладных программ метод отсечений применяется в комбинации с другими, точными или приближенными, методами.

7.3. Метод ветвей и границ

Начало развитию подхода, получившего название метод ветвей и границ, положила работа Ленд и Дойг (1960). Это, скорее, даже не метод, а концепция или процедурная оболочка, на основе которой стали разрабатывать алгоритмы решения целочисленных задач различной природы. Ценность предложенной идеи стала особенно заметна после появления первого точного алгоритма решения задачи коммивояжера, построенного по схеме ветвей и границ (Литтл с соавторами, 1963). Метод можно применять как к полностью, так и частично целочисленным задачам.

Суть идеи схожа с известной шуткой о ловле льва в пустыне: делим пустыню пополам; если льва нет в первой половине, ищем во второй, которую делим пополам и т. д. В отличие от льва оптимум не перемещается, и в этом смысле наша задача легче.

Метод заключается в построении дерева задач, корнем которого является исходная задача, возможно без условия целочисленности (НЗ). Нижележащие задачи порождаются вышележащими так, что их допустимые множества (ДМ) являются непересекающимися подмножествами ДМ вышележащей задачи. Рост дерева происходит за

счет перспективных ветвей. Перспективность определяется по *оценке критерия* терминальной задачи ветви V и *рекорду* Z . Оценка V – это значение критерия, заведомо не хуже оптимального, а Z – достигнутое в процессе решения значение критерия исходной задачи (в качестве начального может приниматься значение, заведомо хуже оптимального). Значит, задача будет порождающей только при условии, что ее оценка лучше рекорда. При этом уровень, на котором находится задача, не имеет значения.

Рассмотрим метод применительно к линейной целочисленной задаче. Хотя нет каких-либо ограничений на число задач, непосредственно порождаемых перспективной, в алгоритмах, как правило, используется разбиение на две задачи, то есть строится бинарное дерево (рис. 7.5). При этом для целочисленных множеств выполняются соотношения

$$\bigcup_i D_i = D, \quad D_i \cap D_j = \emptyset, \quad i \neq j. \quad (7.9)$$

Очевидно, что если, например, V_{22} окажется хуже рекорда или $D_{22} = \emptyset$, правая ветвь обрывается (говорят также, что она прозондирована). Если же оценка V_{22} будет лучше Z , производится ветвление: множество D_{22} разбивается на 2 подмножества. Решение завершится, когда все ветви будут прозондированы.

Вид оценки зависит от направленности критерия: при максимизации используется верхняя оценка, при минимизации – нижняя. Последующее изложение метода будет относиться к задаче на максимум.

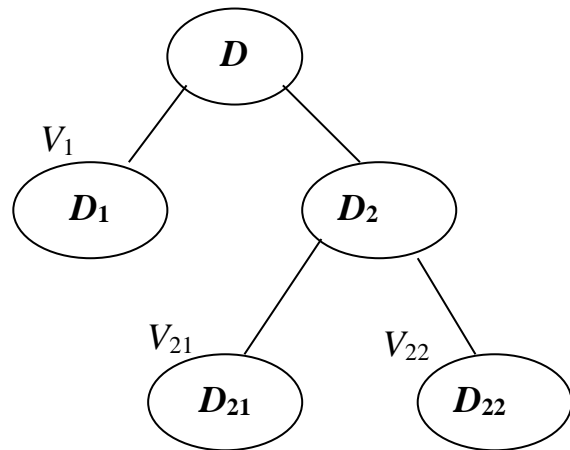


Рис. 7.5

Для алгоритмической реализации схемы ветвей и границ необходимо решить два основополагающих вопроса:

1. Каким образом разбивать перспективное множество на подмножества;
2. Как определять верхнюю оценку критерия на рассматриваемом множестве.

Ответы на них зависят от типа задачи (частично или полностью целочисленная, имеет особые свойства или нет, с булевыми или не булевыми переменными). Ниже рассматривается общий случай.

Пусть известен диапазон возможных значений j -й переменной

$$0 \leq x_j \leq d_j,$$

которая в непрерывном оптимальном решении оказалась нецелочисленной и равной x_j^* . Тогда целочисленное значение этой переменной может

достигаться либо в интервале $0 \leq x_j \leq \lfloor x_j^* \rfloor$, либо в интервале $\lfloor x_j^* \rfloor + 1 \leq x_j \leq d_j$, где $\lfloor x_j^* \rfloor$ - целая часть x_j^* (рис. 7.6).

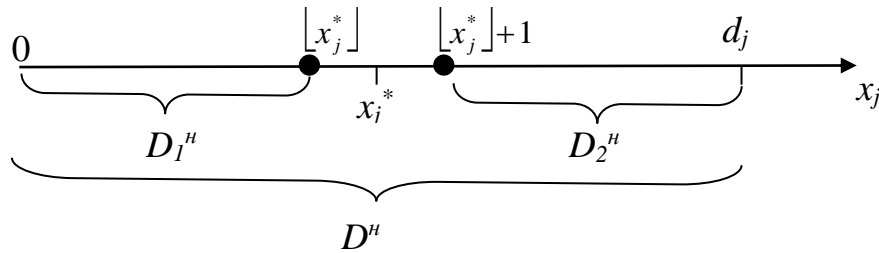


Рис. 7.6

Это соответствует разбиению непрерывного множества D^H на два непересекающихся подмножества D_1^H и D_2^H , объединение которых не равно D^H . В то же время такое разбиение целочисленного множества удовлетворяет соотношениям (7.9). При этом целочисленные множества, как исходное, так и порожденные, включены в соответствующие непрерывные множества. Следовательно, поиск целочисленного решения на непрерывном множестве даст тот же результат, что и на целочисленном. Легко увидеть, что приведенное выделение подинтервалов по одной переменной приводит к разбиению исходного множества на два подмножества при любом числе переменных.

Теперь перейдем ко второму вопросу. Так как целочисленное множество является подмножеством соответствующего непрерывного, оптимальное значение критерия на непрерывном множестве всегда будет не меньше, чем на целочисленном. Поэтому в качестве верхней оценки V можно брать оптимальное значение критерия L^* непрерывной задачи.

Выбор начального значения рекорда зависит от ситуации:

- если известно какое-либо целочисленное значение, то рекорд принимается равным критерию в этом решении;
- при положительности всех коэффициентов критерия можно взять нулевое значение рекорда;
- в иных случаях за начальное значение рекорда берется $-M$, где M - максимально представимое в компьютере число.

По ходу разбиения формируются порождаемые задачи, которые помещаются в *список задач*. Первоначальный список содержит только одну задачу – исходную задачу без условий целочисленности. И в последующем список будет содержать только непрерывные задачи.

Таким образом, базовый алгоритм, реализующий метод ветвей и границ, включает следующие шаги.

1. Задается начальное значение рекорда и в список задач помещается исходная задача без требования целочисленности переменных.
2. Анализируется список задач: если он пуст, то переход на шаг 6. Иначе выбирается одна из задач с удалением ее из списка.

3. Выбранная задача решается одним из методов линейного программирования. Если задача неразрешима или оптимальное значение критерия $L^* \leq Z$, ветвь обрывается (задача прозондирована). Переход на шаг 2.
4. Полученное решение анализируется на целочисленность. Если решение целочисленное, оно фиксируется, рекорду присваивается оптимальное значение критерия решенной непрерывной задачи ($Z:=L^*$), ветвь обрывается и осуществляется переход на шаг 2.
5. Выбирается одна из переменных, имеющих нецелочисленные значения. По ней производится ветвление: порождаются 2 задачи, одна образуется присоединением к решенной (родительской) задаче условия $x_j \leq \lfloor x_j^* \rfloor$, другая – добавлением к родительской ограничения $x_j \geq \lfloor x_j^* \rfloor + 1$. Эти задачи заносятся в список задач. Переход на шаг 2.
6. Вывод результатов (если значение рекорда больше начального, получено оптимальное решение исходной задачи, иначе задача неразрешима).

Приведенный алгоритм является базовым, так как не включает однозначных правил выбора задачи из списка и ветвящей переменной. Для частично целочисленных задач при выборе переменной для ветвления исключаются непрерывные переменные.

Пример 7.3. Применим алгоритм ветвей и границ к задаче

$$L=9x_1+5x_2 \rightarrow \max;$$

$$3x_1 - 6x_2 \geq 1;$$

$$5x_1 + 2x_2 \leq 28;$$

$$\forall x_j \geq 0, \text{ целые.}$$

Отбрасывая условие целочисленности, получаем непрерывную задачу, которую помещаем в список задач. Так как коэффициенты критерия положительны, начальное значение рекорда принимаем равным нулю. Берем из списка единственную задачу и решаем ее. Получаем оптимальное решение в вершине А (рис. 7.7): $x_1^*=4,72$; $x_2^*=2,19$. Ветвление производим по переменной x_1 . Добавляя к решенной задаче ограничение $x_1 \leq 4$, образуем задачу 2, а добавление $x_1 \geq 5$ дает задачу 3. Допустимые множества новых задач показаны на рис. 7.7. Эти задачи помещаем в список задач. Решение задачи 2 достигается в точке В, а задачи 3 – в С. Весь ход решения исходной задачи представлен в виде дерева решений на рис. 7.10. Порядок решения задач из списка отражает счетчик итераций k . На 3-й итерации (задача 4) получено

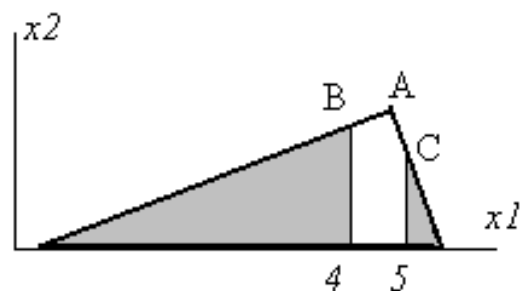


Рис. 7.7

целочисленное решение со значением критерия 41 (точка D на рис. 7.8). Поэтому изменяется рекорд: $Z=41$. Задача 6 имеет нецелочисленное решение (вершина E на рис. 7.9), задача 8 – целочисленное решение в точке F. В результате после 7-й итерации рекорд становится равным 50.

Остальные задачи не имеют допустимых решений, то есть список

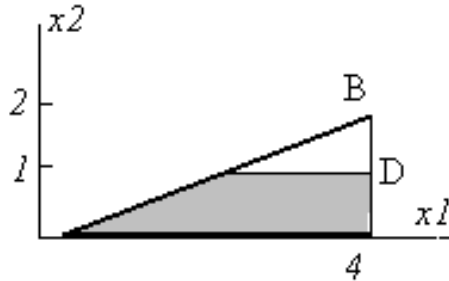


Рис. 7.8



Рис. 7.9

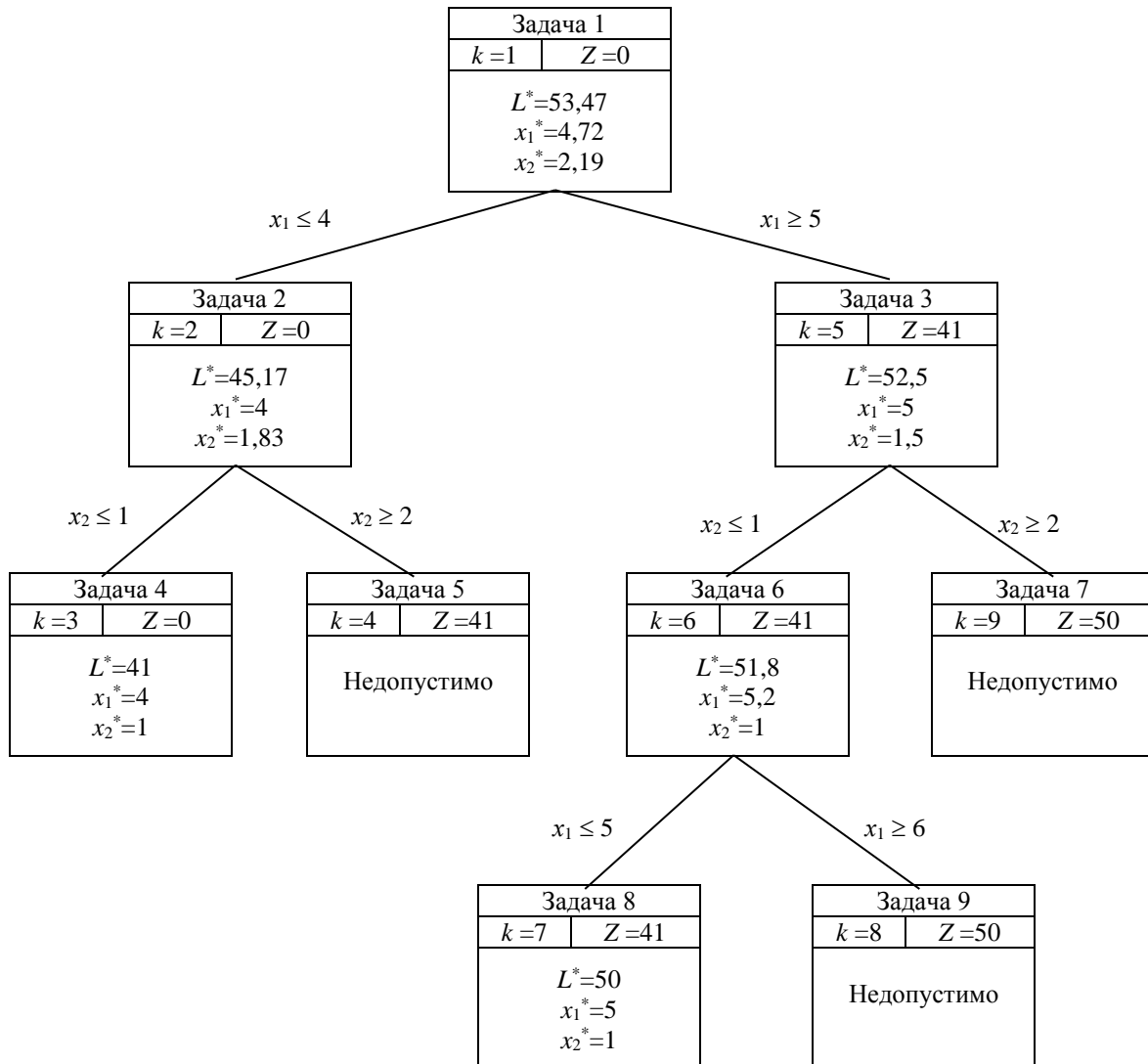


Рис. 7.10

задач исчерпывается и, таким образом, констатируем получение оптимального решения исходной задачи, равное решению непрерывной задачи 8.

Из приведенного дерева решений видно, что число задач в списке могло быть меньше при другом порядке решения задач. Действительно, если бы сначала были решены задачи правой ветви с рекордом $Z=50$, то после решения задачи 2 не произошло бы ветвления, так как верхняя оценка оказалась бы ниже рекорда ($V=L^*=45,17 < 50$).

Естественно возникает вопрос: а как на числе задач и дереве решений

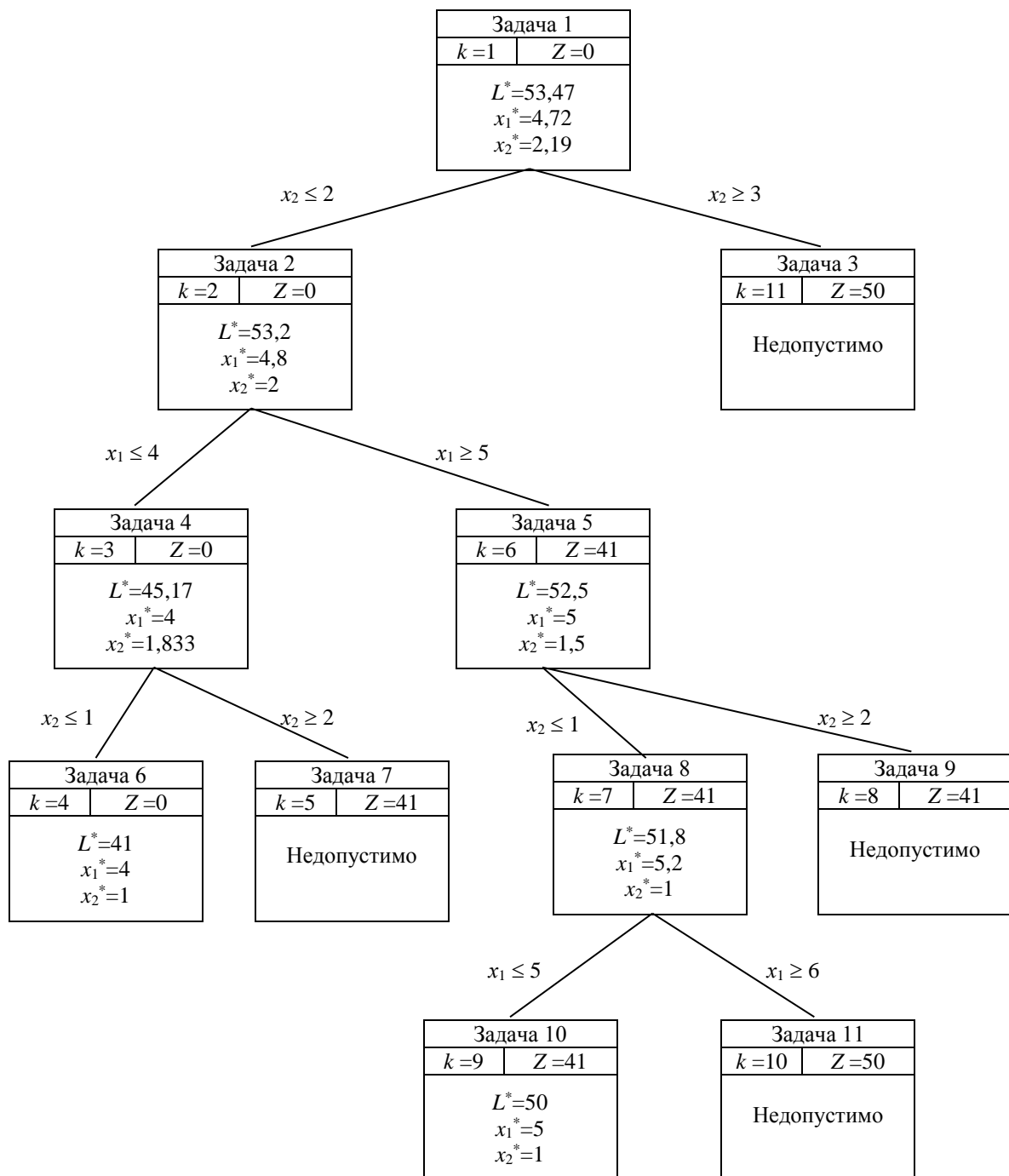


Рис. 7.11

может отразиться выбор другой переменной для ветвления? Так, в нашем примере если после 1-й итерации произвести ветвление по переменной x_2 , то получим дерево, показанное на рис. 7.11. Оно содержит на 2 задачи больше, чем на рис. 7.10. Конечно, оно может быть также другим при ином порядке решения задач.

Таким образом, число решаемых задач существенно зависит от выбора задачи из списка и переменной для ветвления.

Из алгоритма и приведенного примера следует, что ветвь обрывается по одной из трех причин:

1. неразрешимость задачи;
2. задача имеет целочисленное решение;
3. верхняя оценка не больше рекорда.

Теперь сделаем ряд замечаний относительно метода ветвей и границ. Как уже отмечалось, в базовом алгоритме не оговариваются правила выбора задачи и переменной. В большинстве программных реализаций метода используются правила, основанные на эвристических оценках перспективности задач и переменных. В некоторых пакетах, например, "ЛП в АСУ" предлагается несколько вариантов управления процессом решения: от автоматического до ручного, в котором пользователь может сам делать выбор как задачи, так и переменной. Кроме того, алгоритмы, основанные на методе ветвей и границ, могут существенно отличаться в связи с учетом особенностей класса задач. Например, для задачи коммивояжера, определение оценки значительно упрощено (не требуется решать непрерывную линейную задачу).

Метода ветвей и границ имеет преимущества в сравнении с методом отсечений:

- накопление ошибок менее значительное, так как решение идет по разным ветвям;
- при принудительной остановке процесса решения высока вероятность получения целочисленного результата, но без установления его оптимальности;
- при решении непрерывных задач размеры симплекс-таблиц не увеличиваются.

Недостатки метода ветвей и границ:

- Нельзя оценить число задач, которые придется решать. Чем ближе снизу начальное значение рекорда и сверху оценка критерия задачи к искомому оптимальному значению критерия, тем меньше вершин будет иметь дерево решений, а значит, и затрат ресурсов. Однако завышение начального рекорда может привести к неразрешимости задачи, что всегда следует иметь в виду.
- Отсутствие признака оптимальности. Оптимальное решение может быть получено задолго до останова алгоритма, но обнаружить это в

общем случае нельзя. Оптимальность устанавливается только по исчерпанию списка задач.

Очевидно, что эффективность метода повышается с уменьшением диапазонов значений переменных и числа нецелых переменных в решении первой непрерывной задачи.

7.4. Аддитивный алгоритм

Он разработан применительно к задачам с булевыми переменными. В нем выполняются операции только сложения и вычитания (отсюда и название метода). Поэтому не происходит и накопления ошибок. Алгоритм представляет собой реализацию одного из методов частичного перебора. Его можно рассматривать также как частный случай метода ветвей и границ.

Модель задачи должна быть представлена в стандартной форме. Далее будем рассматривать алгоритм применительно к следующей задаче:

$$L = \sum_{j=1}^n c_j x_j \rightarrow \min; \quad (7.10)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}; \quad (7.11)$$

$$\forall x_j = \begin{cases} 0 \\ 1 \end{cases}. \quad (7.12)$$

При этом $\forall c_j \geq 0$, что означает выполнение признака оптимальности симплекс-метода в начальном решении (в задачах на минимум). Действительно, так как коэффициенты дополнительных переменных $c_i=0$, то $z_j=0$ и $\Delta_j = z_j - c_j \leq 0$. Поэтому, если еще и $\forall b_i \geq 0$, то сразу имеем оптимальное решение задачи: все n исходных переменных и критерий равны нулю. Однако обычно не все b_i положительны и нулевое начальное решение оказывается недопустимым.

Если в критерии есть отрицательные коэффициенты, то модель преобразуется: переменные x_k с $c_k < 0$ всюду в модели заменяются на $x_k = 1 - x_k'$ и образующаяся в критерии константа отбрасывается (после получения решения она добавляется к оптимальному значению критерия). Если есть равенства, они преобразуются в неравенства. Неравенства \geq преобразуются в неравенства \leq . Таким образом, любая исходная модель может быть приведена к виду (7.10)-(7.12) с $\forall c_j \geq 0$.

Представим условия (7.11) в канонической форме:

$$\sum_{j=1}^n a_{ij} x_j + S_i = b_i, \quad i = \overline{1, m}, \quad (7.13)$$

где S_i – дополнительные переменные. Тогда начальное решение очевидно: $\forall x_j = 0$ и $S_i = b_i$. Как уже говорилось, если $\forall S_i \geq 0$, оно оптимально. В противном случае осуществляется частичный перебор решений. Для

пояснения алгоритма воспользуемся табл. 7.7, которая аналогична симплексной.

Таблица 7.7

A_0 (Решение)	x_1	x_2	\dots	x_n	S_1	S_2	\dots	S_m
b_1	a_{11}	a_{12}	\dots	a_{1n}	1	0	\dots	0
b_2	a_{21}	a_{22}	\dots	a_{2n}	0	1	\dots	0
\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots	\dots
b_m	a_{m1}	a_{m2}	\dots	a_{mn}	0	0	\dots	1
L	c_1	c_2	\dots	c_n	0	0	\dots	0

В алгоритме переменные разделяются на фиксированные и свободные. Переменную которой присвоено определенное значение, называют фиксированной. Значение свободной переменной можно изменять. Основным объектом алгоритма является *частичное решение* – это решение, в котором часть переменных фиксирована. Оно описывается упорядоченным множеством индексов фиксированных переменных. При этом индекс переменных, равных нулю, записывается со знаком "минус". Например, если на t -й итерации фиксированы $x_2=0$, $x_4=1$, то частичное решение представляется как множество $I_t=\{-2, 4\}$.

Первоначальное частичное множество всегда пустое ($I_0=\emptyset$), а значение рекорда $z=\infty$. Алгоритм состоит из четырех проверок, которые выполняются для того, чтобы определить наличие перспективных свободных переменных. Если такая переменная находится, то изменяя ее значение, можно улучшить результат. По умолчанию принято, что свободные переменные находятся на нижнем уровне (равны нулю). Частичное решение считается *прозондированным*, если оно не может привести к допустимому решению и уменьшению значения критерия.

Пусть имеем частичное решение I_t с критерием L^t и вектором дополнительных переменных $S^t=(S_1^t, S_2^t, \dots, S_m^t)$. К нему применяются следующие проверки.

1. Для каждой свободной переменной x_r проверяются коэффициенты a_{ir} в строках с $S_i^t < 0$ (табл.7.7). Если во всех таких строках $a_{ir} \geq 0$, переменная x_r исключается, так как изменение ее значения с 0 на 1 не приведет к положительности хотя бы одной из рассматриваемых S_i^t .
2. Анализируется возможность улучшения критерия. Если для свободной переменной x_r выполняется неравенство $C_r + L^t \geq z$, изменение ее значения не может привести к уменьшению рекорда. Поэтому она исключается.

Оставшиеся после этих проверок свободные переменные образуют множество P_t . Если оно пустое, то текущее частичное решение не перспективно, то есть считается прозондированным.

3. Выясняется возможность получения допустимого решения на основе данного частичного. В строках с $S_i^t < 0$ проверяется условие

$$\sum_{j \in P_t} \min(0, a_{ij}) > S_i^t. \quad (7.14)$$

Если оно выполняется хотя бы для одной строки, все переменные из P_t исключаются, так как изменение даже всех этих переменных с 0 на 1 не обеспечит допустимость решения (неотрицательности вектора S). В этом случае решение I_t считается прозондированным (ветвь обрывается).

Если условия (7.14) не выполняются, проводится проверка 4.

4. При $P_t \neq \emptyset$ ветвь продолжается. Для получения нового частичного решения из I_t вычисляются оценки каждой переменной из P_t :

$$v_j^t = \sum_i \min(0, S_i^t - a_{ij}). \quad (7.15)$$

Оценка дает суммарную величину недопустимости, остающейся после изменения значения переменной $x_j \in P_t$ с 0 на 1. Как видно из (7.15), отрицательная оценка свидетельствует о наличии недопустимости. Из полученных оценок определяется максимальная

$$\max_j v_j^t = v_k^t. \quad (7.16)$$

Очевидно, что если $v_k^t = 0$, то изменение x_k с 0 на 1 дает допустимое решение с меньшим значением критерия. Поэтому рекорду z присваивается значение $L^t + C_k$, а новое частичное решение $I_{t+1} = \{I_t, k\}$ считается прозондированным. Если же $v_k^t < 0$, то допустимое решение не достигнуто и частичное решение $I_{t+1} = \{I_t, k\}$ подвергается всем проверкам. Если в результате проверок оно окажется прозондированным, новое частичное решение получают из I_{t+1} изменением знака индекса введенной переменной: $I_{t+2} = \{I_t, -k\}$, то есть фиксацией x_k со значением 0.

В общем случае прозондированное частичное решение может содержать положительные и отрицательные индексы. Для получения нового частичного решения изменяется знак самого правого положительного индекса, а стоящие за ним индексы отбрасываются. Так из решения $\{2, -1, -3, 5, -7, -6\}$ следует частичное решение $\{2, -1, -3, -5\}$. Если представить весь процесс решения в виде дерева (подобно методу ветвей и границ), то отбрасывание l последних индексов означает возврат на l уровней вверх. Условием окончания работы аддитивного алгоритма является отсутствие положительных индексов в частичном решении.

Пример 7.4 (Таха). Решим с помощью аддитивного алгоритма следующую задачу.

$$\begin{aligned} L &= -3x_1 - 2x_2 + 5x_3 + 2x_4 - 3x_5 \rightarrow \min; \\ x_1 + x_2 + x_3 + 2x_4 + x_5 &\leq 4; \\ 7x_1 + 3x_3 - 4x_4 + 3x_5 &\leq 8; \\ 11x_1 - 6x_2 + 3x_4 - 3x_5 &\geq 3; \end{aligned}$$

$$\forall x_j = \begin{cases} 1 \\ 0 \end{cases}.$$

Так как C_1, C_2 и C_5 отрицательные, производим замены:

$$x_j = 1 - x_j', j=1, 2, 5.$$

После простых преобразований модель принимает вид

$$\begin{aligned} L_1 &= 3x_1' + 2x_2' + 5x_3 + 2x_4 + 3x_5' \rightarrow \min; \\ -x_1' - x_2' + x_3 + 2x_4 - x_5' &\leq 1; \\ -7x_1' + 3x_3 - 4x_4 - 3x_5' &\leq -2; \\ 11x_1' - 6x_2' - 3x_4 - 3x_5' &\leq -1. \end{aligned}$$

Приводим условия к равенствам:

$$\begin{aligned} -x_1' - x_2' + x_3 + 2x_4 - x_5' + S_1 &= 1; \\ -7x_1' + 3x_3 - 4x_4 - 3x_5' + S_2 &= -2; \\ 11x_1' - 6x_2' - 3x_4 - 3x_5' + S_3 &= -1. \end{aligned}$$

Полученную модель представляем в табличном виде (табл. 7.8).

Таблица 7.8

A_0	x_1'	x_2'	x_3	x_4	x_5'	S_1	S_2	S_3
1	-1	-1	1	2	-1	1		
-2	-7	0	3	-4	-3		1	
-1	11	-6	0	-3	-3			1
0	3	2	5	2	3	0	0	0

В исходном состоянии все переменные свободны и равны нулю. Поэтому начальное частичное решение $I^{(0)} = \emptyset, z = \infty, L_1^{(0)} = 0, S^{(0)} = (1, -2, -1)$.

Так как есть отрицательные S_i , начальное решение неоптимально и необходимо проводить проверки (ниже они обозначены соответствующими им номерами).

Итерация 1.

1. Поскольку $\forall a_{i3} \geq 0$, переменная x_3 исключается.
2. Для всех переменных $C_j + L_1^{(0)} < z$, поэтому не отвергается ни одна переменная.
3. $P_0 = \{1, 2, 4, 5\}$ – множество свободных переменных, которые прошли через первые две проверки. Для строк с отрицательными S_i по табл. 7.8 проверяем условие (7.14):

$$i=2: \sum_{j \in P_0} \min(0, a_{2j}) = -7 + 0 - 4 - 3 = -14 < S_2 = -2;$$

$$i=3: \sum_{j \in P_0} \min(0, a_{3j}) = 0 - 6 - 3 - 3 = -12 < S_3 = -1.$$

Условия не выполняются и все переменные остаются.

4. По формуле 7.15 вычисляем:

$$\begin{aligned} v_1^0 &= \min(0, 1+1) + \min(0, -2+7) + \min(0, -1-11) = 0+0+(-12) = -12; \\ \text{аналогично } v_2^0 &= 0+(-2)+0 = -2, \end{aligned}$$

$$v_4^0 = -1 + 0 + 0 = -1,$$

$$v_5^0 = 0 + 0 + 0 = 0.$$

Находим $\max v_j^0 = v_5^0 = 0$. Отсюда следует, что $k=5$ и новое частичное решение с $x_5' = 1$ является допустимым. В итоге имеем:

$I_1 = \{5\}$, $L_1^1 = 3$, $z = L_1^1 = 3$ и $S^1 = (2, 1, 2)$, так как $S_1^1 = S_1^0 - a_{15} = 1 - (-1) = 2$, $S_2^1 = S_2^0 - a_{25} = -2 - (-3) = 1$, $S_3^1 = S_3^0 - a_{35} = -1 - (-3) = 2$, то есть действительно все $S_i > 0$, что означает допустимость решения. Вывод: решение I_1 прозондировано.

Очередное частичное решение получается изменением знака индекса в I_1 .

Итерация 2.

$I_2 = \{-5\}$, $L_1^2 = 0$, $S^2 = (1, -2, -1)$, $z = 3$.

1. Исключается x_3 .

2. Исключается x_1' , так как $C_1 + L_1^2 = 0 + 3 = z$.

3. $P_2 = \{2, 4\}$.

$$i=2: \quad 0 - 4 = -4 < -2;$$

$$i=4: \quad -6 - 3 = -9 < -1.$$

4. $v_2^2 = 0 + (-2) + 0 = -2$, $v_4^2 = -1 + 0 + 0 = -1$, $\max v_i^2 = v_4^2 = -1$. Следовательно, $k=4$ и новое решение $I_3 = \{-5, 4\}$ недопустимое.

Итерация 3.

$I_3 = \{-5, 4\}$, $L_1^3 = C_4 = 2$, $S^3 = (-1, 2, 2)$, $z = 3$. Свободными являются первые 3 переменные.

1. Исключается x_3 .

2. Исключаются x_1' и x_2' , так как $L_1^3 + C_1 = 5 > 3$ и $L_1^3 + C_2 = 4 > 3$.

$P_3 = \emptyset$, значит, решение I_3 прозондировано.

Так как есть частичное решение I_3 с положительным индексом, образуем из него решение I_4 , заменив 4 на -4 .

Итерация 4.

$I_4 = \{-5, -4\}$, $L_1^4 = 0$, $S^4 = \{1, -2, -1\}$, $z = 3$.

1. Исключается x_3 .

2. Исключается x_1' .

3. $P_4 = \{2\}$.

$i=2: \quad 0 > -2$, следовательно, x_2' исключается и I_4 прозондировано (оно не может привести к допустимому решению).

Больше нет частичных решений с положительными индексами, из чего заключаем, что итерации завершены и оптимальным является решение I_1 : $x_1' = x_2' = x_3 = x_4 = 0$, $x_5' = 1$. Возвращаясь к исходным переменным, получаем

$$x_1^* = x_2^* = 1, \quad x_3^* = x_4^* = x_5^* = 0, \quad L^* = 5.$$

Дерево решений рассмотренного примера показано на рис. 7.12. Здесь хорошо видна аналогия с методом ветвей и границ.

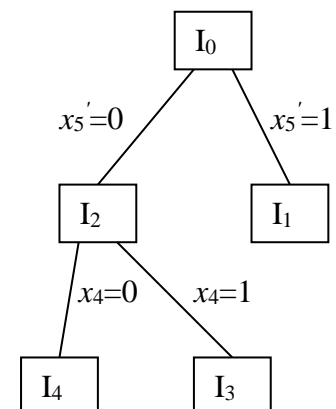


Рис. 7.12

7.5. Другие методы

В.Комлик и В.Емеличев (1968) предложили метод построения последовательности планов, который относится к классу точных методов. Идея метода состоит в следующем. Исходная линейная целочисленная задача максимизации с m условиями заменяется задачей с одним условием (расширенная задача). Для определения этого условия решается m задач, каждая с одним условием. Находится минимум из m оптимальных значений критериев $\min L_i^* = L_t^*$. Задача t берется в качестве расширенной (допустимое множество исходной задачи является ее подмножеством). Если оптимальный план этой задачи не является допустимым для исходной, он удаляется из допустимого множества расширенной задачи и она решается вновь. Новый оптимальный план снова проверяется на допустимость и удаляется при недопустимости. Так строится последовательность невозвращающихся (по критерию) планов. Решение завершается, как только очередной план расширенной задачи окажется допустимым для исходной, а следовательно, он будет ее оптимальным решением. Для общей задачи целочисленного программирования решение расширенной рекомендуется искать методом динамического программирования.

Точные методы целочисленного программирования отличаются высокой трудоемкостью, причем с ростом числа переменных она возрастает экспоненциально. Поэтому они применяются для задач до средней размерности. Практически решаются общие задачи с несколькими сотнями переменных, а специальные задачи (особые свойства, булевы переменные) – с 1-2 тысячами переменных. При этом в более “мощных” алгоритмах метод ветвей и границ сочетается с методом отсечений. Последний применяется при решении подзадач как способ уточнения верхней оценки: отсечения проводятся только до тех пор, пока изменяется критерий.

Для решения задач средней и большой размерности применяют приближенные методы. Класс приближенных методов весьма обширен. Во многих из них используются идеи точных методов, модифицированные для поиска приближенного решения. Значительную группу составляют методы, основанные на эвристиках - правилах, не имеющих строгого обоснования, но отвечающих здравому смыслу (представлениях о свойствах оптимального решения с учетом специфики задачи). Такие методы работают быстро, но невозможно оценить качество получаемого решения. Как правило, находится некоторое локальное решение.

В методе вектора спада (Сергиенко) локальность решения проверяется непосредственно. Вектор спада минимизируемого критерия – это векторная функция от x относительно окрестности радиуса $r > 0$

(замкнутого шара) с центром в точке x . Размерность вектора спада равна числу целочисленных точек в шаре без x . Его компоненты характеризуют поведение критерия при отклонении от x . Они подобны относительным оценкам в симплекс-методе, но вычисляются для целочисленных точек в шаре. Если все компоненты вектора неотрицательны, то текущая точка x является локальным минимумом. В этом случае радиус шара увеличивается и вектор спада пересчитывается. Последовательность радиусов $r_1 < r_2 < \dots < r_m$ и начальная точка задаются априорно. Если некоторая компонента показывает улучшение решения, то осуществляется переход в лучшую точку, которая принимается за текущую, и вектор спада вычисляется относительно этой точки для шара без изменения радиуса. Если установлено, что исследуемая точка является локальным минимумом на шаре с максимальным радиусом, она принимается за решение задачи. Характеристики этого метода сильно зависят от способа вычисления вектора спада и предварительно выбираемых значений параметров алгоритма.

Ряд приближенных методов разработан для решения определенных задач с учетом их специфики, благодаря чему они более эффективны по сравнению с другими приближенными методами.

Кроме перечисленных подходов находят применение приближенные методы, использующие случайный поиск и особенно генетические алгоритмы. Последние сочетают случайные действия с детерминированными. В частности, несколько генетических алгоритмов разработано для задачи коммивояжера.

7.6. Задания для самостоятельной работы

Решить задачи методом ветвей и границ. Корневую задачу решить симплекс-методом, остальные графически. Построить дерево решений.

№1	$L = 7x_1 + 4x_2 \rightarrow \max$ $4x_1 + x_2 \leq 14$ $3x_1 + 2x_2 \leq 12$ $\forall x_j \geq 0$, цел.	№2	$L = 2x_1 + 3x_2 \rightarrow \min$ $7x_1 + 20x_2 \geq 140$ $2x_1 + x_2 \geq 18$ $\forall x_j \geq 0$, цел.
№3	$L = 6x_1 + 4x_2 \rightarrow \min$ $4x_1 + 2x_2 \geq 7$ $x_1 - x_2 \leq 1$ $\forall x_j \geq 0$, цел.	№4	$L = 6x_1 + 9x_2 \rightarrow \max$ $5x_1 + 7x_2 \leq 35$ $4x_1 + 9x_2 \leq 36$ $\forall x_j \geq 0$, цел.
№5	$L = 4x_1 + 4x_2 \rightarrow \max$ $3x_1 + 2x_2 \leq 10$ $x_1 + 2x_2 \leq 9$ $\forall x_j \geq 0$, цел.	№6	$L = 2x_1 + 7x_2 \rightarrow \max$ $4x_1 + 5x_2 \leq 20$ $-x_1 + 2x_2 \leq 1$ $\forall x_j \geq 0$, цел.
№7	$L = 6x_1 + 9x_2 \rightarrow \max$ $8x_1 + 18x_2 \leq 72$	№8	$L = -2x_1 - 3x_2 \rightarrow \min$ $5x_1 + 7x_2 \leq 35$

	$5x_1 + 7x_2 \leq 35$ $\forall x_j \geq 0, \text{ цел.}$		$4x_1 + 9x_2 \leq 36$ $\forall x_j \geq 0, \text{ цел.}$
№9	$L = 9x_1 + 12x_2 \rightarrow \max$ $14x_1 + 9x_2 \leq 51$ $- 6x_1 + 3x_2 \leq 1$ $\forall x_j \geq 0, \text{ цел.}$	№10	$L = x_1 + x_2 \rightarrow \min$ $2x_1 + 4x_2 \geq 7$ $10x_1 + 3x_2 \geq 15$ $\forall x_j \geq 0, \text{ цел.}$
№11	$L = 6x_1 + 8x_2 \rightarrow \max$ $- 12x_1 + 6x_2 \leq 1$ $14x_1 + 9x_2 \leq 51$ $\forall x_j \geq 0, \text{ цел.}$	№12	$L = 9x_1 + 12x_2 \rightarrow \max$ $- 12x_1 + 6x_2 \leq 2$ $14x_1 + 9x_2 \leq 51$ $\forall x_j \geq 0, \text{ цел.}$
№13	$L = 14x_1 + 8x_2 \rightarrow \max$ $6x_1 + 4x_2 \leq 24$ $4x_1 + x_2 \leq 14$ $\forall x_j \geq 0, \text{ цел.}$	№14	$L = 2x_1 + 6x_2 \rightarrow \max$ $10x_1 + 11x_2 \leq 110$ $- x_1 + 2x_2 \leq 7$ $\forall x_j \geq 0, \text{ цел.}$
№15	$L = 4x_1 + 6x_2 \rightarrow \max$ $10x_1 + 14x_2 \leq 70$ $4x_1 + 9x_2 \leq 36$ $\forall x_j \geq 0, \text{ цел.}$	№16	$L = 6x_1 + 8x_2 \rightarrow \max$ $6x_1 - 3x_2 \geq - 1$ $14x_1 + 9x_2 \leq 51$ $\forall x_j \geq 0, \text{ цел.}$
№17	$L = x_1 + x_2 \rightarrow \max$ $2x_1 + 5x_2 \leq 16$ $6x_1 + 5x_2 \leq 27$ $\forall x_j \geq 0, \text{ цел.}$	№18	$L = 2x_1 + 7x_2 \rightarrow \max$ $- 2x_1 + 4x_2 \leq 2$ $4x_1 + 5x_2 \leq 20$ $\forall x_j \geq 0, \text{ цел.}$
№19	$L = x_1 + x_2 \rightarrow \max$ $4x_1 + 5x_2 \leq 18$ $- x_1 + 2x_2 \leq 1$ $\forall x_j \geq 0, \text{ цел.}$	№20	$L = 4x_1 + 14x_2 \rightarrow \max$ $8x_1 + 10x_2 \leq 40$ $- 2x_1 + 4x_2 \leq 2$ $\forall x_j \geq 0, \text{ цел.}$
№21	$L = 3x_1 + 3x_2 \rightarrow \min$ $4x_1 + 8x_2 \geq 14$ $10x_1 + 3x_2 \geq 15$ $\forall x_j \geq 0, \text{ цел.}$	№22	$L = x_1 + x_2 \rightarrow \max$ $x_1 + x_2 \leq 7$ $2x_1 + 11x_2 \leq 38$ $4x_1 - 5x_2 \leq 5$ $\forall x_j \geq 0, \text{ цел.}$

Глава 8. НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

8.1. Характеристика задач

Методы нелинейного программирования применяются для решения задач с нелинейными функциями переменных.

В общем случае задача математического программирования записывается в виде:

$$\left. \begin{aligned} f(\mathbf{x}) &\rightarrow \max(\min); \\ \varphi_i(\mathbf{x}) &\leq (\geq) 0, & i = \overline{1, m_1}; \\ \psi_k(\mathbf{x}) &= 0, & k = \overline{1, m_2}. \end{aligned} \right\} \quad (8.1)$$

Если хотя бы одна функция в модели (8.1) нелинейна, имеем задачу нелинейного программирования (НП). Размерность задачи характеризуется размерностью вектора переменных n и числом условий m_1+m_2 . Однако сложность задачи определяется не столько размерностью, сколько свойствами функций цели и ограничений.

Разнообразие задач НП очень велико. Универсальных методов решения таких задач не существует. Имеется весьма ограниченное число точных методов и намного больше приближенных.

Наиболее развиты методы решения задач *выпуклого программирования*. К этому классу относятся задачи НП с выпуклым допустимым множеством и выпуклой целевой функцией при минимизации или вогнутой при максимизации. Допустимое множество выпуклое, если все функции ψ_k линейные и φ_i выпуклы при неравенстве \leq или вогнуты при \geq . Например, условие $x_1^2+x_2^2 \leq r^2$ порождает выпуклое множество, пересечение которого с прямой $x_1+x_2=0$ дает тоже выпуклое множество. Очевидно, что задачи ЛП относятся к этому классу. Главная особенность задач выпуклого программирования в том, что они *унимодальны*, то есть любой их локальный оптимум является глобальным. Для ряда задач выпуклого программирования с дифференцируемыми функциями разработаны точные методы. Наибольшие сложности возникают при решении многоэкстремальных задач, которые по определению не относятся к классу выпуклых.

Важным классом НП являются задачи *квадратичного программирования*. В них целевая функция представляет собой сумму линейной и квадратичной форм, а все условия линейные. При выпуклости (вогнутости) квадратичной формы они являются частным случаем задач выпуклого программирования.

В нелинейном программировании выделяют также задачи *сепарабельного программирования*. Это задачи, в которых все функции сепарабельны. Функция сепарабельна, если она представляется в виде суммы функций отдельных переменных. Линейная функция – частный случай сепарабельной. Сепарабельная задача может быть одновременно и задачей выпуклого программирования.

Задачи *геометрического программирования* составляют отдельный класс НП. Все функции в таких задачах являются *позиномами*. В общем виде позиномом называется функция

$$f(\mathbf{x}) = \sum_{k=1}^l C_k \cdot x_1^{\alpha_{k1}} \cdot x_2^{\alpha_{k2}} \cdot \dots \cdot x_n^{\alpha_{kn}}, \quad \forall C_k > 0,$$

в которой α_{kj} – любые действительные числа.

Задачи геометрического программирования ставятся только на минимум:

$$\left. \begin{aligned} f(\mathbf{x}) &\rightarrow \min, \\ \varphi_i(\mathbf{x}) &\leq b_i, \\ \mathbf{x} &> 0. \end{aligned} \right\}$$

Такие задачи чаще возникают в конструкторских разработках. Для них разработаны специальные методы.

Пример 8.1. Спроектировать открытый контейнер с прямоугольными стенками и днищем для перевозки из карьера гравия объемом V , если стоимость одной перевозки C не зависит от объема контейнера, а стоимость 1 м^2 днища равна a , передней и задней стенок – b , боковых стенок – d .

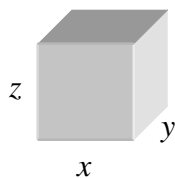


Рис. 8.1

Пусть x – ширина, y – глубина и z – высота контейнера (рис.8.1). Тогда целевая функция, суммарные затраты, запишется в виде

$$Z = \frac{V \cdot C}{x \cdot y \cdot z} + x \cdot y \cdot a + 2 \cdot x \cdot z \cdot b + 2 \cdot y \cdot z \cdot d \rightarrow \min.$$

Получили типичный позином. ▲

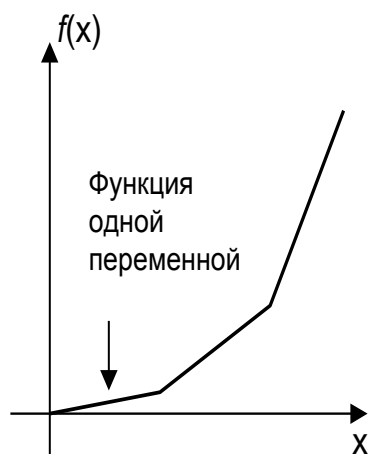


Рис. 8.2

Кусочно-линейное программирование включает специальные методы решения задач с кусочно-линейными функциями. В частности, такими являются функции

$$f(\mathbf{x}) = \max_i f_i(\mathbf{x}) \text{ и } f(\mathbf{x}) = \min_i f_i(\mathbf{x}),$$

если все $f_i(\mathbf{x})$ – линейные функции. Первая из них – выпуклая (рис. 8.2), вторая – вогнутая. Задачи с такими функциями могут входить в класс задач выпуклого программирования. Их решение строится на преобразовании модели к

линейному виду с последующим применением методов ЛП.

К линейным сводятся также задачи *дробно-линейного программирования*. Они отличаются от линейных только дробной целевой функцией, числитель и знаменатель которой – линейные функции.

8.2. Условия оптимальности

Важным свойством задач НП является дифференцируемость функций критерия и ограничений. Для таких задач получены условия оптимальности, на основе которых строится ряд методов НП.

Пусть дана задача в виде

$$\left. \begin{aligned} f(\mathbf{x}) &\rightarrow \max; \\ \varphi_i(\mathbf{x}) &\geq 0, \quad i = \overline{1, m_1}; \\ \psi_k(\mathbf{x}) &= 0, \quad k = \overline{1, m_2}; \\ \mathbf{x} &\geq 0. \end{aligned} \right\} \quad (8.2)$$

Обобщенный метод множителей Лагранжа применим и к условиям-неравенствам. Запишем функцию Лагранжа (регулярную) для задачи (8.2)

$$F(\mathbf{X}, \Lambda) = f(\mathbf{X}) + \sum_{i=1}^{m_1} \lambda_i \varphi_i(\mathbf{X}) + \sum_{k=1}^{m_2} \lambda_k \psi_k(\mathbf{X}). \quad (8.3)$$

В теории НП показано, что эта функция имеет седловую точку $(\mathbf{X}^*, \Lambda^*)$ с максимумом по \mathbf{X} и минимумом по Λ :

$$F(\mathbf{X}, \Lambda^*) \leq F(\mathbf{X}^*, \Lambda^*) \leq F(\mathbf{X}^*, \Lambda). \quad (8.4)$$

Поэтому задача (8.2) сводится к отысканию седловой точки функции (8.3).

Теорема

Пусть f , φ_i и ψ_k – дифференцируемые функции и справедливо свойство *Слейтера* (то есть найдутся такие $\mathbf{X} \in D$, что неравенства φ_i будут строгими). $F(\mathbf{X}, \Lambda)$ – соответствующая функция Лагранжа. Тогда для того чтобы вектор \mathbf{X}^* являлся решением общей задачи максимизации (8.2) необходимо выполнение условий

1) по \mathbf{X} :

$$\left. \frac{\partial F}{\partial X_j} \right|_{\substack{\mathbf{x} = \mathbf{x}^* \\ \Lambda = \Lambda^*}} \leq 0, \quad j = \overline{1, n}; \quad (8.5)$$

$$\left. \frac{\partial F}{\partial X_j} \right|_{\substack{\mathbf{x} = \mathbf{x}^* \\ \Lambda = \Lambda^*}} \cdot X_j^* = 0, \quad j = \overline{1, n}; \quad (8.6)$$

$$\forall X_j^* \geq 0;$$

2) по Λ :

$$\varphi_i(\mathbf{X}^*) \geq 0, \quad i = \overline{1, m_1}; \quad \psi_k(\mathbf{X}^*) = 0, \quad k = \overline{1, m_2}; \quad (8.7)$$

$$\lambda_i^* \cdot \varphi_i(\mathbf{X}^*) = 0, \quad i = \overline{1, m_1}, \quad \forall \lambda_i^* \geq 0; \quad (8.8)$$

$$\lambda_k^* \cdot \psi_k(\mathbf{X}^*) = 0, \quad k = \overline{1, m_2}. \quad (8.9)$$

Приведенные условия оптимальности называются условиями Куна-Таккера. Опуская строгое доказательство, приведем логическое обоснование выражений (8.5)-(8.9).

По существу они являются обобщением классических условий экстремума, определяющих стационарные точки. Условие (8.5) содержит неравенство, так как неотрицательность вектора \mathbf{X} означает, что максимум может быть либо при положительном \mathbf{X} и тогда производная F по \mathbf{X} обязательно равна нулю (случай 1 на рис. 8.3), либо при $\mathbf{X}=0$ и тогда эта

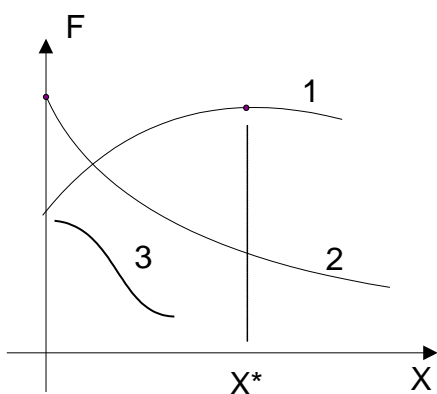


Рис. 8.3

производная может быть как равной нулю, так и отрицательной (случаи 2 и 3 на рис. 8.3). Этим же объясняются условия дополняющей нежесткости (8.6): в точке максимума равны нулю либо \mathbf{X} , либо производная, либо вместе.

Выражения (8.7)-(8.9) можно обосновать аналогично, если учесть, что по Λ рассматривается минимум F и

$$\frac{\partial F}{\partial \lambda_i} = \varphi_i(\mathbf{X}^*), \quad \frac{\partial F}{\partial \lambda_k} = \psi_k(\mathbf{X}^*).$$

Применив условия Куна-Таккера к задаче ЛП, получим равенства второй основной теоремы двойственности как частный случай условий дополняющей нежесткости, а двойственные переменные – как частный случай λ .

Особую роль условия Куна-Таккера играют в решении задач выпуклого программирования, так как для них они являются не только необходимыми, но и достаточными. В следующем разделе это свойство будет использовано для построения точного метода.

8.3. Квадратичное программирование

Задачи квадратичного программирования (КП) имеют место, если целевая функция – сумма линейной и квадратичной форм, а все условия линейные.

Например, в задаче с двумя переменными целевая квадратичная функция записывается следующим образом:

$$f(x_1, x_2) = \boxed{d_1 x_1 + d_2 x_2} + \boxed{1/2(C_{11}x_1^2 + C_{12}x_1x_2 + C_{21}x_1x_2 + C_{22}x_2^2)}.$$

↑
линейная форма
↑
квадратичная форма

В векторной форме она принимает вид

$$f(x_1, x_2) = [d_1, d_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{1}{2} \cdot [x_1, x_2] \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Обобщая на случай многих переменных, получаем:

$$f(\mathbf{X}) = \mathbf{D}^T \cdot \mathbf{X} + \frac{1}{2} \cdot \mathbf{X}^T \cdot \mathbf{C} \cdot \mathbf{X};$$

$$\mathbf{D} = \begin{bmatrix} d_1 \\ d_2 \\ \dots \\ d_n \end{bmatrix}; \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}; \quad \mathbf{C} = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \dots & \dots & \dots & \dots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{bmatrix}.$$

Матрица \mathbf{C} – квадратная, диагонально-симметричная ($C_{ij}=C_{ji}$).

В целом задача квадратичного программирования ставится в виде:

$$f(\mathbf{X}) = \mathbf{D}^T \cdot \mathbf{X} + \frac{1}{2} \cdot \mathbf{X}^T \cdot \mathbf{C} \cdot \mathbf{X} \rightarrow \max; \quad (8.10)$$

$$\mathbf{B} - \mathbf{A}\mathbf{X} \geq 0; \quad (8.11)$$

$$\mathbf{X} \geq 0. \quad (8.12)$$

Чтобы она являлась задачей выпуклого программирования, целевая функция (8.10) должна быть вогнутой.

Свойства функции определяются матрицей \mathbf{C} . Для вогнутости функции необходимо, чтобы матрица \mathbf{C} была отрицательно определенной (строгая вогнутость) или отрицательно полуопределенной. Матрица \mathbf{C} отрицательно определенная, если для всех ненулевых \mathbf{X} справедливо $\mathbf{X}^T \mathbf{C} \mathbf{X} < 0$, и отрицательно полуопределенная, если $\mathbf{X}^T \mathbf{C} \mathbf{X} \leq 0$. В случае минимизации целевая функция должна быть выпуклой, что имеет место при положительно определенной или положительно полуопределенной матрице \mathbf{C} . Практически определить свойство квадратичной функции можно с помощью достаточных условий экстремума: если функция в стационарной точке имеет максимум, она вогнутая, а если минимум, то выпуклая.

Далее будем полагать, что условия вогнутости функции выполняются. Тогда решение задачи КП можно найти на основе следующей теоремы.

Теорема

Для того чтобы вектор \mathbf{X}^* являлся решением задачи (8.10)-(8.12), необходимо и достаточно существования таких неотрицательных m -мерных векторов \mathbf{W} и $\mathbf{\Lambda}$ и неотрицательного n -мерного вектора \mathbf{V} , которые удовлетворяют следующей системе уравнений:

$$\mathbf{D} + \mathbf{C} \cdot \mathbf{X}^* - \mathbf{A}^T \cdot \mathbf{\Lambda} + \mathbf{V} = 0, \quad (8.13)$$

$$\mathbf{B} - \mathbf{A} \cdot \mathbf{X}^* - \mathbf{W} = 0, \quad (8.14)$$

$$\mathbf{V}^T \cdot \mathbf{X}^* = 0, \quad (8.15)$$

$$\mathbf{W}^T \cdot \mathbf{\Lambda} = 0. \quad (8.16)$$

Покажем, что теорема выводится из условий Куна-Таккера. Функция Лагранжа для рассматриваемой задачи имеет вид

$$F(\mathbf{X}, \mathbf{\Lambda}) = \mathbf{D}^T \cdot \mathbf{X} + \frac{1}{2} \cdot \mathbf{X}^T \cdot \mathbf{C} \cdot \mathbf{X} + \mathbf{\Lambda}^T \cdot (\mathbf{B} - \mathbf{A}\mathbf{X}).$$

Записываем условия (8.5):

$$\frac{\partial F}{\partial \mathbf{X}} = \mathbf{D} + \mathbf{C} \cdot \mathbf{X}^* - \mathbf{A}^T \mathbf{\Lambda} \leq 0.$$

Введя в это неравенство неотрицательный вектор дополнительных переменных \mathbf{V} , получаем (8.13). (8.14) – это исходное условие задачи после приведения его к равенству введением неотрицательного вектора дополнительных переменных \mathbf{W} . Очевидно, что производная $\frac{\partial F}{\partial \mathbf{X}} < 0$, когда

дополнительная переменная $V > 0$ и $\frac{\partial F}{\partial \mathbf{X}} = 0$, когда $V = 0$. Таким образом, V играет роль индикатора производной. Поэтому условие дополняющей нежесткости (8.6) принимает вид (8.15). Аналогична взаимосвязь вектора \mathbf{W} с производной F по $\mathbf{\Lambda}$, и отсюда имеем второе условие дополняющей нежесткости (8.16).

Система уравнений (8.13)-(8.16) – нелинейная, так как нелинейны (8.15) и (8.16). Она содержит $(m + n + 2)$ уравнений и $2 \cdot (m + n)$ неизвестных \mathbf{X}^* , $\mathbf{\Lambda}$, \mathbf{V} и \mathbf{W} .

Так как $\mathbf{V}^T \mathbf{X} = \sum_{j=1}^n v_j \cdot x_j$ и векторы \mathbf{V} и \mathbf{X} неотрицательны, из (8.15) следует, что по крайней мере n переменных из v_j и x_j равны 0. Аналогично из (8.16) вытекает, что равны нулю не менее m переменных из w_i и λ_i . Таким образом, в решении системы (8.13)-(8.14) положительными могут быть не более $(m + n)$ переменных. Это свойство системы дает ключ к решению.

Действительно, линейная система (8.13), (8.14) содержит $n+m$ уравнений и $2(n + m)$ неизвестных. Но известно, что в искомом решении число положительных переменных не превышает $(m + n)$. Следовательно, это допустимое базисное решение (опорный план) системы (8.13), (8.14). Поэтому искать решение задачи КП нужно только среди опорных планов этой системы. Такие решения находятся методами линейного программирования. Опорный план системы (8.13), (8.14), удовлетворяющий условиям (8.15), (8.16), будет оптимальным решением задачи КП.

Перепишем уравнения (8.13), (8.14) в обычном виде:

$$\begin{cases} \mathbf{C} \cdot \mathbf{X}^* - \mathbf{A}^T \mathbf{\Lambda} + \mathbf{V} = -\mathbf{D}, \\ \mathbf{A} \cdot \mathbf{X}^* + \mathbf{W} = \mathbf{B}. \end{cases} \quad (8.17)$$

Если вектор \mathbf{D} – неположительный, а вектор \mathbf{B} – неотрицательный, то начальное базисное решение $\mathbf{V} = -\mathbf{D}$, $\mathbf{W} = \mathbf{B}$ удовлетворяет условиям (8.15),

(8.16) и, значит, является оптимальным решением задачи КП. Однако, как правило, вектор D имеет положительные компоненты и такое начальное решение оказывается недопустимым. В этом случае, ориентируясь на использование прямого симплекс-метода, строится искусственное начальное решение: в уравнения (8.17) с отрицательной правой частью вводятся искусственные переменные y_k и они вместе с неотрицательными v_j и w_i образуют базисное решение. В качестве критерия линейной задачи принимается сумма искусственных переменных:

$$L_{\text{иск}} = \sum y_k \rightarrow \min.$$

Для выполнения условий дополняющей нежесткости (8.15)-(8.16) алгоритм симплекс-метода дополняется правилом ограниченного ввода:

если в базисном решении имеется v_j , то не может вводиться x_j (с тем же индексом) и наоборот;

если в базисном решении имеется w_i , то не может вводиться λ_i (с тем же индексом) и наоборот.

Иначе говоря, в базисном решении не могут находиться одновременно переменные v, x (w, λ) с одинаковыми индексами. Если по оценкам претендентом на ввод является переменная, которую согласно правилу нельзя вводить, в базисное решение вводится другая переменная с положительной оценкой.

Признаком выполнения условий теоремы (8.13)-(8.16) и, следовательно, оптимальности решения задачи КП является равенство нулю всех искусственных переменных или $L_{\text{иск}}=0$.

Очевидно, что рассмотренный метод находит за конечное число шагов глобальное решение задачи КП с вогнутой функцией цели. При строгой вогнутости задача имеет одно решение, при нестрогой вогнутости возможно множество решений. Если функция не является вогнутой, метод находит некоторый локальный максимум.

Пример 8.2. Найти решение следующей задачи КП:

$$\begin{aligned} f &= 10x_1 + 20x_2 + x_1 \cdot x_2 - 2x_1^2 - 2x_2^2 \rightarrow \max, \\ 8 - x_2 &\geq 0, \\ 9 - x_1 - x_2 &\geq 0, \\ x_1, x_2 &\geq 0. \end{aligned}$$

Перепише целевую функцию в векторной форме:

$$f = [10, 20] \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{1}{2} \cdot [x_1, x_2] \cdot \begin{bmatrix} -4 & 1 \\ 1 & -4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

По матрице C (гессиану) проверяем достаточные условия: $\Delta_1 = -4 < 0$, $\Delta_2 = 16 - 1 > 0$. Значит, f имеет максимум и строго вогнута.

Записываем первую систему уравнений (8.17):

$$\begin{bmatrix} -4 & 1 \\ 1 & -4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} + \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = - \begin{bmatrix} 10 \\ 20 \end{bmatrix}$$

или

$$\begin{cases} -4x_1 + x_2 - \lambda_2 + V_1 = -10, \\ x_1 - 4x_2 - \lambda_1 - \lambda_2 + V_2 = -20. \end{cases}$$

Добавляем вторую:

$$\begin{cases} x_2 + W_1 = 8, \\ x_1 + x_2 + W_2 = 9. \end{cases}$$

Для образования начального базисного решения вводим в первую систему искусственные переменные y_1 и y_2 :

$$\begin{cases} 4x_1 - x_2 + \lambda_2 - V_1 + y_1 = 10, \\ -x_1 + 4x_2 + \lambda_1 + \lambda_2 - V_2 + y_2 = 20. \end{cases}$$

Критерий линейной задачи:

$$L_{\text{иск}} = y_1 + y_2 \rightarrow \min.$$

Базисные переменные в начальном решении: y_1, y_2, w_1 и w_2 .

Заполняем начальную симплекс-таблицу.

		0	0	0	0	0	0	0	0	0	1	1	
C_{si}	Баз.	A_0	x_1	x_2	λ_1	λ_2	v_1	v_2	w_1	w_2	y_1	y_2	θ
1	y_1	10	4	-1	0	1	-1	0	0	0	1	0	5/2
1	y_2	20	-1	4	1	1	0	-1	0	0	0	1	-
0	w_1	8	0	1	0	0	0	0	1	0	0	0	-
0	w_2	9	1	1	0	0	0	0	0	1	0	0	9
	Δ	30	3	3	1	2	-1	-1	0	0	0	0	
	z	30	3	3	1	2	-1	-1	0	0	0	0	

Выполнив симплекс-преобразования с учетом правила ограниченного ввода, находим оптимальное решение задачи КП. На рис. 8.4 показано

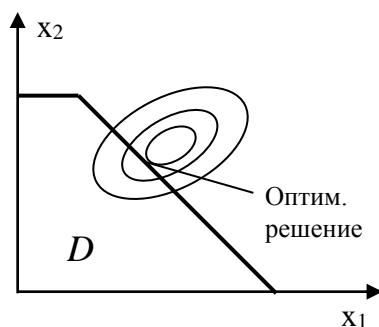


Рис. 8.4

допустимое множество задачи и линии уровня критерия. Оптимум достигается на границе допустимого множества в точке касания с линией уровня.

Этот пример показывает, что в задачах КП в отличие от линейного программирования оптимальное решение может находиться как в любой точке границы (не только в вершине), так и внутри

допустимого множества в случае совпадения с безусловным максимумом.

В заключение отметим, что задача КП может использоваться в качестве аппроксимации при нелинейности критерия, отличной от квадратичной, так как в небольшой области гладкие функции с высокой точностью могут быть представлены квадратичной функцией. Ряд последовательных аппроксимаций с решением задачи КП позволяют найти решение исходной нелинейной задачи с необходимой точностью.

8.4. Сепарабельное программирование (СП)

В сепарабельном программировании рассматриваются задачи, в которых целевая функция и все функции ограничений сепарабельны.

Напомним, что функция многих переменных сепарабельна, если она имеет вид суммы функций отдельных переменных:

$$f(x_1, x_2, \dots, x_n) = \sum_{j=1}^m f_j(x_j). \quad (8.18)$$

Линейные функции всегда сепарабельны и поэтому линейное программирование можно рассматривать как частный случай сепарабельного.

Решение задач СП основано на преобразовании в задачи линейного программирования путем аппроксимации нелинейных функций кусочно-линейными. Таким образом, исходная нелинейная задача заменяется аппроксимирующей линейной. Поэтому рассматриваемый метод является приближенным, а точность решения напрямую зависит от точности аппроксимации и теоретически может быть сколь угодно высокой.

Существует два основных способа записи аппроксимирующей задачи, отличающихся формой представления исходных переменных: в λ - или в δ -постановке.

λ - постановка

Предполагается, что переменные, которые входят в модель *нелинейно*, ограничены снизу и сверху:

$$d_j \leq x_j \leq D_j. \quad (8.19)$$

Для кусочно-линейной аппроксимации в этом диапазоне выбираются узловые точки, чаще в той части, где сильнее нелинейность функции. При этом первый узел совпадает с нижней границей, а последний – с верхней:

$$X_{j1} = d_j, \quad X_{jr_j+1} = D_j,$$

где r_j – число интервалов по переменной x_j (r_j+1 – число узлов). Тогда рассматриваемая переменная x_j может быть выражена через новые переменные λ_{jk} в виде

$$x_j = \sum_{k=1}^{r_j+1} X_{jk} \cdot \lambda_{jk}, \quad (8.20)$$

$$\sum_k \lambda_{jk} = 1, \quad (8.21)$$

$$\forall \lambda_{jk} \geq 0. \quad (8.22)$$

Выражение (8.20) называют *уравнением сетки*. С учетом (8.21) и (8.22) оно представляет переменную x_j в диапазоне (8.19) без потери точности. С использованием узловых точек и новых переменных кусочно-линейная функция, аппроксимирующая $f_j(x_j)$, записывается в виде

$$\hat{f}_j = \sum_k f_j(X_{jk}) \cdot \lambda_{jk}, \quad (8.23)$$

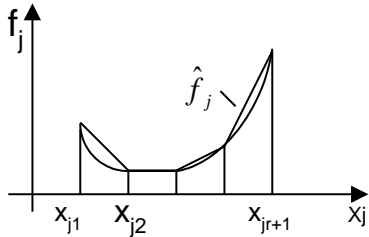


Рис. 8.5

где $f_j(X_{jk})$ — значение функции в узловых точках (рис. 8.5). Очевидно, что \hat{f}_j — функция, линейная относительно λ_{jk} . Пусть N — множество индексов нелинейных $f_j(x_j)$. Тогда функция, аппроксимирующая $f(X)$, имеет вид

$$\hat{f}(X, \Lambda) = \sum_{j \in N} f_j(x_j) + \sum_{j \in N} \sum_k f_j(X_{jk}) \cdot \lambda_{jk}. \quad (8.24)$$

Итак, чтобы построить линейную аппроксимирующую модель, необходимо:

1. для каждой переменной, входящей нелинейно, записать уравнение сетки;
2. во всей модели заменить переменные из п.1, входящие в линейные f_j , соответствующими уравнениями сетки;
3. все функции, содержащие нелинейности, представить в виде (8.24);
4. добавить ограничения (8.21), (8.22) для всех новых переменных.

Если переменная x_j входит нелинейно в несколько функций, узлы сетки выбираются с учетом нелинейности всех таких функций, так как для одной переменной может быть только одно уравнение сетки.

Поясним запись ограничений. Пусть имеется исходное ограничение

$$\sum \varphi_{ij}(x_j) \leq b_i$$

со всеми нелинейными φ_{ij} . Тогда после аппроксимации оно принимает вид

$$\sum_j \sum_{k=1}^{r_j+1} \varphi_{ij}(X_{jk}) \cdot \lambda_{jk} \leq b_i.$$

В общем случае левая часть ограничения записывается аналогично (8.24).

Хотя аппроксимирующая задача линейная, получаемое на ней решение не всегда является приближением к решению исходной задачи. Дело в том, что одно и то же значение x_j можно получить по уравнению сетки при разных λ_{jk} , то есть представить через разные пары узлов. Например, некоторое значение x_j можно выразить через смежные узлы, в интервале которых находится значение, а можно через любую другую пару узлов, лежащих слева и справа, в том числе через первый и последний

узел. Во всех случаях, кроме первого аппроксимация функции будет грубой и тем грубее, чем дальше отстоят узлы от данного значения x_j .

Отсюда следует *правило смежных весов*: из одного уравнения сетки отличными от нуля могут быть не более 2-х переменных λ_{jk} со смежными значениями k .

Если аппроксимирующая задача является задачей выпуклого программирования, то это правило выполняется автоматически и решение находится методом ЛП без каких-либо дополнений. Оптимальное решение аппроксимирующей задачи будет приближением глобального решения исходной задачи.

В противном случае алгоритм ЛП должен включать правило ограниченного ввода:

если в базисном решении находится λ_{jk} , то допустимыми для ввода могут быть только λ_{jk+1} или λ_{jk-1} .

При этом нельзя утверждать, что получаемое решение является приближением к глобальному оптимуму исходной задачи. Скорее оно будет приближением локального оптимума.

Свойства задачи зависят от всех функций модели:

1. Если все ограничения линейные, то для выпуклости задачи достаточно, чтобы были вогнутыми все f_j критерия (выпуклы при минимизации)

2. При нелинейности критерия и ограничений для выпуклости задачи должны быть вогнуты все f_j и выпуклы все φ_{ij} .

3. Если хотя бы одна f_j не вогнута при максимизации и/или одна φ_{ij} не выпукла, задача не является выпуклой.

Заметим, что, если все функции кусочно-линейные, переход к новым переменным не связан с потерей точности и при выполнении условий задач выпуклого программирования получаемое решение является точным и глобальным.

Пример 8.3. Задача

$$\begin{aligned} f &= 6x_1 - x_1^2 + 7x_2 \rightarrow \max, \\ 2x_1^2 - 5x_1 + 3x_2^2 &\leq 8, \\ 1 \leq x_1 < 4, \quad x_2 &\geq 0 \end{aligned}$$

является задачей сепарабельного программирования. Здесь

$$\begin{aligned} f_1(x_1) &= 6x_1 - x_1^2; \quad f_2(x_2) = 7x_2; \\ \varphi_{11}(x_1) &= 2x_1^2 - 5x_1; \quad \varphi_{12}(x_2) = 3x_2^2. \end{aligned}$$

Так как $f_1(x_1)$ и $f_2(x_2)$ – вогнутые, а $\varphi_{11}(x_1)$ и $\varphi_{12}(x_2)$ – выпуклые, имеем задачу выпуклого программирования. Обе переменные входят нелинейно, поэтому нужно строить две сетки. Оценим верхний предел x_2 : находим $\min \varphi_{11} = -3.125$, затем из ограничения получаем максимально возможное значение $x_2 = 1.93$. Берем $D_2 = 2$. Пусть узловыми будут значения по x_1 : 1, 2,

3, 4; по x_2 : 0, 1, 2. Записываем уравнения сеток: $x_1 = \lambda_{11} + 2\lambda_{12} + 3\lambda_{13} + 4\lambda_{14}$, $x_2 = \lambda_{22} + 2\lambda_{23}$. В итоге получаем модель аппроксимирующей задачи в виде

$$\begin{aligned} \hat{f} &= 5\lambda_{11} + 8\lambda_{12} + 9\lambda_{13} + 8\lambda_{14} + 7\lambda_{22} + 14\lambda_{23} \rightarrow \max, \\ -3\lambda_{11} - 2\lambda_{12} + 3\lambda_{13} + 12\lambda_{14} + 3\lambda_{22} + 12\lambda_{23} &\leq 8, \\ \lambda_{11} + 2\lambda_{12} + 3\lambda_{13} + 4\lambda_{14} &\geq 1, \\ \lambda_{11} + 2\lambda_{12} + 3\lambda_{13} + 4\lambda_{14} &\leq 4, \\ \lambda_{11} + \lambda_{12} + \lambda_{13} + \lambda_{14} &= 1, \\ \lambda_{21} + \lambda_{22} + \lambda_{23} &= 1, \\ \forall \lambda_{jk} &\geq 0. \end{aligned}$$

Эта задача решается любым универсальным методом ЛП без добавления правила ограниченного ввода.

δ-постановка

Построение аппроксимирующей задачи основано так же на кусочно-линейном приближении, но меняется уравнение сетки. По узлам сетки вычисляются расстояния между смежными узлами (длины интервалов)

$$\delta_{jk} = X_{jk+1} - X_{jk}$$

и уравнение сетки записывается в виде

$$x_j = d_j + \sum_{k=1}^{r_j} \delta_{jk} y_{jk}; \quad (8.25)$$

$$0 \leq y_{jk} \leq 1, \quad (8.26)$$

где y_{jk} – новые переменные.

Из представления переменной в виде (8.25), (8.26) следует:

- $x_j = d_j$, когда $\forall y_{jk} = 0$;
- x_j находится в первом интервале, когда $y_{j1} \in (0, 1)$, остальные $y_{jk} = 0$;
- x_j находится во втором интервале, когда $y_{j1} = 1$, $y_{j2} \in (0, 1)$, остальные $y_{jk} = 0$;
- x_j находится в k -ом интервале, когда $y_{j1} = y_{j2} = \dots = y_{jk-1} = 1$, $0 \leq y_{jk} \leq 1$, остальные $y_{jk} = 0$.

Таким образом, для правильной аппроксимации должно выполняться установленное соответствие между значениями переменной x_j и y_{jk} . Это требование аналогично правилу смежных весов. При ином представлении значения x_j будет нарушена кусочно-линейная аппроксимация функции.

Для аппроксимации нелинейной составляющей функции критерия вычисляются разности ее значений в смежных узлах

$$\Delta_{jk} = f_j(X_{jk+1}) - f_j(X_{jk}),$$

с помощью которых записывается аппроксимирующая функция

$$\hat{f}_j(Y_j) = f_j(d_j) + \sum_{k=1}^{r_j} \Delta_{jk} \cdot y_{jk}. \quad (8.27)$$

Тогда функция, аппроксимирующая критерий, имеет вид

$$\hat{f} = \sum_j \hat{f}_j(Y_j).$$

Аналогично аппроксимируются ограничения $\varphi_{ij}(x_j)$:

$$\Delta_{ijk} = \varphi_{ij}(x_{jk+1}) - \varphi_{ij}(x_{jk}),$$

$$\hat{\varphi}_{ij} = \varphi_{ij}(d_j) + \sum_{k=1}^{r_j} \Delta_{ijk} y_{jk}.$$

Как и в λ -постановке, если имеет место задача выпуклого программирования, то требования к переменным y_{jk} выполняются автоматически и полученное решение будет приближенным глобальным решением исходной задачи. В противном случае, необходимо придерживаться правила ограниченного ввода относительно переменных y_{jk} : если первые k переменных равны единице, вводить можно только y_{jk+1} .

При практическом решении сепарабельных задач сначала можно взять малое число узлов и получить приближенное оптимальное решение. Затем в качестве исходных принять интервалы, на которых лежат оптимальные x_j , и выполнить аппроксимацию функций только на этих интервалах с малыми расстояниями между узлами. Такой способ снижает размерность решаемых задач и повышает точность получаемого решения.

Следует заметить, что в ряде случаев несепарабельная функция может быть преобразована к сепарабельной. Способ преобразования зависит от структуры функции. Например, произведение двух сепарабельных функций $S(X) \cdot T(X)$ можно привести к сепарабельному виду, заменив его переменной v с дополнительными равенствами

$$S(X) = z - y; \quad T(X) = z + y.$$

Тогда $v = (z - y)(z + y) = z^2 - y^2$ – сепарабельная функция. Так функция $f = x_1 + x_2 \cdot x_3^2$ заменяется на сепарабельную $f = x_1 + v$ с дополнительными сепарабельными ограничениями

$$\begin{cases} x^2 - (z - y) = 0; \\ x_3^2 - (z + y) = 0; \\ v - z^2 + y^2 = 0. \end{cases}$$

Пример 8.4. Покажем, что некоторые стохастические задачи могут сводиться к сепарабельным. Стохастические модели описывают ситуации выбора решения в условиях риска, обусловленного влиянием случайных факторов. Предполагается, что закон распределения случайных величин известен.

Пусть зависимости от искомым переменных линейны, но коэффициенты критерия и ограничений зависят от случайной величины ω (состояния среды). В этом случае в качестве критерия берется обычно математическое ожидание линейной формы $M(L) = M[C^T(\omega)X] = \bar{C}^T X$, а

запись ограничений зависит от требований к их выполнению. При допустимости некоторых нарушений условий задачи ограничения записываются в вероятностной форме:

$$P\left[\sum a_{ij}(\omega)x_j \leq b_i(\omega)\right] \geq p_i, \quad i = \overline{1, m}.$$

где p_i – заданное значение вероятности. выполнения i -го условия. Такое ограничение заменяется эквивалентным детерминированным условием

$$\sum_{j=1}^n \bar{a}_{ij}x_j + t_{p_i} \sqrt{\sum_{j=1}^n \sigma_{ij}^2 x_j^2 + \sigma_i^2} \leq \bar{b}_i, \quad (*)$$

где \bar{a}_{ij}, \bar{b}_i – математические ожидания, σ_{ij} – дисперсия a_{ij} , σ_i – дисперсия b_i , $t_{p_i} = t(p_i)$ – значение функции, обратной функции распределения (например, нормального).

В результате детерминированная модель стохастической задачи включает линейный критерий и существенно нелинейные ограничения (*). Очевидно, что она не является сепарабельной. Сделаем простое преобразование. Обозначим

$$S_i = \sqrt{\sum_j \sigma_{ij}^2 x_j^2 + \sigma_i^2}.$$

Тогда каждое ограничение (*) заменяется двумя условиями:

$$\begin{aligned} \sum_{j=1}^n \bar{a}_{ij}x_j + t_{p_i} S_i &\leq \bar{b}_i, \\ S_i^2 - \sum \sigma_{ij}^2 x_j^2 - \sigma_i^2 &= 0. \end{aligned}$$

Первое из них – линейное, а второе – сепарабельное. Таким образом, стохастическая задача приведена к сепарабельной.

Примечание. Если случайным является только вектор ограничений, то, как следует из (*), стохастическая задача сводится к линейной.

8.5. Задачи дробно-линейного программирования

Если целевая функция представляет собой отношение линейных функций, а все условия линейные, то задача относится к классу задач дробно-линейного программирования.

В общем случае целевая функция имеет вид

$$f = \frac{c_0 + \sum_{j=1}^n c_j x_j}{d_0 + \sum_{j=1}^n d_j x_j}. \quad (8.28)$$

Такая функция легко преобразуется в линейную, если ее знаменатель при всех допустимых значениях переменных строго положителен. Для этого введем новую переменную r следующим образом

$$\frac{1}{d_0 + \sum_{j=1}^n d_j x_j} = r. \quad (8.29)$$

Очевидно, что при оговоренном условии она может быть только больше нуля. Тогда функция (8.28) принимает вид

$$f = c_0 r + \sum_{j=1}^n c_j x_j r.$$

Произведя замену произведения переменных

$$x_j r = y_j, \quad (8.30)$$

окончательно имеем

$$f = c_0 r + \sum_{j=1}^n c_j y_j. \quad (8.31)$$

Получили линейную функцию от n неотрицательных переменных y_j и одной положительной переменной r . Эта функция должна рассматриваться вместе с условием, следующим из (8.29):

$$d_0 r + \sum_{j=1}^n d_j x_j r = 1$$

или после замены (8.30)

$$d_0 r + \sum_{j=1}^n d_j y_j = 1. \quad (8.32)$$

Чтобы завершить построение эквивалентной линейной модели, следует ограничения задачи записать в новых переменных. Для этого умножим обе части каждого ограничения

$$\sum_j a_{ij} x_j \leq b_i$$

на r :

$$\sum_j a_{ij} x_j r \leq b_i r$$

и, произведя замену, получаем

$$\sum_j a_{ij} y_j - b_i r \leq 0, \quad i = \overline{1, m}. \quad (8.33)$$

В результате преобразований имеем задачу линейного программирования с критерием (8.31), ограничениями (8.32), (8.33) и переменными $r > 0$, $y_j \geq 0$, $\forall j$. Получив ее решение одним из методов ЛП, вычисляем исходные переменные по очевидной формуле

$$x_j^* = \frac{y_j^*}{r^*}.$$

Чтобы гарантировать выполнение условия положительности знаменателя, целесообразно вводить его явно в модель, то есть добавлять неравенство

$$d_0 + \sum_{j=1}^n d_j x_j \geq \varepsilon,$$

где ε – очень малая положительная константа.

Возможность перехода к линейной задаче геометрически обусловлена тем, что линии уровня дробно-линейной функции описываются линейным уравнением. Действительно, пусть $f = \bar{f} = \text{const}$. Тогда из (8.28) следует

$$d_0 \bar{f} + \bar{f} \sum_j d_j x_j = c_0 + \sum_j c_j x_j$$

или

$$\sum_j (c_j - \bar{f} d_j) x_j = d_0 \bar{f} - c_0. \quad (8.34)$$

Это обычное линейное уравнение относительно x_j . Поэтому линии уровня функции (8.28) в многомерном пространстве – гиперплоскости, а в двухмерном – прямые. Однако с изменением \bar{f} они не перемещаются параллельно, а поворачиваются вокруг множества вращения.

Множество вращения – это множество точек размерности $n-2$, образованное пересечением нулевых линий уровня числителя и знаменателя:

$$\begin{cases} c_0 + \sum_j c_j \cdot x_j = 0, \\ d_0 + \sum_j d_j \cdot x_j = 0. \end{cases}$$

При $n=2$ оно состоит из одной точки, а при $n=3$ представляет собой прямую (ось вращения), образованную пересечением двух плоскостей.

Пример 8.5. Представим графически следующую задачу

$$\begin{aligned} f = \frac{2x_1 + x_2 - 1}{x_1 + 2x_2 - 1} &\rightarrow \max, ; \\ 3x_1 + 2x_2 &\geq 6, \\ 0 \leq x_1 &\leq 3, \\ 0 \leq x_2 &\leq 3. \end{aligned}$$

Уравнения нулевых линий уровня числителя и знаменателя образуют систему

$$\begin{cases} 2x_1 + x_2 = 1 \\ x_1 + 2x_2 = 1, \end{cases}$$

из которой находим точку вращения: $x_1=x_2=1/3$. На рис. 8.6 это точка А. Нулевые линии показаны пунктиром, а направление поворота, в котором целевая функция возрастает, – стрелками. Отсюда ясно, что оптимальное решение достигается в вершине В: $x_1^* = 2$, $x_2^* = 0$, $L^* = 3$.

8.6. Методы "спуска"

Так называются численные итерационные методы оптимизации, ориентированные на поиск минимума.

Общая схема решения заключается в генерации последовательности приближений, асимптотически сходящейся к минимуму. Как правило, это локальный минимум, но в случае задач выпуклого программирования он является и глобальным. За конечное число итераций методы позволяют получить приближенное решение с заданной наперед точностью. Сходимость метода, как и скорость сходимости, зависят от свойств задачи и начального приближения. Скорость сходимости во многом определяет эффективность метода. Основными показателями эффективности (скорости сходимости) являются число итераций и число вычислений функции при прочих равных условиях. Количество итераций для одной и той же задачи сильно зависит от начального приближения (начальной точки) и требуемой точности. При этом количество необходимых итераций растет гораздо быстрее, чем точность.

При выборе метода следует учитывать свойства целевой функции: унимодальность или многоэкстремальность, дифференцируемость, выпуклость-вогнутость или их отсутствие и т. д. Кроме того, функции могут обладать "неприятными" для метода особенностями, такими как седловые точки и овражность. "Овраг" (при максимизации "гребень") проявляется в том, что вдоль него функция изменяется намного слабее, чем в поперечном направлении. На карте линий уровня он виден по сильной вытянутости линий вдоль "дна" оврага с одновременной "сплюснутостью" в поперечнике. Самым простым примером такой функции является эллипсоид

$$f(x) = \frac{x_1^2}{a^2} + \frac{x_2^2}{b^2}.$$

При $a \gg b$ функция имеет овраг вдоль оси x_1 , которая является его "дном" (рис.8.7).

Чем сильнее овражность и нелинейность "дна", тем в большей

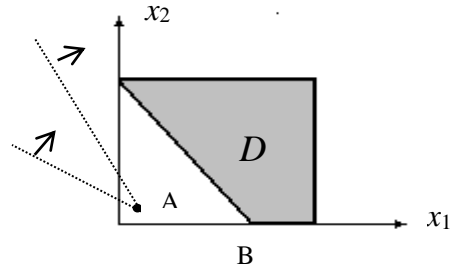


Рис. 8.6

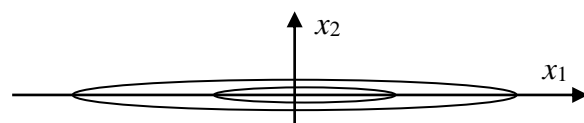


Рис. 8.7

степени замедляется скорость сходимости метода или возможен останов при недостижении окрестности минимума. Поэтому учет таких свойств особенно важен при выборе метода оптимизации.

Различают методы безусловной оптимизации, применяемые для нахождения минимума без ограничений на переменные, и условной оптимизации, когда поиск производится при наличии ограничений.

По информации, используемой для определения направления поиска, выделяют методы

- нулевого порядка или прямые, если вычисляется только значение целевой функции;
- первого порядка или градиентные, использующие первые производные (градиент);
- второго порядка, требующие вычисления также вторых производных;
- случайного поиска, применяющие механизм случайного выбора направления;
- генетические, сочетающие элементы детерминизма и случайности выбора;
- комбинированные.

При оптимизации n -мерных функций часто используются методы одномерной минимизации. Поэтому сначала рассмотрим эти методы, затем многомерные методы безусловной оптимизации и, наконец, методы поиска условного минимума. Все приводимые ниже методы предназначены для минимизации *унимодальных* функций.

8.7. Методы одномерной минимизации

Сначала рассматриваются прямые методы. Первые два из них ориентированы на поиск безусловного минимума, остальные – на поиск минимума на заданном непрерывном интервале.

8.7.1. Метод деления шага пополам

Начальная точка и начальный шаг задаются на основе предварительных знаний о функции. Движение с неизменным шагом в одном направлении продолжается пока значение функции уменьшается. При увеличении значения направление меняется на противоположное и шаг уменьшается в два раза. Поиск заканчивается, когда при очередном шаге значение ухудшилось, а длина шага меньше заданной точности (рис. 8.8). Очевидно, что после прохождения минимума происходят

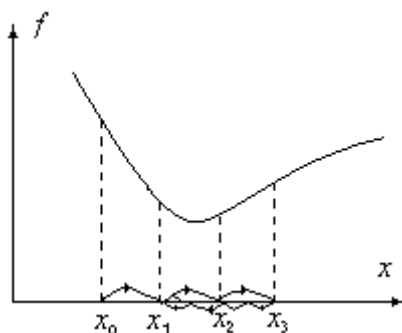


Рис. 8.8

колебания вокруг него с уменьшающейся амплитудой.

Алгоритм.

1. Задать начальную точку x_0 , начальный шаг h_0 и точность ε ; $k=0$.
2. Вычислить $f(x_k)$.
3. $x_{k+1} = x_k + h_k$.
4. Вычислить $f(x_{k+1})$.
5. Если $f(x_{k+1}) < f(x_k)$, то $h_{k+1} = h_k$ и идти на 9.
6. Если $h_k < \varepsilon$, перейти на 10.
7. Если $k=0$, то $h_{k+1} = -h_k$ и идти на 9.
8. $h_{k+1} = -h_k/2$;
9. $k = k+1$ и перейти на 3.
10. Конец. ▲

Проверка в п.7 необходима для того, чтобы не уменьшать шаг до достижения окрестности минимума. После окончания алгоритма в качестве оптимального x^* может быть взято любое значение между x_k и x_{k+1} . При сильной изменчивости функции в окрестности минимума условие п.6 можно заменить на $(h_k < \varepsilon) \ \& \ (f(x_{k+1}) - f(x_k) < \varepsilon_f)$, где ε_f – точность по функции.

8.7.2. Квадратичная аппроксимация

Для нахождения приближенного минимума исходная функция аппроксимируется функцией второго порядка

$$\bar{f}(x) = a(x - x_0)^2 + b(x - x_0) + c, \quad (8.35)$$

где x_0 – точка отсчета реального диапазона значений x , в частности $x_0=0$. Для определения коэффициентов, входящих в функцию (8.35), выбираются 3 точки и в них вычисляется значение функции f . Получается простая система 3-х уравнений с 3-мя неизвестными a , b и c :

$$\begin{aligned} f(x_1) &= a(x_1 - x_0)^2 + b(x_1 - x_0) + c, \\ f(x_2) &= a(x_2 - x_0)^2 + b(x_2 - x_0) + c, \\ f(x_3) &= a(x_3 - x_0)^2 + b(x_3 - x_0) + c. \end{aligned}$$

Стационарная точка функции (8.35) вычисляется из равенства ее производной нулю:

$$x_a = x_0 - \frac{b}{2a}. \quad (8.36)$$

Она используется в условии останова и как новая точка для аппроксимации.

Алгоритм.

1. Задать первую точку x_1 , шаг Δx и точность по координате ε и функции ε_f .

2. Определить 2-ю точку: $x_2 = x_1 + \Delta x$.
3. Вычислить $f(x_1)$ и $f(x_2)$.
4. Если $f(x_2) < f(x_1)$, принять $x_3 = x_2 + \Delta x$, иначе $x_3 = x_1 - \Delta x$.
5. Вычислить $f(x_3)$.
6. Вычислить $f_m = \min\{f(x_1), f(x_2), f(x_3)\}$ и определить точку x_m , соответствующую f_m .
7. По трем точкам и значениям функции в них найти коэффициенты в (8.35).
8. Вычислить x_a по формуле (8.36).
9. Если $(|f_m - f(x_a)| < \varepsilon_f) \ \& \ (|x_m - x_a| < \varepsilon)$, окончить поиск.
10. Если $f_m < f(x_a)$, взять x_m и две ближайшие к ней точки, иначе взять x_a и две ближайшие к ней точки. Выбранные точки пронумеровать в порядке возрастания значений, вычислить f в новой точке и перейти на 6.

8.7.3. Метод деления интервала пополам

Этот метод, как и два следующих, относится к методам сокращения интервала неопределенности. Предполагается, что точка минимума находится на заданном интервале $[a, b]$.

Рассматриваемый метод также называют трехточечным. Точки выбираются так, что делят интервал на 4 равные части (рис. 8.9):

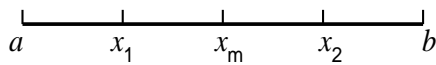
$$x_m = (a+b)/2, \quad x_1 = (a + x_m)/2, \quad x_2 = (x_m + b)/2.$$


Рис. 8.9

Функция вычисляется в этих точках и после сравнения ее значений интервал сокращается в 2 раза. С новым интервалом действия повторяются.

Алгоритм.

1. Задать точность по координате ε .
2. Вычислить x_m и $f(x_m)$.
3. Вычислить x_1 и x_2 .
4. Вычислить $f(x)$ в этих точках.
5. Если $f(x_1) < f(x_m)$, то принять $b = x_m$, $x_m = x_1$. Иначе, если $f(x_2) < f(x_m)$, положить $a = x_m$, $x_m = x_2$, иначе – $a = x_1$, $b = x_2$.
6. Если $b - a < \varepsilon$, закончить поиск, иначе перейти на 3.

Нетрудно подсчитать, что после n вычислений функции интервал неопределенности уменьшится в $2^{\frac{n-1}{2}}$ раз. Доказано, что этот метод эффективнее других прямых методов, использующих равноудаленные точки.

8.7.4. Метод золотого сечения

Золотое сечение – это определенное отношение части к целому. Отрезок АВ делится точкой С в отношении золотого сечения (рис. 8.10), если

$$\frac{AC}{AB} = \frac{CB}{AC}. \quad (8.37)$$

Положим $AB = 1$, $AC = x$, $CB = 1 - x$, тогда из (8.37) получаем уравнение

$$x^2 + x - 1 = 0,$$

из которого следует

$$x = \frac{\sqrt{5}-1}{2} \approx 0.618, \quad 1-x = \frac{3-\sqrt{5}}{2} \approx 0.382..$$

Эти отношения используются для выбора двух точек внутри интервала неопределенности. Они располагаются, как показано на рис. 8.11. Каждая из точек делит интервал $[a, b]$ в отношении золотого сечения.

В этих точках вычисляется функция. Если $f(x_1) > f(x_2)$, то отбрасывается часть интервала $[a, x_1]$, если $f(x_1) < f(x_2)$, то отсекается часть $[x_2, b]$, а при равенстве значений функции – любая из них.

Оставшаяся часть интервала равна $(\sqrt{5}-1)/2$ от величины исходного. Очевидно, что после такого сокращения интервала одна из внутренних точек остается с изменением индекса, а вторая берется на основе золотого сечения или, что одно и то же, симметрично оставшейся (рис. 8.12). Сокращение интервала продолжается до достижения заданной точности.

Алгоритм.

1. Задать точность по координате ε .
2. Вычислить $x_1 = a + ((3-\sqrt{5})/2)(b-a)$, $x_2 = a + ((\sqrt{5}-1)/2)(b-a)$.
3. Вычислить $f(x_1), f(x_2)$.
4. Если $f(x_1) > f(x_2)$, положить $a = x_1$,
 $x_1 = x_2$, $x_2 = a + ((\sqrt{5}-1)/2)(b-a)$ или
 $x_2 = a + b - x_1$, иначе – $b = x_2$, $x_2 = x_1$,
 $x_1 = a + ((3-\sqrt{5})/2)(b-a)$, или $x_1 = a + b - x_2$.
5. Если $(b-a) < \varepsilon$, закончить поиск.
6. Вычислить функцию в новой точке и перейти на 4. ▲

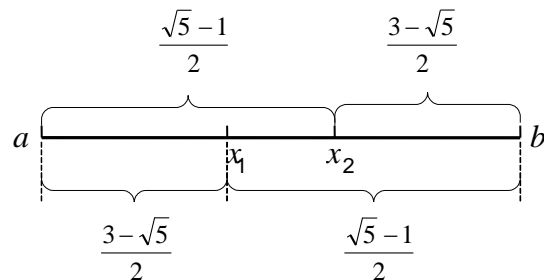


Рис. 8.11

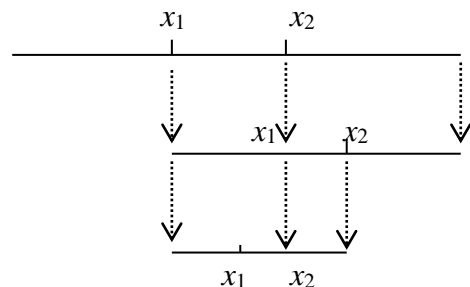


Рис. 8.12

Итерации алгоритма графически иллюстрируются на рис. 8.12.

Покажем, что сохраняемая точка (x_1 или x_2) делит сокращенный интервал также в отношении золотого сечения. Пусть на k -й итерации внутренние точки делят интервал $[a^k, b^k]$ в отношении золотого сечения. Обозначив $y^k = b^k - a^k$, $y_1^k = x_1 - a^k = b^k - x_2$, имеем

$$\frac{y_1^k}{y^k} = \frac{3 - \sqrt{5}}{2}.$$

Тогда для нового, сокращенного, интервала находим

$$y^{k+1} = \frac{\sqrt{5} - 1}{2} \cdot y^k,$$

$$y_1^{k+1} = y^k - 2y_1^k = y^k - 2 \cdot \frac{3 - \sqrt{5}}{2} \cdot y^k = y^k (1 - 3 + \sqrt{5}) = (\sqrt{5} - 2)y^k.$$

В результате получаем:

$$\frac{y_1^{k+1}}{y^{k+1}} = \frac{\sqrt{5} - 2}{\sqrt{5} - 1} \cdot 2 = 2 \cdot \frac{(\sqrt{5} - 2)(\sqrt{5} + 1)}{(\sqrt{5} - 1)(\sqrt{5} + 1)} = \frac{3 - \sqrt{5}}{2}.$$

Благодаря этому свойству, внутренние точки не сливаются при любом числе итераций.

Согласно алгоритму функция вычисляется 2 раза на начальном интервале и по одному разу на всех последующих. Поэтому после n вычислений функции интервал неопределенности составит $\left(\frac{\sqrt{5} - 1}{2}\right)^{n-1}$ от величины первоначального. При заданной точности можно найти необходимое количество вычислений функции n из условия

$$\left(\frac{\sqrt{5} - 1}{2}\right)^{n-1} (b - a) < \varepsilon.$$

8.7.5. Метод Фибоначчи

Схема метода почти полностью совпадает с методом золотого сечения. Отличие в том, что вместо золотого сечения используется отношение чисел Фибоначчи: на k -й итерации доли малого и большого отрезков интервала равны F_{n-k-1} / F_{n-k+1} и F_{n-k} / F_{n-k+1} соответственно.

Числа Фибоначчи F_v вычисляются по известным соотношениям: $F_0 = F_1 = 1$, $F_v = F_{v-1} + F_{v-2}$, $v \geq 2$.

Точки x_1 и x_2 вычисляются по формулам:

$$x_1^k = a_k + \frac{F_{n-k-1}}{F_{n-k+1}} \cdot (b_k - a_k), \quad (8.38)$$

$$x_2^k = a_k + \frac{F_{n-k}}{F_{n-k+1}} \cdot (b_k - a_k). \quad (8.39)$$

Как видно, они идентичны приведенным в предыдущем разделе. Однако если при использовании золотого сечения внутренние точки не могут

сливаться, то здесь это не так. Действительно, при $k=n-1$ из (8.38) и (8.39) имеем

$$x_1^{n-1} = a_{n-1} + \frac{F_0}{F_2} \cdot (b_{n-1} - a_{n-1}), \quad x_2^{n-1} = a_{n-1} + \frac{F_1}{F_2} \cdot (b_{n-1} - a_{n-1}).$$

Но так как $F_0/F_2 = F_1/F_2 = 1/2$, то $x_1^{n-1} = x_2^{n-1}$ и, следовательно, точки сливаются в середине интервала. Поэтому до начала итераций необходимо определить значение n , гарантирующее достижение минимума с заданной точностью ε . После 1-й итерации длина интервала составит F_{n-1}/F_n от величины исходного, после 2-й – $(F_{n-2}/F_{n-1})(F_{n-1}/F_n)$, ..., после $(n-1)$ -й –

$$\left(\frac{F_{n-1}}{F_n}\right) \cdot \left(\frac{F_{n-2}}{F_{n-1}}\right) \cdot \left(\frac{F_{n-3}}{F_{n-2}}\right) \cdot \dots \cdot \left(\frac{F_1}{F_2}\right) = \frac{F_1}{F_n} = \frac{1}{F_n}.$$

Значит, длина последнего интервала будет равна $(b_1 - a_1)/F_n$, где $[a_1, b_1]$ – исходный интервал. Для обеспечения заданной точности требуется, чтобы

$$\frac{b_1 - a_1}{F_n} \leq \varepsilon \quad \text{или} \quad F_n \geq \frac{b_1 - a_1}{\varepsilon}. \quad (8.40)$$

Таким образом, соотношение (8.40) позволяет определить номер числа Фибоначчи по исходным данным. На начальном интервале точки вычисляются по формулам (8.38) и (8.39) при $k=1$. На последующих итерациях числа Фибоначчи не требуются, так как одна точка переносится из предшествующей итерации, а вторая берется симметрично ей, то есть лучше использовать вторые формулы из п.4 алгоритма золотого сечения.

После слияния внутренних точек остается неопределенность с положением минимума. Для ее устранения вторая точка берется слева или справа от центра на расстоянии $\varepsilon_1 \cong (0,01 \div 0,05)\varepsilon$. Для случая сдвига второй точки влево (рис. 8.13) при $f(x_1) < f(x_1 - \varepsilon_1)$ минимум лежит в интервале (2), в противном случае – в интервале (1).

Метод Фибоначчи является самым эффективными из всех прямых методов. Очень близок к нему метод золотого сечения: при $n > 9$ они практически совпадают по эффективности и чем больше n , тем ближе эти методы. А в пределе отношение, используемое в методе Фибоначчи на 1-й итерации, становится равным золотому сечению:

$$\lim_{n \rightarrow \infty} \left(\frac{F_{n-2}}{F_n} \right) = \frac{3 - \sqrt{5}}{2}.$$

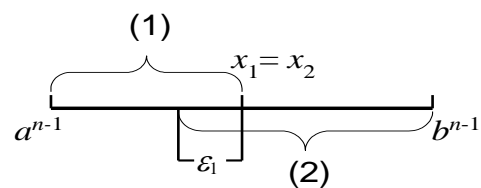


Рис. 8.13

8.7.6. Метод первого порядка

Он применим к дифференцируемым функциям в случаях, когда первая производная может вычисляться с высокой точностью.

Рассматриваемый метод средней точки алгоритмически является самым простым из методов одномерной оптимизации на заданном интервале. Он заключается в определении средней точки интервала $c=(a+b)/2$ и вычислении в ней производной.

Если $f'(c) < 0$, полагают $a = c$, при $f'(c) > 0$, принимают $b = c$. Условие окончания поиска учитывает точность по координате ε и по производной функции ε_1 :

$$((b-a) < \varepsilon) \ \& \ (|f'(c)| < \varepsilon_1).$$

Эффективность метода зависит от трудоемкости вычисления производной. Задаваемая точность ε_1 не должна быть выше точности вычисления производной. С ростом трудоемкости и снижением точности вычислений производной следует отдавать предпочтение прямым методам.

8.7.7. Методы второго порядка

Методы применимы к дважды дифференцируемым функциям. При этом предъявляются высокие требования к точности вычисления производных. Наиболее удачными считаются задачи, в которых известно аналитическое выражение первой производной, а еще лучше – и второй.

Здесь приводится только один из наиболее часто используемых методов – метод Ньютона – Рафсона.

Он основан на линейной аппроксимации первой производной минимизируемой функции f в окрестности текущей точки x_k :

$$f' \approx \bar{f}' = f'(x_k) + f''(x_k)(x - x_k).$$

В стационарной точке аппроксимации, которая принимается за очередное приближение x_{k+1} , производная равна нулю:

$$\bar{f}'(x_{k+1}) = 0.$$

Отсюда следует рекуррентная формула для построения последовательности приближений к искомому минимуму:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}. \quad (8.41)$$

Очевидно, что применение (8.41) возможно только при условии, что $f''(x_k) \neq 0$ для каждого k . Поиск завершается по условию достижения точности, заданной величиной первой производной $|f'(x_k)| < \varepsilon_1$ или расстоянием между двумя точками $|x_{k+1} - x_k| < \varepsilon$. Возможно одновременное использование этих условий.

В общем случае процедура (8.41) не гарантирует сходимость к стационарной точке. Если начальная точка достаточно близка к стационарной, то метод сходится. При сходимости обеспечивается высокая

скорость приближения к минимуму. На рис. 8.14 приведен случай сходимости метода. Очередному приближению соответствует точка пересечения оси x аппроксимирующей прямой. Как видно, последовательность точек x_0, x_1, x_2, \dots приближается к минимуму x^* .

Для некоторых функций результат поиска зависит от выбора начальной точки. Так, например, при начальной точке, взятой правее максимуму производной, так как показано на рис. 8.15, метод расходится.

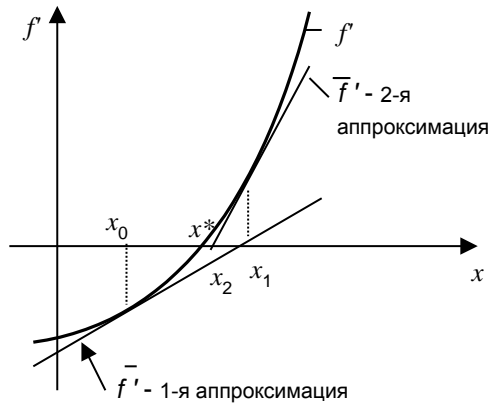


Рис. 8.14

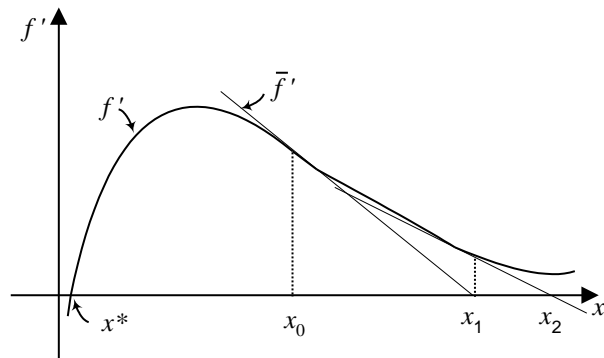


Рис. 8.15

8.8. Многомерный поиск безусловного минимума

Здесь рассматриваются методы прямого поиска (разд. 8.8.1, 8.8.2 и 8.8.3), градиентные методы (разд. 8.8.4), метод второго порядка, алгоритмы случайного поиска и приводятся краткие сведения о других методах.

8.8.1. Метод Гаусса-Зейделя (покоординатного спуска)

Процедура поиска сводится к решению последовательности задач одномерной минимизации по каждой переменной. Пусть выбрана начальная точка

$$\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}).$$

Зафиксируем все переменные, кроме первой, на начальных значениях и решаем задачу

$$f(x_1, x_2^{(0)}, \dots, x_n^{(0)}) \Rightarrow \min_{x_1}$$

одним из одномерных методов. Фиксируем x_1 на полученном в решении значении x_1' и делаем свободной переменную x_2 . Приходим к очередной одномерной задаче

$$f(x_1', x_2, x_3^{(0)}, \dots, x_n^{(0)}) \Rightarrow \min_{x_2} \rightarrow x_2'.$$

Аналогично строятся и решаются последующие одномерные задачи, последняя из которых имеет вид:

$$f(x'_1, x'_2, \dots, x'_{n-1}, x'_n) \Rightarrow \min_{x_n} \rightarrow x'_n.$$

Эти n задач составляют один цикл. Его результатом является точка X^1 . Она принимается за начальную точку для следующего аналогичного цикла. Поиск заканчивается, когда расстояние между двумя последовательными точками становится меньше заданной величины:

$$\|x^{(k+1)} - x^{(k)}\| < \varepsilon.$$

Работу метода иллюстрирует рис. 8.16, на котором показана траектория поиска минимума функции $f=(2-x_1)^4+2(x_1-2x_2)^2$.

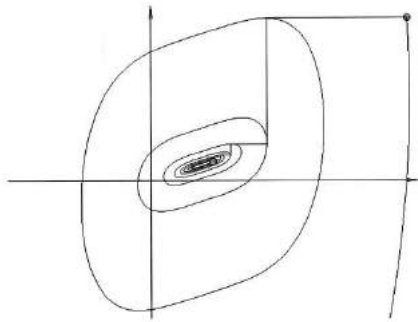


Рис. 8.16

Метод отличается алгоритмической простотой. Однако ему присущ ряд недостатков. Его эффективность существенно зависит от направления осей координат относительно линий уровня. Это хорошо видно на примере квадратичной функции: при совпадении координат с осями эллипсов минимум достигается за один цикл из любой начальной точки (рис. 8.17), а при их повороте число циклов значительно возрастает (рис. 8.18). Из этого

примера также следует, что метод неэффективен в условиях оврага. Если функция не дифференцируема в отдельных точках, поиск может остановиться, не достигнув окрестности минимума. Рис. 8.19 демонстрирует такой случай: точка останова A далека от искомого минимума.

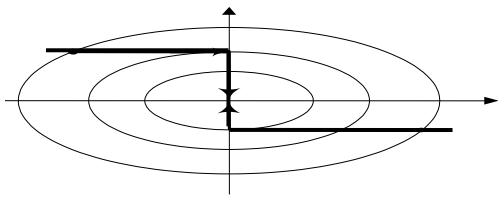


Рис. 8.17

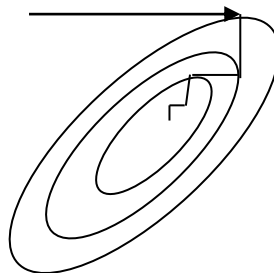


Рис. 8.18

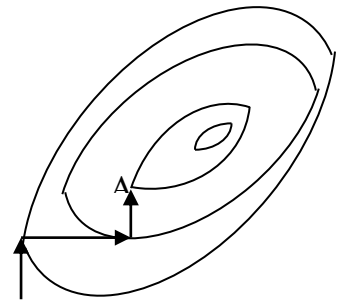


Рис. 8.19

Из анализа траекторий поиска в приведенных примерах можно заключить, что эффективность поиска повысится, если к описанным одноступенчатым циклам добавить движение в направлении, проходящем через точки $X^{(k)}$ и $X^{(k+1)}$. Это движение называют ускоряющим шагом. Он используется в методе, рассматриваемом в следующем разделе.

8.8.2. Метод Хука-Дживса (метод конфигураций)

В этом методе каждая итерация состоит из двух фаз:

- 1) исследующий поиск;

2) движение по образцу (ускоряющий шаг).

Исследующий поиск аналогичен одному циклу покоординатного спуска. Конечную точку цикла называют базовой. Две последовательные базовые точки определяют направление поиска на 2-й фазе. Точка, получаемая в результате ускоряющего шага, называется временной. Начальная точка одновременно является базовой и временной.

Сначала приведем вариант метода, предложенный Хуком и Дживсом и описываемый в литературе под названием "метод конфигураций". В нем используются только

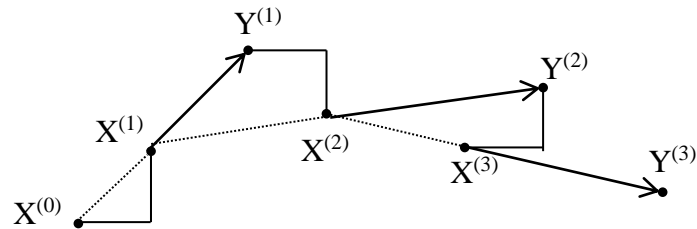


Рис. 8.20

дискретные шаги. Процедура поиска показана на рис. 8.20. Из начальной точки $X^{(0)}$ делается шаг Δ по x_1 в одну сторону и, если успешно, то значение фиксируется, иначе направление шага изменяется на противоположное. Также делается по остальным координатам. Итогом первого исследующего поиска является точка $X^{(1)}$. В направлении $X^{(1)} - X^{(0)}$ осуществляется движение по образцу, дающее временную точку $Y^{(1)} = X^{(1)} + \alpha(X^{(1)} - X^{(0)})$, где α – коэффициент ускорения (рекомендуется брать от 1 до 2). Из полученной временной точки снова проводится исследующий поиск, приводящий в очередную базовую точку $X^{(2)}$. Ускоряющий шаг в направлении $X^{(2)} - X^{(1)}$ дает временную точку $Y^{(2)}$. Фазы поиска повторяются без изменения величины шага, если $f(Y^{(k)}) < f(X^{(k)})$. В противном случае, если $\Delta < \varepsilon$, поиск заканчивается и $X^{(k)}$ принимается за решение, а иначе следует положить $Y^{(k)} = X^{(k)}$, уменьшить шаг в два раза и перейти к исследующему поиску из точки $Y^{(k)}$. Иначе говоря, после каждого ускоряющего шага проверяется его успешность, и в зависимости от результата выбираются последующие действия..

Модификация метода Хука-Дживса заключается в замене дискретных шагов одномерной минимизацией. В этом варианте исследующий поиск полностью совпадает с одним циклом покоординатного спуска, то есть по каждой координате выполняются не дискретные шаги, а ищется минимум. При движении по образцу также ищется минимум функции только по одной неизвестной – коэффициенту ускорения α :

$$f(\mathbf{X}^{(k)} + \alpha(\mathbf{X}^{(k)} - \mathbf{X}^{(k-1)})) \Rightarrow \min_{\alpha}.$$

По оптимальному значению α^* определяется временная точка

$$\mathbf{Y}^{(k)} = \mathbf{X}^{(k)} + \alpha^*(\mathbf{X}^{(k)} - \mathbf{X}^{(k-1)})$$

Поиск завершается, когда расстояние между двумя смежными базовыми точками становится меньше заданной величины:

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| < \varepsilon.$$

К этому условию можно добавить и требование по точности функции:

$$|f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)})| < \varepsilon_f.$$

На рис. 8.21 и 8.22 приведены примеры применения метода с одномерной минимизацией. Как видно, такой вариант метода обеспечивает быстрое приближение к области искомого решения. Другим важным достоинством метода является его работоспособность в условиях оврага. Траектория поиска хорошо приспосабливается к изгибам дна оврага (рис. 8.22).

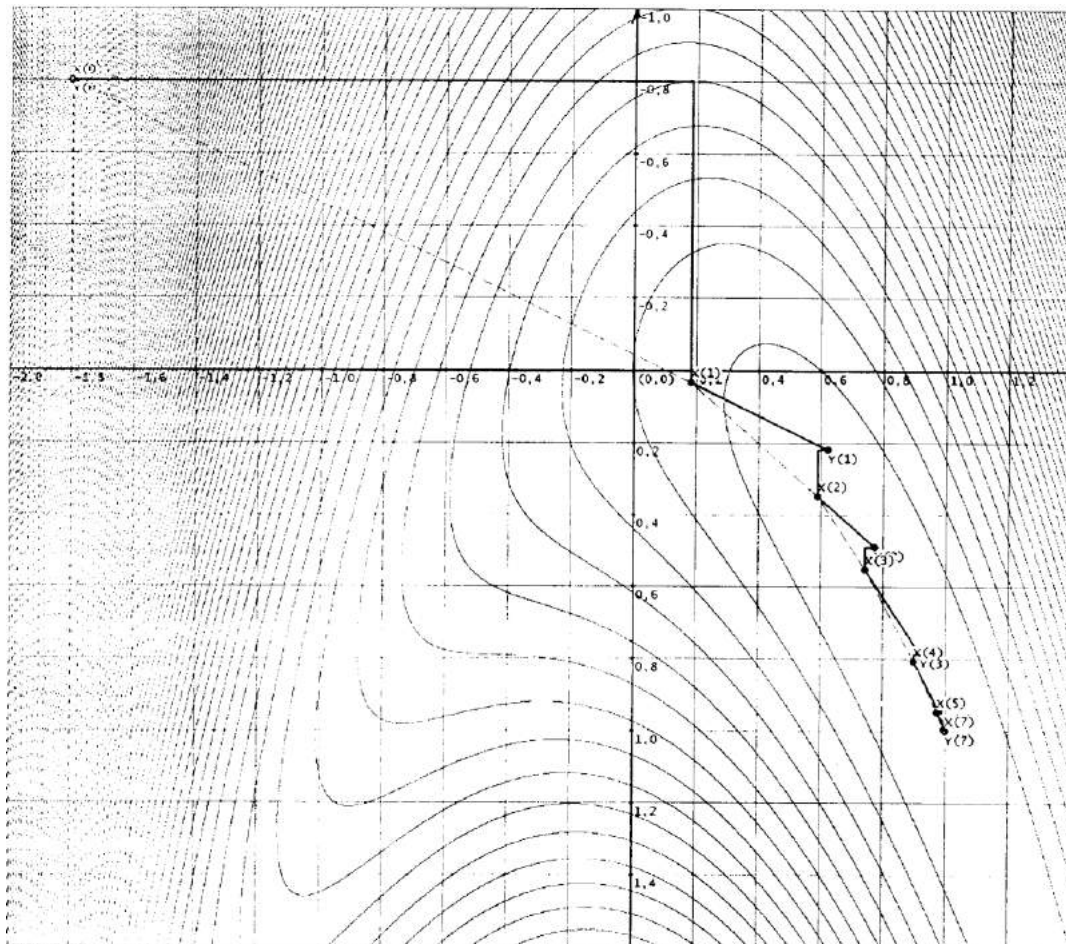


Рис. 8.21

8.8.3. Симплексный метод

В этом методе движение к минимуму осуществляется с использованием симплекса – выпуклого многогранника с числом вершин на единицу больше размерности пространства. Примером симплекса на плоскости является любой треугольник, в трехмерном пространстве – пирамида с треугольником в основании.

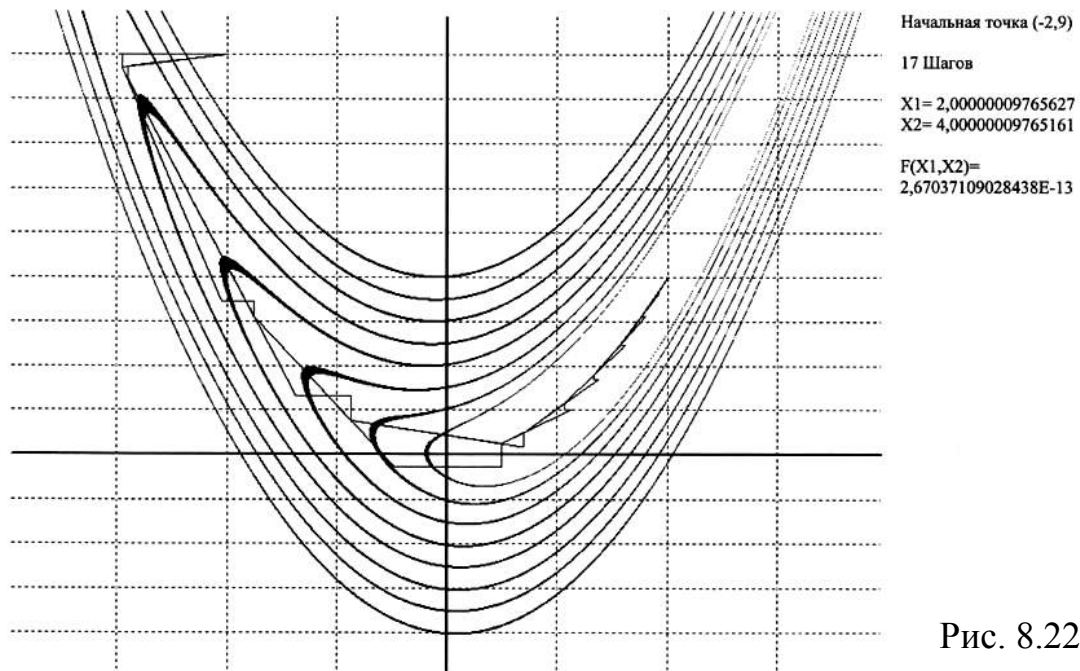


Рис. 8.22

Симплекс обладает замечательным свойством: если взять только одну точку вне симплекса в качестве новой вершины, то, соединив ее с вершинами прилежащей грани, получим новый симплекс. На рис. 8.23 слева из исходного симплекса с вершинами 1,2,3 построено 2 других симплекса, а справа новый симплекс построен на грани 2,3,4 исходного и новой вершины 5.

Очевидно, что если взять одну из вершин симплекса, то все остальные вершины будут на противолежащей грани симплекса. Эти свойства справедливы для любой размерности пространства.

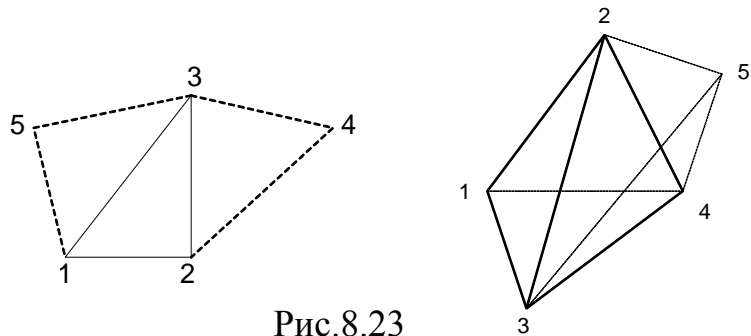


Рис.8.23

Зная значения функции в вершинах симплекса, легко определить направление, в котором функция может улучшиться. В этом направлении строится новый симплекс: определяется самая худшая вершина (по значению функции) и она отражается через центр противолежащей грани, как показано на рис. 8.24. В построенном симплексе значение функции неизвестно только в вершине 4. Таким образом, после построения нового симплекса функция вычисляется всего один раз *при любой размерности пространства*. Лишь в начальном

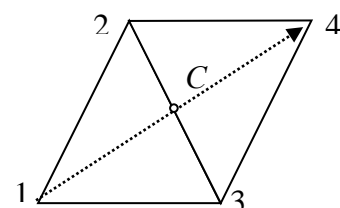


Рис. 8.24

симплексе необходимо вычислять функцию во всех вершинах.

Последовательное отражение наихудших вершин перемещает симплекс в направлении уменьшения значения функции, что в конечном итоге приводит к отысканию минимума с заданной точностью. Характер

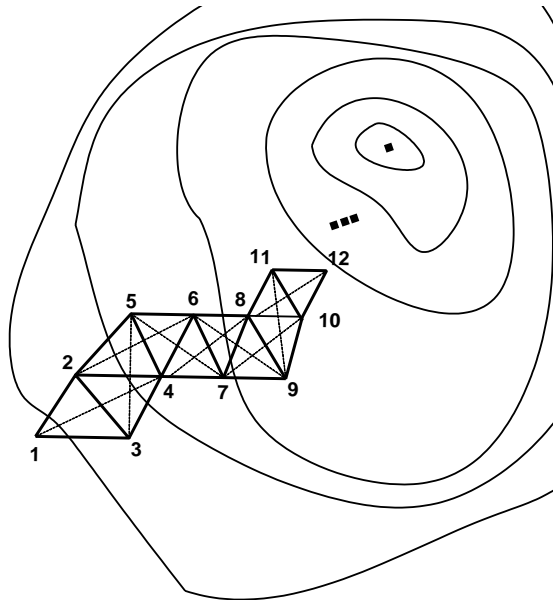


Рис. 8.25

движения симплекса представлен на рис. 8.25. Отражение вершин показано пунктирными линиями. Сначала отражается вершина 1, затем в полученном симплексе отражается вершина 3, симплекс 2,4,5 заменяется на 4,5,6 и так далее. Очевидно, что в процессе движения, особенно вблизи экстремума, новая вершина может оказаться не лучше отраженной. Для предотвращения заикливания либо отражается другая вершина, либо уменьшается размер нового симплекса.

Рассмотрим вычислительную сторону метода. В качестве начального симплекса обычно берется регулярный симплекс. Регулярность означает равенство длин всех ребер. Он строится следующим образом. Начальная точка $X^{(0)}$ принимается за базовую вершину, относительно которой располагаются остальные вершины. Их координаты вычисляются по формуле

$$x_j^{(i)} = \begin{cases} x_j^{(0)} + \delta_1, & \text{если } j \neq i, \\ x_j^{(0)} + \delta_2, & \text{если } j = i; \quad i, j = \overline{1, n}, \end{cases}$$

где i – номер вершины, j – координата вершины (индекс переменной), n – число переменных (размерность пространства), а приращения δ_1 и δ_2 зависят от n и выбранного масштабного множителя μ :

$$\delta_1 = \left(\frac{\sqrt{n+1} + n - 1}{n\sqrt{2}} \right) \mu,$$

$$\delta_2 = \left(\frac{\sqrt{n+1} - 1}{n\sqrt{2}} \right) \mu.$$

При $\mu=1$ все ребра имеют единичную длину.

Вычисление координат новой вершины, получающейся в результате отражения одной из вершин последнего симплекса, производится следующим образом. Пусть отражается вершина k (например, вершина 1 на рис. 8.24). Тогда сначала определяется центр противолежащей грани C

$$\mathbf{X}^c = \frac{1}{n} \sum_{\substack{i=1 \\ i \neq k}}^{n+1} \mathbf{x}^{(i)},$$

а затем координаты новой вершины s :

$$\mathbf{x}^{(s)} = \mathbf{x}^{(k)} + \lambda \cdot (\mathbf{x}^C - \mathbf{x}^{(k)})$$

или

$$\mathbf{x}^{(s)} = \mathbf{x}^C + \theta (\mathbf{x}^C - \mathbf{x}^{(k)}), \quad (8.42)$$

где $\lambda > 0$ и θ – параметры отражения.

Предложен целый ряд методов, основанных на поиске по симплексам, отличающихся способом построения очередного симплекса и значениями параметров отражения. Лучшим из них считается метод *Нелдера – Мида*. В нем заложено гибкое управление симплексом, при котором он может как уменьшаться, так и увеличиваться, приспособливаясь к поверхности функции. Оно использует 3 варианта отражения с соответствующими параметрами:

$$\theta = \begin{cases} \alpha - \text{нормальное отражение,} \\ \beta - \text{сжатие,} \\ \gamma - \text{растяжение.} \end{cases}$$

Значения коэффициентов отражения были подобраны экспериментально: $\alpha = 1$; $\beta = 0,5$; $\gamma = 2$. Предложены и другие наборы значений, однако они дают лучшие результаты только в частных случаях.

В алгоритме Нелдера-Мида используются три вершины: A – с наибольшим значением целевой функции (худшая), B – со следующим значением и L – с наименьшим значением (лучшая). Всегда отражается вершина A .

Сначала производится нормальное отражение. Получаемая в результате точка D считается временной (рис. 8.26). Далее идут проверки:

- если $f^L < f^D < f^B$, нормальное отражение приемлемо и временная точка фиксируется;
- если $f^B \leq f^D < f^A$, то выполняется сжатие с $\theta = \beta$ и фиксируется точка D_β ;
- если $f^D \geq f^A$, симплекс сжимается с $\theta = -\beta$ и фиксируется точка $D_{-\beta}$;
- если $f^D < f^L$, то производится растяжение ($\theta = \gamma$), дающее точку D_γ , которая фиксируется, если она не хуже временной; иначе она не учитывается и фиксируется временная точка.

В условии останова поиска может использоваться показатель размера симплекса, например, максимальная длина ребра, и/или разность значений целевой функции $f^A - f^L$.

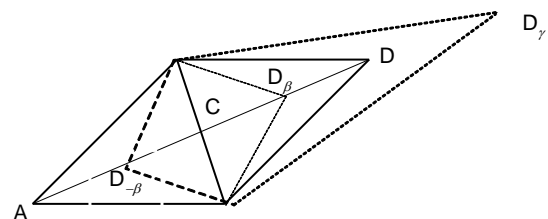


Рис. 8.26

Метод характеризуется как эффективный. Он мало чувствителен к помехам или ошибкам при определении значений целевой функции, так как определяющими являются отношения значений, а не абсолютные величины. Благодаря изменению размеров симплекса он может работать в условиях оврага.

Интересен вариант метода, в котором на всех итерациях симплекс остается регулярным. В нем используется коэффициент уменьшения длины ребра $\beta \geq 0,85$. Симплекс сжимается, стягиваясь к лучшей вершине, как показано на рис. 8.27, и после этого происходит отражение. Такое значение β обусловлено тем, что на вершины уменьшенного симплекса

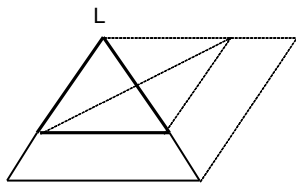


Рис. 8.27

переносятся значения функции соответствующих вершин до сжатия. Тем самым исключаются дополнительные вычисления функции. Возможно также увеличение симплекса за счет удлинения всех ребер с коэффициентом $1/\beta$ при сохранении положения наихудшей вершины.

8.8.4. Градиентные методы

Если функция дифференцируема, вычисление производных дает информацию о поведении функции в исследуемой точке и, следовательно, позволяет находить лучшее направление поиска..

Скорость изменения функции $f(X)$ в произвольном направлении l определяется производной

$$\frac{df}{dl} = \sum_{i=1}^n \frac{\partial f}{\partial x_i} \frac{dx_i}{dl}. \quad (8.43)$$

Здесь частные производные $\frac{dx_i}{dl}$ представляют собой направляющие косинусы. Геометрическая иллюстрация для случая функции двух переменных приведена на рис. 8.28. Очевидно, что

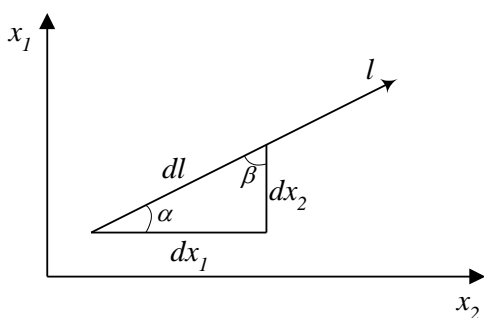


Рис. 8.28

$$\left. \begin{aligned} \frac{dx_1}{dl} &= \cos(\hat{x}_1, l) = \cos \alpha, \\ \frac{dx_2}{dl} &= \cos(\hat{x}_2, l) = \cos \beta. \end{aligned} \right\}$$

Поверхность уровня целевой функции, описываемая уравнением $f(X) = \text{const}$, имеет размерность $n - 1$.

Зададим координаты следующим образом. Проведем через рассматриваемую точку $n - 1$ взаимно ортогональные касательные к

поверхности уровня, приняв их за оси координат. В качестве последней оси возьмем нормаль к поверхности уровня (рис. 8.29).

В такой системе координат производные функции по всем x_j равны нулю. Поэтому из (8.43) имеем

$$\frac{df}{dl} = \frac{\partial f}{\partial N} \cdot \cos(N, l).$$

Отсюда следует, что максимальная скорость увеличения функции будет в направлении l , совпадающем с нормалью. Вектор, имеющий направление нормали к поверхности функции в данной точке и длину $\partial f / \partial N$, называется *градиентом*.

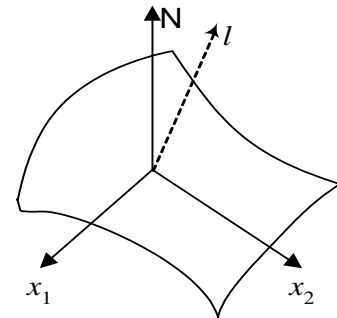


Рис. 8.29

Обозначается градиент как $\text{grad } f(X)$ или $\nabla f(X)$. Он полностью определяется своими проекциями – производными функции по переменным:

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

В задачах на минимум используется противоположное направление – *антиградиент*.

Значения производных могут быть найдены по приближенным формулам:

$$\frac{\partial f}{\partial x_i} \approx \frac{\Delta f}{\Delta x_i} = \frac{f(x_1, \dots, x_{i-1}, x_i + \Delta x_i, x_{i+1}, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{\Delta x_i},$$

$$\frac{\partial f}{\partial x_i} \approx \frac{\Delta f}{\Delta x_i} = \frac{f(x_1, \dots, x_{i-1}, x_i + \Delta x_i / 2, x_{i+1}, \dots, x_n) - f(x_1, \dots, x_i - \Delta x_i / 2, \dots, x_n)}{\Delta x_i}.$$

Более точный результат дает вторая формула, но она требует больше вычислений. Чем точнее необходимо вычислить производную, тем меньше должно быть Δx . Однако недопустимо, чтобы разность значений функции была соизмерима с погрешностью вычисления.

Если переменные имеют разные единицы измерения, можно перейти к относительным переменным y_i , используя минимально и максимально возможные значения переменных x_i :

$$y_i = \frac{x_i - x_i^{\min}}{x_i^{\max} - x_i^{\min}}.$$

Очевидно, что значения y_i лежат в диапазоне $[0, 1]$.

Знание градиента (антиградиента) позволяет осуществлять перемещение из текущей точки в направлении наибольшего улучшения функции. Для линейных функций оно постоянно и поэтому его требуется определять всего один раз. В нелинейных функциях значение градиента

зависит от вектора \mathbf{X} , то есть его необходимо пересчитывать при любом изменении \mathbf{X} .

В градиентном методе поиск минимума производится согласно рекуррентным формулам:

$$\begin{aligned}\mathbf{x}^{k+1} &= \mathbf{x}^k - h^k \cdot \nabla f(\mathbf{x}^k), \\ \mathbf{x}^{k+1} &= \mathbf{x}^k - h^k \cdot \frac{\nabla f(\mathbf{x}^k)}{\|\nabla f(\mathbf{x}^k)\|},\end{aligned}\quad (8.44)$$

где h^k - шаг. В первой формуле величина изменения переменных $\Delta \mathbf{X}$ зависит как от шага, так и от величины градиента. Более удобно, когда расстояние между последовательными точками зависит только от величины шага. Поэтому предпочтительнее вторая формула. В ней используется нормированный градиент, длина которого всегда равна единице (за счет деления на длину). Поэтому он задает лишь направление, но не влияет на величину $\Delta \mathbf{X}$.

Формула (8.44) в записи по отдельным переменным принимает вид:

$$x_i^{k+1} = x_i^k - h^k \cdot \frac{\partial f / \partial x_i}{\sqrt{\sum_{j=1}^n (\partial f / \partial x_j)^2}}. \quad (8.45)$$

Алгоритм градиентного метода.

- 1) Задать начальную точку, начальное значение шага и точность по величине градиента ε , вычислить $f(\mathbf{X}^0)$ и положить $k = 0$.
- 2) В текущей точке \mathbf{X}^k вычислить градиент (производные $\partial f / \partial x_i$) и его длину.
- 3) Проверить: если $\|\nabla f(\mathbf{x}^k)\| < \varepsilon$, закончить поиск.
- 4) Определить новую точку \mathbf{X}^{k+1} согласно (8.45) и вычислить в ней значение функции.
- 5) Проверить: если $f(\mathbf{X}^{k+1}) \leq f(\mathbf{X}^k)$, увеличить k на единицу и перейти на шаг 2; если иначе, уменьшить h^k в два раза и перейти на шаг 4. ▲

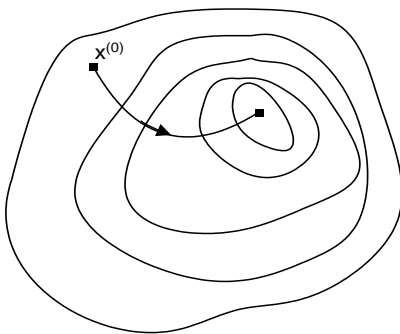


Рис. 8.30

При гладкой производной траектория поиска в идеале (при непрерывном вычислении градиента) тоже будет гладкой и ортогональной линиям уровня. При конечной величине шага траектория становится кусочно-линейной. Качественный характер поиска показан на рис. 8.30, а на рис. 8.31 приведена реализация алгоритма с двух начальных точек применительно к функции $f(\mathbf{X}) = (x_1 - x_2)^2 + (x_2 - 2)^4$.

Обе траектории приводят практически в одну точку.

Самой трудоемкой частью алгоритма является вычисление градиента, а он вычисляется после каждого успешного шага. В методе *наискорейшего спуска*, который представляет собой модификацию градиентного метода, градиент определяется реже, особенно в начальный период поиска.

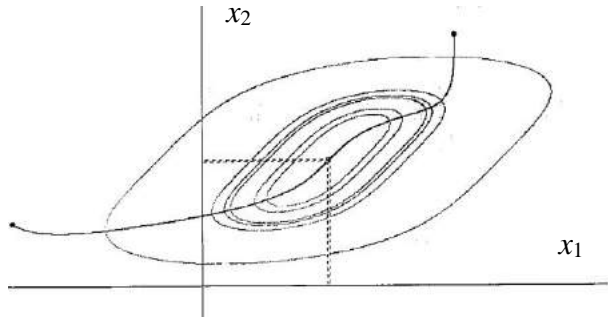


Рис.8.31

Модификация заключается в том, что после вычисления градиента вместо одного дискретного шага ищется минимум на направлении антиградиента, то есть проводится одномерный поиск по h :

$$f\left(\mathbf{x}^k - h \cdot \frac{\nabla f}{\|\nabla f\|}\right) \Rightarrow \min_h. \quad (8.46)$$

В результате решения задачи (8.46) определяется оптимальный шаг h^* , по которому находится следующая точка:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - h^* \cdot \frac{\nabla f}{\|\nabla f\|}. \quad (8.47)$$

Очевидно, что при таком определении новой точки, значение функции в ней будет лучше, чем в \mathbf{x}^k .

Алгоритм наискорейшего спуска.

- 1) Задать начальную точку и точность по величине градиента ε , положить $k = 0$.
- 2) В текущей точке \mathbf{x}^k вычислить градиент (производные $\partial f / \partial x_i$) и его длину.
- 3) Проверить: если $\|\nabla f(\mathbf{x}^k)\| < \varepsilon$, закончить поиск.
- 4) Провести одномерный поиск (8.46).
- 5) Определить новую точку \mathbf{x}^{k+1} согласно (8.47), увеличить k на единицу и перейти на шаг 2. ▲

На рис. 8.32 показаны две траектории движения к минимуму функции $f(\mathbf{X}) = (x_1 - x_2)^2 + (x_2 - 2)^4$, полученные алгоритмом. Минимум на направлении антиградиента достигается в точке касания с линией уровня, а градиент в этой точке ортогонален ей. Поэтому каждое

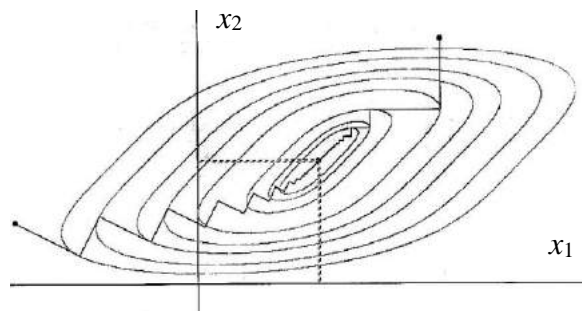


Рис. 8.32

последующее направление ортогонально непосредственно предшествующему. Из рис. 8.32 видно, что с приближением к экстремуму частота вычисления градиента увеличивается, и вблизи минимума метод наискорейшего спуска вырождается в градиентный (см. рис. 8.31 и 8.32).

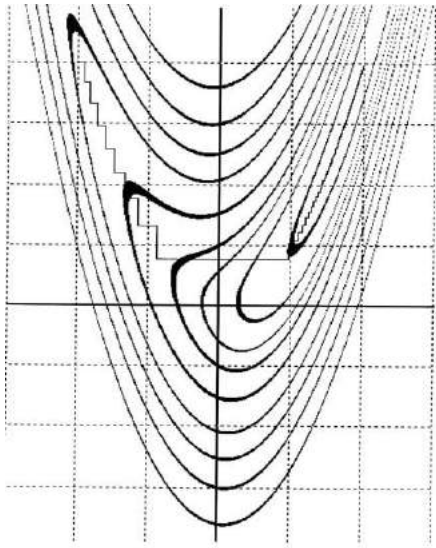


Рис. 8.33

Градиентные методы плохо работают в условиях оврага: при попадании на дно оврага резко падает скорость движения и возможна остановка поиска до достижения окрестности минимума (рис. 8.33, $f=(2-x_1)^2+3(x_1^2-x_2)^2$).

8.8.5. Метод Ньютона

Это классический метод второго порядка. Если направление градиента находится из линейного представления функции в окрестности текущей точки, то в методе Ньютона используется квадратичная аппроксимация функции цели.

Квадратичная аппроксимация дифференцируемой функции f в точке X^k записывается в виде

$$f(\mathbf{x}) \approx f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k) + 1/2 \cdot (\mathbf{x} - \mathbf{x}^k) \cdot \mathbf{H}(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k), \quad (8.48)$$

где \mathbf{H} – матрица вторых производных функции f (матрица Гессе). В стационарной точке градиент функции равен нулю, что применительно к (8.48) дает

$$\nabla f(\mathbf{x}^k) + \mathbf{H}(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k) = 0. \quad (8.49)$$

Полагая матрицу \mathbf{H} неособенной (существует обратная к ней матрица), из (8.49) получаем рекуррентный способ генерирования последовательности точек

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{H}^{-1}(\mathbf{x}^k) \cdot \nabla f(\mathbf{x}^k). \quad (8.50)$$

Исходя из вывода формулы (8.50) ясно, что для квадратичной функции цели X^{k+1} является точкой минимума. Следовательно, минимум такой функции из любой начальной точки достигается за один шаг (8.50). Для нелинейной унимодальной функции, отличной от квадратичной, точки последовательности (8.50) асимптотически приближаются к минимуму. Условие окончания поиска такое же, как и в градиентных методах: $\|\nabla f(\mathbf{x}^k)\| < \varepsilon$. (В случае линейной аппроксимации матрица \mathbf{H} становится единичной и поиск вырождается в градиентный).

Необходимым условием сходимости метода является существование обратной матрицы во всех генерируемых точках. Доказательство сходимости метода получено при условии, что начальная точка достаточно близка к точке минимума. При этом метод имеет высокую скорость

сходимости. Если поиск начинается с удаленной точки и функция существенно отличается от квадратичной, метод может не сходиться или давать слабую сходимость. Причина такого поведения в том, что значение функции в точке X^{k+1} не обязательно меньше, чем в точке X^k . Так на рис. 8.34 траектория поиска имеет зигзагообразный характер из-за того, что после 2-й итерации значение функции ухудшилось.

Для устранения отмеченных недостатков предложены модификации метода.

Чтобы обеспечить регуляризацию матрицы H (невыврожденность), она деформируется добавлением к ней единичной матрицы с неотрицательным скалярным множителем ε : $H' = H + \varepsilon \cdot E$. Значение ε зависит от X и подбирается так, чтобы все собственные значения матрицы H' были положительными. Тогда эта матрица положительно определена и существует обратная матрица H'^{-1} , которая также положительно определена.

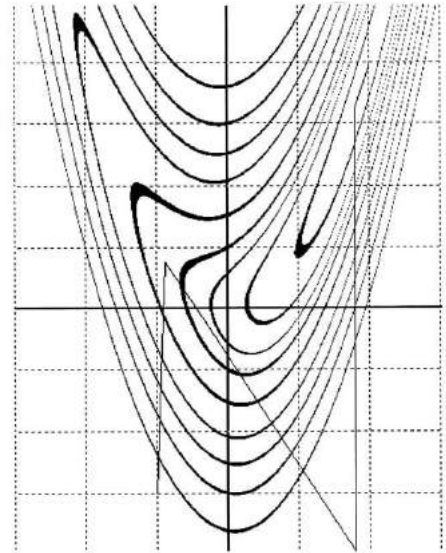


Рис. 8.34

Возможное ухудшение функции в очередной точке исключается введением в формулу (8.50) изменяемого параметра h . Модифицированная формула принимает вид

$$\mathbf{x}^{k+1} = \mathbf{x}^k - h^k \cdot \mathbf{H}^{-1}(\mathbf{x}^k) \cdot \nabla f(\mathbf{x}^k). \quad (8.51)$$

При этом возможен вариант с дискретным шагом h , как в градиентном методе, либо с определением оптимального h^* с помощью одномерной минимизации (наискорейший спуск):

$$f(\mathbf{x}^k - h \cdot \mathbf{H}^{-1} \cdot \nabla f) \Rightarrow \min_h.$$

Пример поиска минимума функции $f=(1-x_1)^2+5(x_1^2-x_2)^2$ методом Ньютона с регулируемым шагом приведен на рис. 8.35. Если сравнить траектории поиска на рис. 8.34 и 8.35, то легко заметить преимущество модифицированного варианта метода.

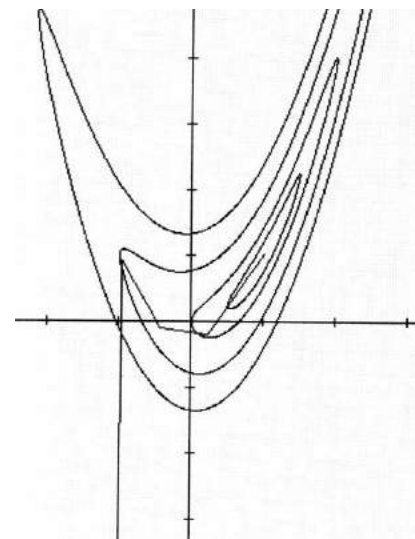


Рис. 8.35

8.8.6. Методы сопряженных направлений

Как и метод Ньютона, методы сопряженных направлений основаны на свойствах квадратичных функций. В связи с этим говорят о сопряженных направлениях относительно квадратичной функции.

Пусть дана матрица $H_{n \times n}$. Направления $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k$ ($k \leq n$) называются сопряженными или Н-сопряженными, если они линейно независимы и

$$\mathbf{d}_i^T \cdot \mathbf{H} \cdot \mathbf{d}_j = 0 \quad \text{при } i \neq j. \quad (8.52)$$

Эти векторы определяют сопряженные направления. Для квадратичной функции двух переменных сопряженные направления получаются следующим образом. Возьмем произвольное направление \mathbf{d}_1 и на нем найдем минимум, двигаясь из точки X^1 . Повторим поиск минимума на \mathbf{d}_1 из точки $X^2 \neq X^1$ (рис. 8.36).

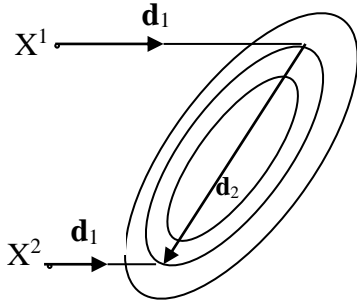


Рис. 8.36

Направление \mathbf{d}_2 , определяемое прямой, проходящей через найденные минимумы, является сопряженным с направлением \mathbf{d}_1 . При этом направление \mathbf{d}_2 проходит через искомый минимум функции f . Следовательно, при любой начальной точке минимум квадратичной функции двух переменных достигается за два одномерных поиска вдоль сопряженных направлений.

Пример 8.6. Используя сопряженные направления, найти минимум функции $f = -6x_2 + x_1^2 + x_2^2 - x_1 x_2$ (точка минимума $X^*=(2,4)$).

Запишем матрицу Гессе $\mathbf{H} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$.

За первое направление возьмем $\mathbf{d}_1^T = (1,0)$ и определим $\mathbf{d}_2^T = (a,b)$. Компоненты \mathbf{d}_2 найдем из условия (8.52):

$$(1,0) \cdot \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = 0 \Rightarrow 2a - b = 0.$$

Положив $a = 1$, получаем $b = 2$ и $\mathbf{d}_2^T = (1,2)$. Возьмем начальную точку $X^0=(-1;1)$. Найдем минимум на направлении \mathbf{d}_1 . Для этого подставим в функцию $X = X^0 + h\mathbf{d}_1$, то есть $x_1 = x_1^0 + h = -1 + h$, $x_2 = x_2^0 = 1$. Тогда $f = h^2 - 3h - 3$ и минимум по h будет при $h^*=1,5$.

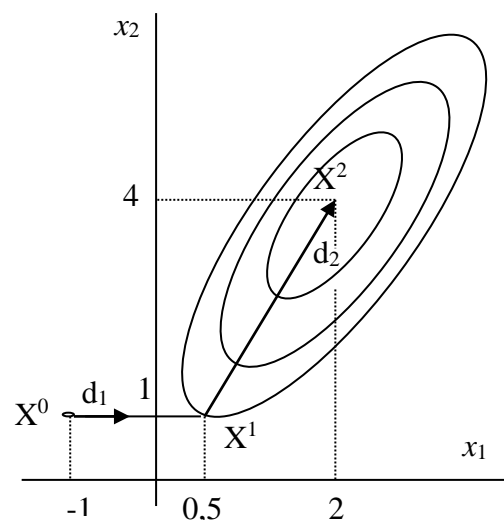


Рис. 8.37

Следовательно, минимум на \mathbf{d}_1 достигается в точке $X^1=(0,5;1)$. Приняв ее за начальную для поиска вдоль \mathbf{d}_2 и подставляя в функцию $x_1 = 0,5 + h$, $x_2 = 1 + 2h$, получаем $f = 3h^2 - 9h - 5,25$. Находим $h^*=1,5$ и соответствующую новую точку $X^2=(2;4)$. Как видим, второй одномерный поиск привел в точку искомого минимума f (рис. 8.37). ▲

Для квадратичной функции n переменных сопряженные направления позволяют найти минимум не более чем за n одномерных поисков. В случае нелинейной функции, отличной от квадратичной, конечное число итераций дает только приближенное решение.

Методы, основанные на концепции сопряженных направлений, различаются способами построения таких направлений. Ряд из них относятся к прямым методам, например, метод *Пауэлла* и его модификация – метод *Зангвилла*. Другие используют первые производные, например, метод сопряженных градиентов (метод *Флетчера-Ривса*). Одним из самых эффективных является метод *Дэвидона-Флетчера-Пауэлла*. В нем генерируются сопряженные направления с использованием градиента и матрицы D , аппроксимирующей обратную матрицу H^{-1} . Поэтому его относят также к квазиньютоновским методам. Рассмотрение этих методов выходит за рамки настоящего пособия.

8.8.7. Методы случайного поиска

Рассматриваемые здесь методы основаны на использовании случайного механизма задания начальной точки и выбора направления движения. Так как в процессе поиска вычисляются значения только целевой функции, эти методы можно отнести к классу прямых.

Случайный механизм выбора направления реализуется с помощью датчика случайных чисел β , равномерно распределенных на интервале $[-b, b]$. Направление задается случайным вектором

$$\Xi = (\xi_1, \xi_2, \xi_3, \dots, \xi_n),$$

компоненты которого вычисляются по формуле

$$\xi_i = \frac{\beta_i}{\sqrt{\sum_{j=1}^n \beta_j^2}},$$

где n случайных чисел β_i генерируются датчиком. Очевидно, что такой случайный вектор имеет единичную длину и определяет только направление. При этом все направления равновероятны.

Приведем несколько простых алгоритмов случайного поиска.

Алгоритм с возвратом при неудачном шаге

1. Задать начальный шаг h^0 , число проб $s \leq n$ и точность ε .
2. Сгенерировать или задать начальную точку X^0 и вычислить в ней функцию f ; положить $i=1$ (i – счетчик проб).
3. Сгенерировать случайный вектор направления $\cdot \Xi_i^k$.
4. На направлении Ξ_i^k определить точку $X^{k+1} = X^k + h^k \cdot \Xi_i^k$ и вычислить в ней функцию f .
5. Проверить: а) если $f(X^{k+1}) < f(X^k)$, положить $k=k+1$, $i=1$ и вернуться на 3; б) если $f(X^{k+1}) \geq f(X^k)$ и $i < s$, положить $i = i+1$ и вернуться на 3; в)

если $f(X^{k+1}) \geq f(X^k)$, $i=s$ и $h^k > \varepsilon$, положить $h^k = h^k/2$, $i=1$ и вернуться на 3. г) если $f(X^{k+1}) \geq f(X^k)$, $i=s$ и $h^k \leq \varepsilon$, поиск закончить, приняв X^k за точку минимума. ▲

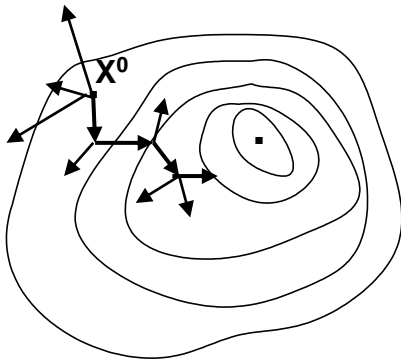


Рис. 8.38

Таким образом, поиск останавливается, если в текущей точке s направлений, сгенерированных подряд, оказались неудачными при шаге, меньшем заданной точности. На рис. 8.38 показан характер движения при поиске по данному алгоритму (жирной линией выделены успешные шаги).

Алгоритм с обратным шагом

Он построен путем модификации предыдущего алгоритма с целью уменьшить число розыгрышей направлений. Внесено следующее изменение: если сгенерированное направление неудачное, то после возврата в исходную точку X^k делается шаг в противоположном направлении; если же и оно неудачное, то генерируется новое направление из точки X^k . Логика действия очевидна: если в данном направлении функция ухудшается, то можно ожидать, что в противоположном направлении она улучшится. Понятно, что эффект модернизации алгоритма тем выше, чем ближе функция в текущей области к линейной.

Алгоритм наилучшей пробы

Здесь используются два шага: пробный α и рабочий h . Величина пробного шага соответствует необходимой точности. Задается также число пробных шагов m , меньшее числа переменных n , причем разница между m и n увеличивается с ростом n .

В текущей точке генерируются m направлений Ξ_j и на них делаются пробные шаги. Вычисляются изменения функции

$$\Delta f_j = f(\mathbf{x}^k + \alpha \cdot \Xi_j) - f(\mathbf{x}^k), \quad j = \overline{1, m}.$$

В направлении с наименьшим (отрицательным) приращением Δf_j выполняется рабочий шаг:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + h \cdot \Xi_j.$$

Поиск заканчивается, если $\forall \Delta f_j \geq 0$.

По аналогии с предыдущим алгоритмом можно рассматривать и неудачные направления: вычислить $\max_j |\Delta f_j|$. Если максимум соответствует положительному приращению Δf_j , то рабочий шаг делается в противоположном направлении.

Алгоритм статистического градиента

Все параметры и начальные действия такие же, как в предыдущем алгоритме. Отличие состоит в выборе направления движения. Оно определяется с учетом всех разыгранных направлений:

$$\mathbf{y} = \sum_{j=1}^m \Xi_j \cdot \Delta f_j. \quad (8.53)$$

Направление \mathbf{y} называется *статистическим градиентом*. В пределе ($m \rightarrow \infty$) он стремится к градиенту ∇f . Но определение \mathbf{y} требует меньше вычислений, чем ∇f , и тем в большей степени, чем сильнее неравенство $m < n$.

Согласно (8.53) каждый компонент вектора \mathbf{y} вычисляется по формуле

$$y_i = \sum_{j=1}^m \xi_{ij} \cdot \Delta f_j, \quad i = \overline{1, n}.$$

Новая точка находится перемещением на рабочий шаг h в направлении статистического антиградиента:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - h \cdot \frac{\mathbf{y}^k}{\|\mathbf{y}^k\|}.$$

Поиск завершается при выполнении условия $\|\mathbf{y}^k\| < \varepsilon$ или $\forall \Delta f_j \geq 0$. ▲

Эффективность рассмотренных алгоритмов можно повысить за счет незначительных изменений их отдельных элементов или шагов. Так, при успешном шаге можно продолжать шаги в найденном направлении до тех пор, пока функция улучшается (использование идеи одномерной минимизации).

Другая возможная модификация заключается во введении в алгоритм процедуры самообучения (адаптации). Это касается вероятностных характеристик случайного вектора Ξ . Первоначально все направления этого вектора равновероятны. В процессе поиска накапливается информация об эффективности разыгранных направлений, на основе которой корректируется распределение вероятностей направлений случайного вектора. В результате повышается вероятность удачных направлений при одновременном снижении вероятности неудачных.

Как следует из логики алгоритмов случайного поиска, при прочих равных условиях их эффективность выше вдали от экстремума и снижается по мере приближения к нему, так как падает вероятность выпадения удачных направлений.

Целесообразность применения случайного поиска возрастает с увеличением числа переменных, а при средней и большой размерности он становится практически безальтернативным. В отличие от методов, использующих производные, алгоритмы случайного поиска, как и прямые методы, могут применяться в условиях помех (погрешностей вычислений).

8.8.8. Генетические алгоритмы

По своей концепции генетические алгоритмы близки к методам случайного поиска. В них сочетаются элементы случайности и детерминированности, что характерно для всех природных процессов. Заимствованием механизмов живой природы и обусловлено название "генетические".

Идея генетических алгоритмов была высказана Д.Холландом в 60-е годы и затем развита им в фундаментальной работе "Адаптация в естественных и искусственных системах"(1975). Ипользовать их в численной оптимизации предложил Д.Гольдберг (1995). Но в основном генетические алгоритмы находят применение для обучения нейронных сетей. Поэтому здесь они будут рассмотрены только на уровне идей.

Как уже отмечалось, генетические алгоритмы строятся по принципам функционирования живой природы. Детерминированная составляющая алгоритмов в большей степени представлена в моделировании процессов отбора, размножения и наследования, а случайная – процесса мутации.

Любая альтернатива (вариант) представляется в виде строки (как правило, битовой) фиксированной длины, с которой манипулируют вне связи с содержанием задачи. Строка как абстрактная модель называется *кодом*. В коде в общем случае представлен набор параметров, зависящих от аргументов целевой функции. Код и его структура определяют *генотип*. Применительно к задачам оптимизации экземпляры кода, называемые *хромосомами* или *особями*, есть точки в пространстве поиска. Совокупность особей образует *популяцию*, а последовательные популяции – *поколения*.

Основными параметрами конкретного алгоритма являются размер популяции и вероятности применения операторов. Первоначально по содержанию задачи формируется генотип и создается исходная популяция. Размер популяции N рекомендуется брать исходя из оценки числа возможных альтернатив r :

$$N = 2 \log_2 r.$$

К текущей популяции применяется оператор отбора, в результате чего образуется множество родительских пар. При максимизации целевой функции вероятность особи стать родителем может вычисляться, например, по очевидной формуле

$$p_i = \frac{f_i}{\sum_i f_i}, \quad (8.54)$$

где f_i – значение критерия на i -й особи.

Для создания потомков используется оператор скрещивания, моделирующий процесс наследования за счет передачи части свойств от родителей к потомкам. Вероятность его применения рекомендуется брать не ниже 0,9. Он производит обмен подстроками родительских особей и,

следовательно, от пары родителей образуется два потомка. Как это происходит, зависит от выбранной процедуры скрещивания. Например, при длине строки n из первых $n-1$ равновероятных натуральных чисел разыгрывается одно число, которое принимается за *точку разбиения*. Затем подстроки родителей, находящиеся справа от точки разбиения, меняются местами.

К новым особям применяется оператор мутации. Вместе с оператором скрещивания он позволяет расширить область поиска, получить особи со свойствами, отсутствующими у родителей. Вероятность мутации берется обычно не выше 0,01. Процесс мутации заключается в случайной перестановке двух элементов строки, в смене значения случайно взятого элемента строки с 0 на 1 или наоборот, в циклическом сдвиге элементов строки и т.п.

Добавление потомков приводит к расширению популяции. В алгоритмах стационарного состояния все поколения имеют одинаковый размер. Поэтому на следующем шаге алгоритма производится сокращение популяции оператором редукции. Вероятность его применения к особи можно определить через вероятность отбора p_i :

$$s_i = \frac{\frac{1}{p_i}}{\sum_j \frac{1}{p_j}}.$$

На последнем шаге цикла проверяется условие останова. В качестве критерия останова используют число поколений, качество поколения (пороговое значение), близость особей между собой и др.

Для повышения эффективности генетических алгоритмов предлагаются способы распараллеливания вычислений, сокращения размера популяции за счет выделения кластеров и замены каждого одной особью, разрабатываются алгоритмы с изменяемым размером популяции.

Отличительной чертой генетических алгоритмов является одновременное использование набора точек в пространстве поиска вместо перехода от точки к точке в традиционных методах. Эта особенность позволяет применять такие алгоритмы для решения многоэкстремальных задач.

В заключение приведем небольшой пример, иллюстрирующий одну итерацию алгоритма.

Пример 8.7. Найти целочисленный максимум функции $f(x_1, x_2) = 675 - x_1^2 - x_2^2 + 20x_1 + 10x_2$ на области $5 \leq x_1 \leq 36, 2 \leq x_2 \leq 17$.

Сначала сделаем преобразования: $x_1 = y_1 + 5, x_2 = y_2 + 2$, тогда $0 \leq y_1 \leq 31, 0 \leq y_2 \leq 15$. В этом случае новые аргументы функции y_1, y_2 можно представить в двоичной системе, а за код принять строку из 9 бит. Первые 5 бит будут отражать y_1 , остальные – y_2 . Определим размер популяции:

$N=2\log_2(16*32)=18$. Ограничимся $N=10$. Используя датчик случайных чисел, создаем исходную популяцию и для всех ее особей вычисляем значение f и вероятность отбора p_i (табл. 8.1).

Таблица 8.1

№ строки	Экз. кода	y_1	y_2	x_1	x_2	f	p_i
1	011001010	12	10	17	12	702	0,1199
2	100001101	16	13	21	15	579	0,0989
3	110011100	25	12	30	14	319	0,0545
4	001011010	5	10	10	12	751	0,1283
5	010001011	8	11	13	13	727	0,1242
6	011101000	14	8	19	10	694	0,1186
7	101010111	21	7	26	9	528	0,0902
8	111010001	29	1	34	3	220	0,0376
9	001101110	6	14	11	16	678	0,1158
10	100010010	17	2	22	4	655	0,1119

Действия оператора отбора, использующего распределение p_i , привели к выбору родительских пар (2,7), (5,10), (6,9), (1,4), (4,9). В каждой паре точки разбиения выбираются независимо с помощью датчика целых случайных чисел от 1 до 8 с равными вероятностями. Результаты работы оператора скрещивания приведены в табл. 8.2. Точки разбиения показаны косой чертой. Значения аргументов и функции даны для потомков.

Таблица 8.2

№ строки	Экз. кода родителей	Экз. кода потомков	y_1	y_2	x_1	x_2	f
2	100/001101	100010111	17	7	22	9	640
7	101/010111	101001101	20	13	25	15	475
5	0100010/11	010001010	8	10	13	12	742
10	1000100/10	100010011	17	3	22	5	656
6	01110/1000	011101110	14	14	19	16	598
9	00110/1110	001101000	6	8	11	10	774
1	01/1001010	011011010	13	10	18	12	687
4	00/1011010	001001010	4	10	9	12	750
4	00101/1010	001011110	5	14	10	16	679
9	00110/1110	001101010	6	10	11	12	750

Теперь предположим, что оператор мутации сработал для 7 потомка, поменяв местами биты 1 и 2. Код этого потомка сменился с 011101110 на 101011010, значение функции стало 495. Расширенная популяция включает 10 исходных особей и 10 потомков. Сокращаем популяцию до

принятого размера, отбрасывая особи с меньшими значениями функции. Полученная после этого популяция первого поколения представлена в табл. 8.3. В ней только первые 4 особи из исходной популяции.

Таблица 8.3

№ строки	Экз. кода	x_1	x_2	f
1	011001010	17	12	702
2	001011010	10	12	751
3	010001011	13	13	727
4	011101000	19	10	694
5	010001010	13	12	742
6	001101000	11	10	774
7	011011010	18	12	687
8	001001010	9	12	750
9	001011110	10	16	679
10	001101010	11	12	750

Из сравнения данных в таблицах 8.1 и 8.3 видно, что новая популяция лучше исходной. Если среднее значение целевой функции на начальной популяции было равно 585,3, то на новом поколении оно возросло до 725,6.

Следующий шаг заключается в проверке условия окончания поиска. Оно может быть представлено в разных вариантах: $\min f_i > f_{\min}$ или $\max f_i - \min f_i < \varepsilon_f$, или $n \geq n_{\text{задан}}$. В первом варианте все особи должны быть лучше заданного уровня качества, во втором критерием выступает близость особей, в третьем – число поколений (итераций). Если предположить, что выбранное условие выполняется на первом поколении, то поиск на этом заканчивается. За решение принимаем лучший результат: $x_1=11$, $x_2=10$, $f=774$. Оптимальному решению соответствуют значения: $x_1=10$, $x_2=5$, $f=800$. Сравнение показывает, что по значению критерия получено неплохое приближение к оптимуму. Понятно, что продолжение итераций приведет к дальнейшему улучшению популяции и получению более близкого к оптимальному решения. ▲

Из описанной схемы генетического алгоритма и приведенного примера можно сделать вывод, что самым важным этапом применения алгоритма является определение генотипа (структуры кода) и для каждой задачи он индивидуален. Кроме того, специфика задачи может диктовать и требования к работе операторов. Например, в задаче коммивояжера оператор скрещивания не должен порождать потомков, в коде которых один пункт представлен более одного раза. Таким образом, по своей сути генетические алгоритмы являются эвристическими.

8.9. Методы условной оптимизации

В общем случае задача может содержать ограничения в виде равенств и неравенств, которые и определяют допустимое множество D . Наличие ограничений существенно усложняет решение задачи оптимизации. В большинстве случаев условный оптимум достигается на границе допустимого множества, при этом даже унимодальная функция может иметь на границе несколько локальных экстремумов.

Из большого числа методов условной оптимизации можно выделить 3 основные группы:

- методы *возможных направлений*: метод проектирования градиента, методы Зойтендейка, Вулфа и др.;
- методы *штрафных и барьерных функций*;
- модификации методов безусловной оптимизации.

Методы первой группы отличаются способом определения возможных направлений. Направление \mathbf{d} в точке $\mathbf{x}^k \in D$ называется *возможным направлением*, если существует $\lambda \neq 0$, при котором

$$\mathbf{x}^k + \lambda \cdot \mathbf{d} \in D,$$

$$f(\mathbf{x}^k + \lambda \cdot \mathbf{d}) < f(\mathbf{x}^k).$$

Эти условия означают, что на направлении \mathbf{d} найдутся *допустимые* точки, в которых значение функции лучше, чем в точке \mathbf{x}^k .

Ниже рассматривается один из методов возможных направлений.

8.9.1. Метод проектирования градиента

Градиент дает направление, в котором функция возрастает с наибольшей скоростью. Однако при условной оптимизации оно, как правило, не является возможным направлением. Поэтому используют не сам градиент (антиградиент), а его проекцию на поверхность ограничений, точнее, на плоскость, аппроксимирующую эту поверхность в текущей точке. Очевидно, что проекция градиента определяет направление наискорейшего изменения функции на поверхности ограничений.

Приведем один из вариантов метода проектирования градиента сначала для задачи с ограничениями-равенствами, а затем для общего случая. Метод применим, если целевая функция и все функции ограничений дифференцируемы.

Пусть ограничения заданы в виде

$$\psi_j(\mathbf{x}) = 0, \quad j = \overline{1, m}; \quad m < n. \quad (8.55)$$

Найдем возможное направление \mathbf{l} , на котором скорость изменения целевой функции максимальна:

$$\frac{df}{dl} = \sum_{i=1}^n \frac{\partial f}{\partial x_i} \frac{dx_i}{dl} \Rightarrow \max_l. \quad (8.56)$$

В допустимой области D функции ψ_j постоянны. Поэтому искомое направление должно удовлетворять системе равенств

$$\frac{d\psi_j}{dl} = \sum_{i=1}^n \frac{\partial \psi_j}{\partial x_i} \cdot \frac{dx_i}{dl} = 0, \quad j = \overline{1, m}. \quad (8.57)$$

Из связи направления с координатами следует:

$$dl^2 = \sum_{i=1}^n (dx_i)^2,$$

что перепишем в виде

$$1 - \sum_{i=1}^n \left(\frac{dx_i}{dl} \right)^2 = 0. \quad (8.58)$$

Таким образом, для нахождения наилучшего возможного направления необходимо решить задачу оптимизации (8.56) – (8.58). Так как условия имеют вид равенств, а функции дифференцируемы, для решения этой вспомогательной задачи воспользуемся методом Лагранжа.

Запишем функцию Лагранжа:

$$F\left(\frac{d\mathbf{x}}{dl}, \Lambda\right) = \sum_{i=1}^n \frac{\partial f}{\partial x_i} \frac{dx_i}{dl} + \lambda_0 \left(1 - \sum_{i=1}^n \left(\frac{dx_i}{dl} \right)^2 \right) + \sum_{j=1}^m \lambda_j \sum_{i=1}^n \frac{\partial \psi_j}{\partial x_i} \frac{dx_i}{dl}. \quad (8.59)$$

Неизвестными в ней являются векторы $\frac{d\mathbf{x}}{dl}$ и Λ . Взяв производные $\frac{\partial F}{\partial \left(\frac{dx_i}{dl} \right)}$, $i = \overline{1, n}$ и приравняв их нулю, получаем

$$\frac{\partial f}{\partial x_i} - 2\lambda_0 \frac{dx_i}{dl} + \sum_{j=1}^m \lambda_j \frac{\partial \psi_j}{\partial x_i} = 0, \quad i = \overline{1, n}.$$

Отсюда выразим компоненты искомого вектора:

$$\frac{dx_i}{dl} = \frac{1}{2\lambda_0} \left(\frac{\partial f}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial \psi_j}{\partial x_i} \right), \quad i = \overline{1, n}. \quad (8.60)$$

Подставив (8.60) в (8.58), находим:

$$2\lambda_0 = \sqrt{\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial \psi_j}{\partial x_i} \right)^2}.$$

С учетом этого выражения формула (8.60) принимает вид

$$\frac{dx_i}{dl} = \frac{\frac{\partial f}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial \psi_j}{\partial x_i}}{\sqrt{\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial \psi_j}{\partial x_i} \right)^2}}. \quad (8.61)$$

Для определения неизвестных множителей λ_j воспользуемся системой (8.57). Подставив в нее (8.61), получаем:

$$\sum_{i=1}^n \frac{\partial \psi_k}{\partial x_i} \left(\frac{\partial f}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial \psi_j}{\partial x_i} \right) = 0, \quad k = \overline{1, m}.$$

После раскрытия скобок окончательно имеем:

$$\sum_{i=1}^n \frac{\partial \psi_k}{\partial x_i} \sum_{j=1}^m \lambda_j \frac{\partial \psi_j}{\partial x_i} = - \sum_{i=1}^n \frac{\partial \psi_k}{\partial x_i} \frac{\partial f}{\partial x_i}, \quad k = \overline{1, m}. \quad (8.62)$$

Так как направление ищется в конкретной точке, то все производные в (8.62) – известные константы. Поэтому система уравнений (8.62) является линейной системой относительно λ_j . Ее решение не вызывает затруднений (при условии, что матрица системы не является особенной). Найдя значения λ_j и подставив их в (8.61), получаем компоненты проекции градиента (в знаменателе (8.61) имеем длину проекции градиента). Движение осуществляется в направлении, противоположном проекции.

Аналогично градиентному методу новая точка вычисляется по формуле

$$x_i^{k+1} = x_i^k - h^k \frac{\frac{\partial f}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial \psi_j}{\partial x_i}}{\sqrt{\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial \psi_j}{\partial x_i} \right)^2}}, \quad i = \overline{1, n}. \quad (8.63)$$

Алгоритм.

1. Задать начальную точку, удовлетворяющую системе (8.55), начальное значение h^0 и точность по величине проекции градиента ε .
2. В текущей точке вычислить градиенты всех функций (f и ψ_j) и решить систему (8.62).
3. Вычислить проекцию градиента по формуле (8.61).
4. Проверить: если $\sqrt{\sum_{i=1}^n \left(\frac{\partial f}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial \psi_j}{\partial x_i} \right)^2} \leq \varepsilon$, завершить поиск.
5. Вычислить новую точку по формуле (8.63).
6. Проверить: если значение целевой функции улучшилось, перейти на шаг 2, иначе уменьшить h^k в два раза и перейти на шаг 5. ▲

Качественный характер работы алгоритма иллюстрирует рис. 8.39. Здесь функции зависят от 2-х переменных, поэтому в каждой точке на

линии ограничения может быть всего 2 направления, лучшее из которых определяет проекция градиента. В многомерной задаче таких направлений бесконечное множество.

При линейных ограничениях могут возникать проблемы поиска лишь при очень малых значениях градиентов функций ограничений и совпадении их направлений,

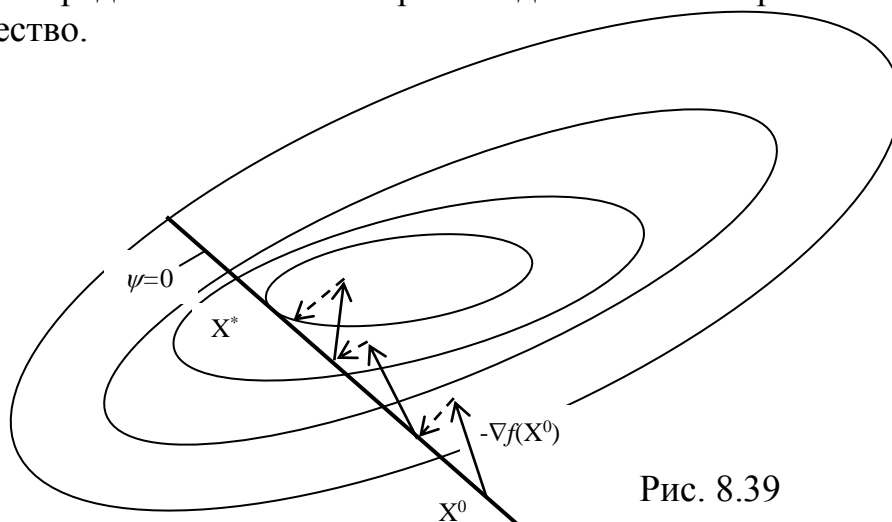


Рис. 8.39

так как это приводит к вырожденности матрицы системы (8.62).

В случае нелинейных ограничений движение на основе линейной аппроксимации нарушает равенства. Поэтому в алгоритм необходимо внести изменения, обеспечивающие выполнение равенств с необходимой точностью ε_ψ . С этой целью алгоритм дополняется проверкой величины невязки, которая выполняется в каждой новой точке. Если $\sum_{j=1}^m (\psi_j)^2 > \varepsilon_\psi$, то включается процедура коррекции, заключающаяся в минимизации невязки:

$$\sum_{j=1}^m (\psi_j)^2 \Rightarrow \min_{x_i}. \quad (8.64)$$

Для решения задачи (8.64) можно применить любой метод безусловной оптимизации, но в данном контексте целесообразен метод градиентов. При этом значения $(n-m)$ переменных фиксируются, так как для выполнения равенств достаточно m переменных. Понятно, что при частых коррекциях трудоемкость алгоритма значительно возрастает.

Теперь рассмотрим случай, когда ограничения заданы в виде неравенств $\varphi_j(\mathbf{x}) \leq 0$. Поиск начинается из точки, в которой удовлетворяются все ограничения. Пока они выполняются как строгие, движение осуществляется по градиентному методу. Когда достигается граница допустимого множества, одно или несколько ограничений обращаются в равенства. Такие ограничения называют активными. В точках с активными ограничениями направление движения определяется по проекции градиента на поверхность этих ограничений, то есть применяется вышеприведенный алгоритм. Пример поиска минимума при двух линейных неравенствах показан на рис. 8.40. Очевидно, что при смешанных и нелинейных ограничениях алгоритм весьма существенно

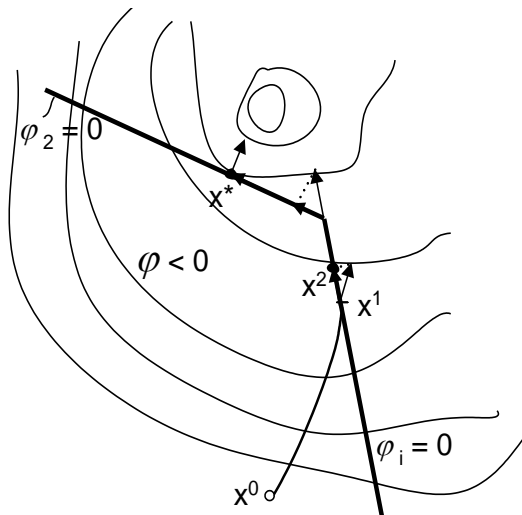


Рис. 8.40

усложняется и требует большого объема вычислений. Поэтому метод проектирования градиентов целесообразно применять при линейных ограничениях.

8.9.2. Метод штрафных функций

Совершенно иной подход используется в методах штрафных и барьерных функций. Ограничения задачи специальным образом отражаются в критерии, в результате чего критерий модифицируется, а исходная задача на условный экстремум сводится к задаче на

безусловный экстремум.

В методе штрафных функций в критерий вводится штраф при нарушении условий задачи. Пусть в общем случае имеем задачу

$$f(\mathbf{x}) \rightarrow \min; \quad (8.65)$$

$$\varphi_i(\mathbf{x}) \leq 0, \quad i = \overline{1, m_1}; \quad (8.66)$$

$$\psi_i(\mathbf{x}) = 0, \quad i = \overline{1, m_2}. \quad (8.67)$$

Тогда можно построить вспомогательную функцию

$$\Theta(\mathbf{x}) = f(\mathbf{x}) + \alpha H(\mathbf{x}), \quad (8.68)$$

где $H(\mathbf{x})$ – функция штрафа, α – параметр штрафа.

Вспомогательная функция играет роль модифицированного критерия, который при выполнении всех ограничений должен совпадать с исходным. Поэтому необходимо, чтобы в допустимой области $H(\mathbf{x})$ равнялась нулю, а вне ее была положительной. Для задачи (8.65)-(8.67) функция штрафа включает две составляющие $H(\mathbf{x}) = H_\varphi(\mathbf{x}) + H_\psi(\mathbf{x})$, учитывающие ограничения-неравенства и ограничения-равенства соответственно и удовлетворяющие условиям

$$H_\varphi(\cdot) = \begin{cases} 0, & \forall \varphi_i \leq 0; \\ > 0, & \exists \varphi_i > 0; \end{cases} \quad H_\psi(\cdot) = \begin{cases} 0, & \forall \psi_i = 0; \\ > 0, & \exists \psi_i \neq 0. \end{cases} \quad (8.69)$$

Возможны разные конструкции функций, обладающих указанными свойствами. Типичные представители составляющих штрафной функции имеют вид

$$H_\varphi(\mathbf{x}) = \sum_{i=1}^{m_1} [\max\{0, \varphi_i(\mathbf{x})\}]^p,$$

$$H_\psi(\mathbf{x}) = \sum_{i=1}^{m_2} |\psi_i(\mathbf{x})|^p,$$

где p – натуральное число. Для дифференцируемости функций берут четные значения p , обычно $p = 2$.

Чем больше α , тем сильнее влияет функция штрафа и, значит, тем точнее выполняются условия задачи.

Пример 8.8. Дана задача

$$\begin{aligned} f(x) &= x \rightarrow \min; \\ \varphi(x) &= 3 - x \leq 0. \end{aligned}$$

Ответ очевиден: $x^* = 3$. Теперь сведем эту задачу к определению безусловного экстремума вспомогательной функции. Построим штрафную функцию в соответствии с (8.69): $H = [\max(0, 3-x)]^2$. Тогда приходим к задаче

$$\Theta = x + \alpha [\max(0, 3-x)]^2 \rightarrow \min.$$

На рис. 8.41 и 8.42 показаны функции αH и Θ для двух значений α . Видно, что точки минимума вспомогательной функции с увеличением α приближаются к точке условного минимума исходной задачи. Такой же вывод следует из аналитического решения. Действительно, при $x < 3$

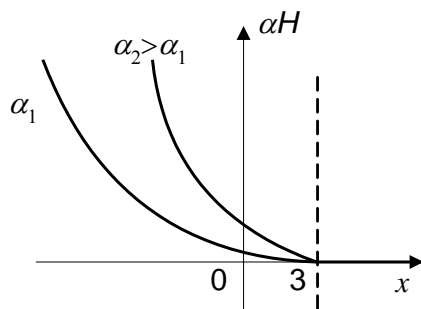


Рис. 8.41

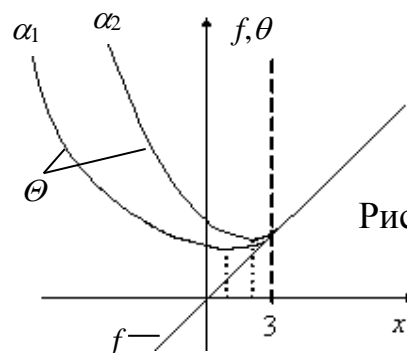


Рис. 8.42

вспомогательная функция имеет вид

$$\Theta = x + \alpha (3 - x)^2.$$

Находим минимум этой функции:

$$\frac{d\Theta}{dx} = 0 \Rightarrow x_\alpha^* = 3 - \frac{1}{2\alpha}.$$

Отсюда получаем $\lim_{\alpha \rightarrow \infty} x_\alpha^* = x^* = 3$.

Пример 8.9. Рассмотрим влияние параметра шага в задаче

$$f = (1 - x_1)^2 + 5(x_2 - x_1^2)^2 \rightarrow \min,$$

$$x_2 = \frac{2}{3}x_1 - 4.$$

Здесь $\psi = x_2 - \frac{2}{3}x_1 + 4$ и $\theta = (1 - x_1)^2 + 5(x_2 - x_1^2)^2 + \alpha(x_2 - \frac{2}{3}x_1 + 4)^2$. На

рис. 8.43 построены линии уровня функции θ для разных значений α и линия ограничения ψ . При $\alpha=0$ имеем $\theta=f$, и минимум θ достигается в точке безусловного минимума f : $x_1=x_2=1$. С увеличением α меняется форма

линий уровня θ и положение минимума. При $\alpha=1$ минимум θ смещается к

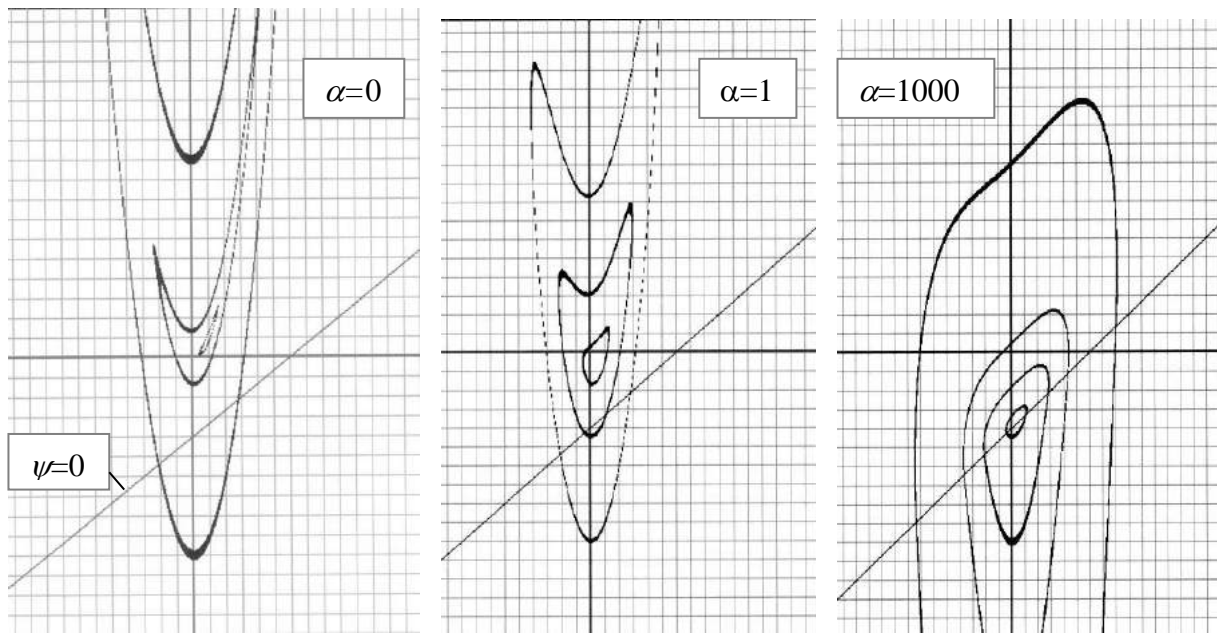


Рис. 8.43

линии ограничения, а при $\alpha=1000$ он практически точно совпадает с условным минимумом задачи.

В обоих примерах с увеличением α генерируемые точки приближаются к оптимальному решению извне допустимого множества. Поэтому ряд авторов называют рассматриваемый метод методом *внешних штрафов*.

Таким образом, чтобы безусловный минимум вспомогательной функции был близок к условному минимуму решаемой задачи, необходимо брать очень большое значение α , теоретически бесконечное. Однако при больших α возникают серьезные трудности при поиске минимума вспомогательной функции. Поэтому предлагается решать последовательность задач минимизации Θ с возрастающими значениями α . При этом в качестве начальной точки следующей задачи берется оптимальная точка предыдущей. Такой прием использован в следующем алгоритме штрафных функций.

Алгоритм.

1. Задать: начальную точку \mathbf{x}^0 , точность ε , начальное значение α_0 и число $\beta > 1$.
2. Минимизировать $\Theta(\mathbf{x})$ одним из методов безусловной оптимизации, в результате чего определяется \mathbf{x}_k^* .
3. Проверить: если $\alpha_k H(\mathbf{x}_k^*) < \varepsilon$, то остановиться, приняв \mathbf{x}_k^* за оптимальное решение задачи.
4. Положить $\alpha_{k+1} = \beta \alpha_k$, за начальную точку принять \mathbf{x}_k^* и вернуться на шаг 2. ▲

Рекомендуется выбирать значения параметров алгоритма из диапазонов: $\alpha_0 \in (0,1]$, $\beta \in (1,10]$. Начальную точку следует задавать в недопустимой области.

Пример 8.10. Алгоритмом штрафных функций решить задачу

$$f = 4x_1^2 + 3x_2^2 - 4x_1x_2 + x_1 \rightarrow \min,$$

$$4x_1^2 + 9x_2^2 = 36.$$

Возьмем начальную точку $\mathbf{x}^0 = (-5; 5)$, $\alpha_0 = 0,21$, $\beta = 5$ и $\varepsilon = 0,0001$. Применяя для минимизации Θ метод Ньютона, получаем результаты, представленные в табл. 8.4.

Таблица 8.4

№ итерации	α	x_1	x_2	f	Θ	αH
0	0.21	-5	5	270	283.533	13.533
1	1.05	-0.191	-0.132	-0.094	0.939	1.032
2	5.25	-0.209	-0.169	-0.09	5.035	5.125
3	26.25	-0.654553	-1.059105	1.651353	13.504372	11.853019
4	131.25	-0.990765	-1.731532	5.068137	7.691651	2.623514
5	656.25	-1.046856	-1.843717	5.814225	6.343889	0.529664
6	3281.25	-1.057736	-1.865478	5.964774	6.070887	0.106113
7	16406.25	-1.059899	-1.869804	5.994933	6.016163	0.02123
8	82031.25	-1.060331	-1.870668	6.000967	6.005213	0.004246
9	410156.25	-1.060417	-1.870841	6.002174	6.003023	0.000849
10	2050781.25	-1.060434	-1.870876	6.002415	6.002585	0.00017
11	$>10^7$	-1.060434	-1.870884	6.002469	6.002503	3.397E-05

Как следует из таблицы, решение с заданной высокой точностью получено за 11 итераций алгоритма. Заметим, что несмотря на увеличение α значение αH сходится к нулю, обеспечивая сходимость алгоритма. Траектория поиска и линии уровня функции f изображены на рис. 8.44. ▲

Другой пример поиска методом штрафных функций приведен на рис. 8.45 для задачи

$$f = 4(x_1 - 2)^2 + (x_2 - x_1^2)^2 \rightarrow \min,$$

$$x_2 = 0,2x_1^2 - 6.$$

Поиск проведен из начальной точки $(-2; -7)$ с параметрами $\alpha_0 = 0,1$ и $\beta = 2$. Здесь, как и в предыдущем примере, на первой итерации алгоритма движение направлено в

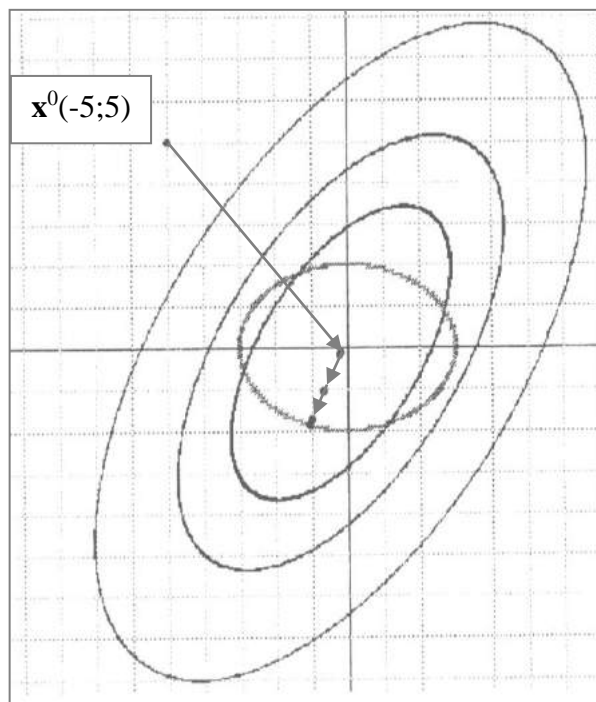


Рис. 8.44

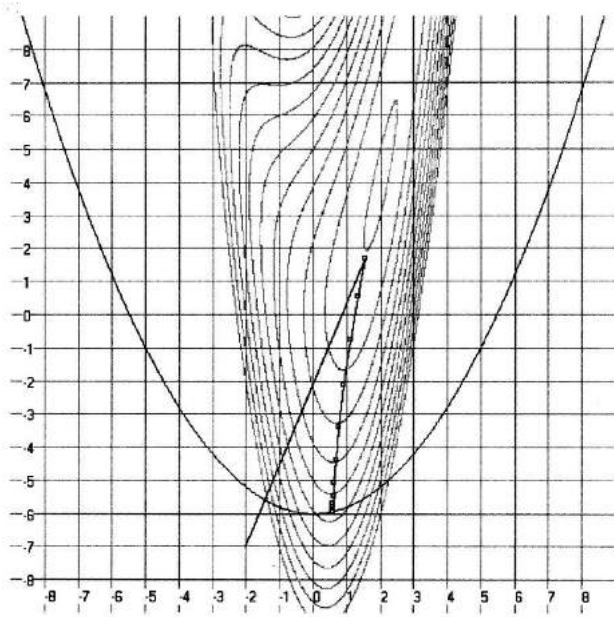


Рис. 8.45

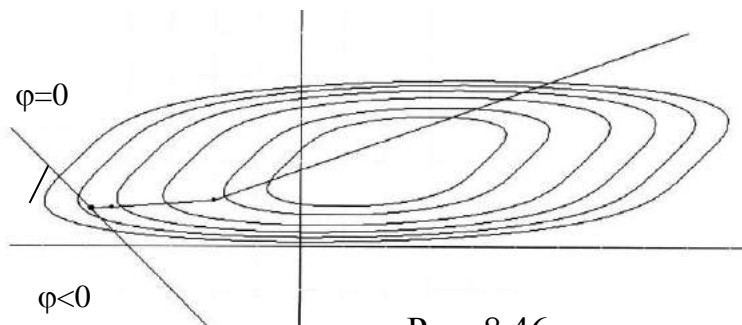


Рис. 8.46

сторону безусловного минимума целевой функции. Это объясняется небольшим начальным значением параметра штрафа, при котором основное влияние на направление оказывает целевая функция. С возрастанием α направление поиска ориентируется на условный экстремум.

Еще один пример поиска показан на рис. 8.46 для задачи

$$f = (x_2 - x_1)^2 + 4(x_2 - 2)^4 \rightarrow \min,$$

$$\varphi = x_1 + x_2 + 4 \leq 0.$$

Здесь поиск производился при $\alpha_0=1$ и $\beta=10$. Такие параметры обусловили другой характер движения к условному минимуму: первая итерация уже не приводит в окрестность безусловного экстремума и траектория не имеет резких изменений направления поиска.

Таким образом, выбор параметров поиска имеет существенное влияние на эффективность алгоритма.

8.9.3. Метод барьерных функций

В отличие от метода штрафных функций, данный метод применим к задачам с ограничениями только в виде неравенств.

Суть метода заключается в том, что поиск начинается обязательно из внутренней точки и последующие точки не должны выходить из допустимой области. С этой целью задача модифицируется так, что при приближении к границе допустимой области растет барьер, препятствующий выходу на границу.

Исходная задача на условный экстремум задается в виде

$$f(\mathbf{x}) \rightarrow \min; \quad (8.70)$$

$$\varphi_i(\mathbf{x}) \leq 0, \quad i = \overline{1, m}. \quad (8.71)$$

Она преобразуется в задачу безусловной минимизации вспомогательной функции

$$\Theta(\mathbf{x}) = f(\mathbf{x}) + \mu B(\mathbf{x}),$$

где $B(\mathbf{x})$ – барьерная функция, μ – параметр барьера.

Обязательное условие: внутренность области не должна быть пустой (имеются точки, в которых $\forall \varphi_i(\mathbf{x}) < 0$).

Барьерная функция строится так, чтобы она была неотрицательной и непрерывной на допустимом множестве и стремилась к бесконечности при приближении изнутри к границе:

$$B(\cdot) = \begin{cases} \geq 0, & \forall \varphi_i < 0; \\ \infty, & \exists \varphi_i = 0. \end{cases}$$

Как и в случае штрафной функции, существует несколько конструкций $B(\mathbf{x})$, удовлетворяющих этим условиям. Но в основном используется барьерная функция в виде

$$B(\mathbf{x}) = \sum_{i=1}^m \frac{-1}{\varphi_i(\mathbf{x})}. \quad (8.72)$$

Понятно, что решение вспомогательной задачи зависит от значения параметра барьера. Покажем на задаче из примера 8.7 влияние μ на результат минимизации Θ .

Пример 8.11. Исходная задача:

$$\begin{aligned} f(x) &= x \rightarrow \min; \\ \varphi(x) &= 3 - x \leq 0. \end{aligned}$$

Барьерную функцию строим согласно (8.72). Тогда вспомогательная функция имеет вид

$$\Theta = x + \mu \cdot \frac{-1}{(3-x)} = x + \frac{\mu}{x-3}.$$

Находим точку минимума Θ :

$$\frac{d\Theta}{dx} = 0 \Rightarrow x_{\mu}^* = 3 + \sqrt{\mu}.$$

Отсюда получаем

$$\lim_{\mu \rightarrow 0} x_{\mu}^* = x^* = 3.$$

Следовательно, с уменьшением μ точки минимума вспомогательной функции приближаются к минимуму исходной задачи. Геометрическая иллюстрация решения приведена на рис. 8.47.

Как хорошо видно на рис. 8.47, при оптимуме исходной задачи на границе допустимого множества последовательность точек минимума Θ с уменьшением μ приближается к оптимальному решению изнутри допустимой области. По этой причине метод барьеров называют еще методом *внутренних штрафов*.

В связи с возможными трудностями поиска при малых значениях μ решается не одна, а последовательность вспомогательных задач с уменьшающимися значениями параметра барьера.

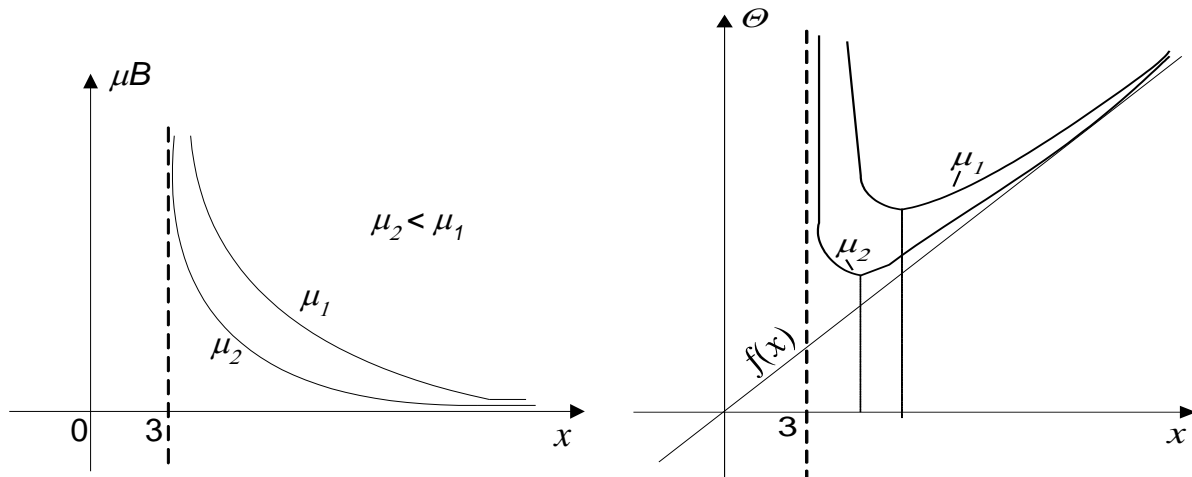


Рис. 8.47

Алгоритм.

1. Выбрать начальную точку \mathbf{x}^0 так, чтобы $\forall \varphi_i(\mathbf{x}^0) < 0$; задать точность ε , начальное значение μ_0 и число $\beta \in (0, 1)$.
2. Минимизировать $\Theta(\mathbf{x})$ одним из методов безусловной оптимизации, в результате чего определяется \mathbf{x}_k^* .
3. Проверить: если $\mu_k B(\mathbf{x}_k^*) < \varepsilon$, то остановиться, приняв \mathbf{x}_k^* за оптимальное решение задачи.
5. Положить $\mu_{k+1} = \beta \mu_k$, за начальную точку принять \mathbf{x}_k^* и вернуться на 2. ▲

Значение μ_0 можно брать из интервала $[2, 10]$. Важное замечание касается п.2 алгоритма: в процессе поиска минимума вблизи границы из-за дискретности шагов возможен выход за допустимую область, где барьерная функция становится отрицательной, что повлечет расхождение поиска. Поэтому необходима явная проверка на допустимость точек на каждом шаге при минимизации Θ .

Пример 8.10 (Базара, Шетти). Исходная задача:

$$f(\mathbf{x}) = (x_1 - 2)^4 + (x_1 - 2x_2)^2 \rightarrow \min;$$

$$\varphi(\mathbf{x}) = x_1^2 - x_2 \leq 0.$$

Решение находим, используя соответствующую вспомогательную функцию

$$\Theta = (x_1 - 2)^4 + (x_1 - 2x_2)^2 - \frac{\mu}{x_1^2 - x_2}.$$

За начальную точку возьмем допустимую точку $(0; 1)$, значения μ и β принимаем равными 10. Результаты поиска алгоритмом барьерных функций представлены в табл. 8.5 и на рис. 8.48.

Таблица 8.5

№ итерации	μ	x_1	x_2	f	Θ	μB
1	10	0.7079	1.5315	8.3338	18.0388	9.705
2	1.0	0.8282	1.1098	3.8214	6.1805	2.3591
3	0.1	0.8989	0.9638	2.5282	3.1701	0.6419
4	0.01	0.9294	0.9162	2.1291	2.3199	0.1908
5	0.001	0.9403	0.9011	2.0039	2.0629	0.0590
6	0.0001	0.94389	0.89635	1.9645	1.9829	0.0184

Как и следовало ожидать, с уменьшением μ значение μB стремится к нулю.

Завершая рассмотрение методов штрафных и барьерных функций, отметим, что можно построить алгоритм, использующий как штрафы, так и барьеры. Для этого достаточно записать смешанную вспомогательную функцию в виде

$$\Theta(\mathbf{x}) = f(\mathbf{x}) + \mu B(\mathbf{x}) + \frac{1}{\mu} H(\mathbf{x}),$$

где барьерная функция $B(\mathbf{x})$ применяется к неравенствам, а штрафная функция $H(\mathbf{x})$ – к ограничениям-равенствам. Последовательность задач минимизации Θ решается с уменьшающимися значениями параметра μ .

8.9.4. Другие методы условной оптимизации

Если все ограничения задачи заданы в виде неравенств, то для поиска условного минимума могут применяться модификации некоторых методов безусловной оптимизации. Так в методах случайного поиска модификация заключается в изменении условия успешности шага или направления. Шаг считается успешным, если он приводит в точку, в которой улучшается значение целевой функции и выполняются все ограничения. С этой целью добавляется проверка каждой точки на принадлежность допустимому множеству. В остальном алгоритмы остаются без изменений.

Аналогичное дополнение алгоритма Хука-Дживса делает его пригодным для поиска условного минимума. Генетические алгоритмы также могут использоваться для условной оптимизации. Для этого в них вводится детерминированный оператор жизнеспособности: если особь не

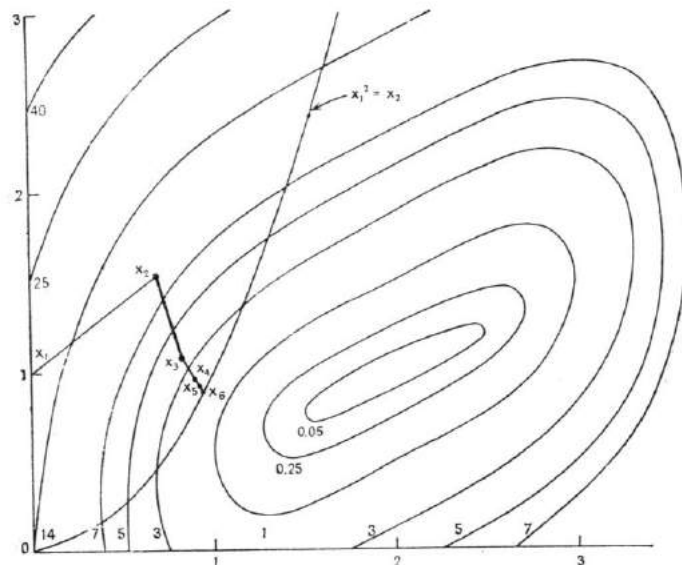


Рис. 8.48

удовлетворяет условиям "жизни", она погибает. Такой оператор должен применяться к каждой особи.

В данной главе не затронуты вопросы, возникающие при минимизации *многоэкстремальных* функций. Задача поиска глобального минимума многократно сложнее локальной оптимизации. Напомним, что при минимизации унимодальных функций в одних методах направление спуска выбирается по локальной модели целевой функции, например, линейной (градиентные методы) или квадратичной (метод Ньютона), в других – без использования модели, например, симплексный и случайные методы. В случае многоэкстремальной функции методы поиска строятся также на основе либо модели глобального поведения функции, либо эвристических представлений.

Несмотря на интенсивные исследования проблемы глобальной оптимизации сегодня нет эффективных методов, построенных на идее глобального поведения функции. Практическое применение находят в основном подходы, использующие локальный спуск из многих начальных точек с последующим выбором лучшего из найденных решений. Многократный спуск иногда называют *мультистартом*. Ему присущи такие недостатки как возможность неоднократного спуска в одну и ту же точку и отсутствие гарантии попадания в область притяжения глобального минимума. Эффективность мультистарта повышают за счет обеспечения выхода из "мелких" минимумов, исключения повторных спусков в найденные минимумы и т.п. С этой целью процессу спуска придают инерцию, которая позволяет проскакивать "неглубокие" минимумы (метод тяжелого шарика), изменяют целевую функцию для придания ей "туннельного эффекта", обеспечивающего переход из найденного в более глубокий минимум, используют редукцию задачи и другие идеи.

8.10. Задания для самостоятельной работы

1. Решить следующие задачи квадратичного программирования симплекс-методом с использованием правила ограниченного ввода. Результаты представить графически (допустимое множество и оптимумы).

$1) f = 2x_1 + 2x_2 - x_1^2 - 2x_2^2 + 2x_1x_2 \rightarrow \max$ $2x_1 + x_2 \leq 4$ $-x_1 + x_2 \leq 2$ $x_1, x_2 \geq 0$	$2) f = -x_1 + 6x_2 - x_1^2 - 3x_2^2 + 3x_1x_2 \rightarrow \max$ $4x_1 + 3x_2 \leq 12$ $-x_1 + x_2 \leq 1$ $x_1, x_2 \geq 0$
$3) f = 15x_1 + 8x_2 - x_1^2 - 3x_2^2 + x_1x_2 \rightarrow \max$ $3x_1 + x_2 \leq 15$ $x_1 + 2x_2 \leq 10$ $x_1, x_2 \geq 0$	$4) f = 15x_1 + 8x_2 - x_1^2 - 3x_2^2 + x_1x_2 \rightarrow \max$ $6x_1 + 2x_2 \leq 30$ $2x_1 + 4x_2 \leq 20$ $x_1, x_2 \geq 0$

5) $f = -x_1 + 6x_2 - x_1^2 - 3x_2^2 + 3x_1x_2 \rightarrow \max$ $-2x_1 + x_2 \leq 2$ $x_1 + x_2 \leq 3$ $x_1, x_2 \geq 0$	6) $f = -4x_1 + 8x_2 - x_1^2 - 1,5x_2^2 + 2x_1x_2 \rightarrow \max$ $3x_1 + 5x_2 \leq 15$ $x_1 - x_2 \leq 1$ $x_1, x_2 \geq 0$
7) $f = 4x_1 + 4x_2 - 3x_1^2 - x_2^2 + 2x_1x_2 \rightarrow \max$ $3x_1 + 6x_2 \leq 18$ $x_1 - 4x_2 \leq 4$ $x_1, x_2 \geq 0$	8) $f = 3x_1 + 10x_2 - 2x_1^2 - 5x_2^2 \rightarrow \max$ $4x_1 - x_2 \geq 8$ $x_1 + x_2 \leq 12$ $x_1, x_2 \geq 0$
9) $f = 4x_1 + 4x_2 - 3x_1^2 - x_2^2 + 2x_1x_2 \rightarrow \max$ $3x_1 + 4x_2 \leq 12$ $x_1 - 2x_2 \leq 2$ $x_1, x_2 \geq 0$	10) $f = 5x_1 + 10x_2 - x_1^2 - x_2^2 + 0,5x_1x_2 \rightarrow \max$ $8 - 2x_2 \geq 0$ $5 - x_1 - x_2 \geq 0$ $x_1, x_2 \geq 0$
11) $f = 4x_1 + 4x_2 - 2x_1^2 - 4x_2^2 + 4x_1x_2 \rightarrow \max$ $-x_1 + x_2 \leq 2$ $2x_1 + x_2 \leq 4$ $x_1, x_2 \geq 0$	12) $f = 30x_1 + 16x_2 - 2x_1^2 - 6x_2^2 + 2x_1x_2 \rightarrow \max$ $3x_1 + x_2 \leq 15$ $x_1 + 2x_2 \leq 10$ $x_1, x_2 \geq 0$
13) $f = -2x_1 + 12x_2 - 2x_1^2 - 6x_2^2 + 6x_1x_2 \rightarrow \max$ $-2x_1 + x_2 \leq 2$ $2x_1 + 2x_2 \leq 6$ $x_1, x_2 \geq 0$	14) $f = 2x_1 + 8x_2 - 4x_1^2 - 4x_2^2 + 2x_1x_2 \rightarrow \max$ $2x_1 + 4x_2 \leq 24$ $7x_1 + 5x_2 \geq 35$ $x_1, x_2 \geq 0$
15) $f = -5x_1 + 6x_2 - x_1^2 - 2x_2^2 + 2x_1x_2 \rightarrow \max$ $2x_1 + 6x_2 \leq 30$ $2x_1 + x_2 \leq 8$ $x_1, x_2 \geq 0$	16) $f = 15x_1 + 8x_2 - x_1^2 - 3x_2^2 + x_1x_2 \rightarrow \max$ $6x_1 + 2x_2 \leq 24$ $2x_1 + 4x_2 \leq 18$ $x_1, x_2 \geq 0$
17) $f = x_1 + 4x_2 - 2x_1^2 - 2x_2^2 + x_1x_2 \rightarrow \max$ $x_1 + 2x_2 \leq 12$ $7x_1 + 5x_2 \geq 35$ $x_1, x_2 \geq 0$	18) $f = 10x_1 + 20x_2 - 2x_1^2 - 2x_2^2 + x_1x_2 \rightarrow \max$ $4 - x_2 \geq 0$ $5 - x_1 - x_2 \geq 0$ $x_1, x_2 \geq 0$
19) $f = 3x_1 - 2x_2 - 0,5x_1^2 - x_2^2 + x_1x_2 \rightarrow \max$ $-x_1 + 2x_2 \leq 2$ $2x_1 - x_2 \leq 2$ $x_1, x_2 \geq 0$	20) $f = -x_1 + 6x_2 - x_1^2 - 3x_2^2 + 3x_1x_2 \rightarrow \max$ $-2x_1 + 2x_2 \leq 2$ $x_1 + 0,75x_2 \leq 3$ $x_1, x_2 \geq 0$
21) $f = 10x_1 - 12x_2 + 2x_1^2 + 4x_2^2 - 4x_1x_2 \rightarrow \min$ $x_1 + 3x_2 \leq 12$ $x_1 + x_2 \geq 6$ $x_1, x_2 \geq 0$	22) $f = 8x_1 + 8x_2 - 6x_1^2 - 2x_2^2 + 4x_1x_2 \rightarrow \max$ $3x_1 + 4x_2 \leq 12$ $x_1 - 2x_2 \leq 2$ $x_1, x_2 \geq 0$
23) $f = 2x_1 + 2x_2 - 1,5x_1^2 - 0,5x_2^2 + x_1x_2 \rightarrow \max$ $3x_1 + 6x_2 \leq 18$ $x_1 - 4x_2 \leq 4$ $x_1, x_2 \geq 0$	24) $f = 5x_1 - 6x_2 + x_1^2 + 2x_2^2 - 2x_1x_2 \rightarrow \min$ $x_1 + 3x_2 \leq 12$ $x_1 + x_2 \geq 6$ $x_1, x_2 \geq 0$

2. Найти минимум функции на заданном интервале $[a, b]$ а) методом золотого сечения, б) методом Фибоначчи. Данные приведены в таблице.

$$f = k(x-a)^\alpha (b-x)^\beta$$

№	k	a	b	α	β	ε
1	-0.5	2	10	1.2	3	0.1
2	-1.2	5	14	2	1.5	0.15
3	-2	1	12	3	1	0.15
4	-0.1	4	16	1	2	0.2
5	-3	10	15	2	2	0.05
6	-1.5	3	13	1.8	2.5	0.15
7	-2	-3	9	3	1.2	0.2
8	-1	-5	6	2.5	2	0,18

3. Найти безусловный минимум нижеследующих функций двух переменных а) методом Хука-Дживса, б) методом наискорейшего спуска, в) методом Ньютона. Спуск провести из двух различных начальных точек. Одномерный метод и точность выбрать по своему усмотрению.

$$3.1. f = x_1 - 6x_2 + x_1^2 + 3x_2^2 - 3x_1x_2,$$

$$3.2. f = (x_1 - 2)^4 + (x_1 - 2x_2)^2,$$

$$3.3. f = (x_1 - 2)^2 + 10(x_2 - x_1^2)^2,$$

$$3.4. f = (x_1 - 2)^4 + 6(x_1 - 2x_2)^2,$$

$$3.5. f = 10x_1 - 12x_2 + 2x_1^2 + 4x_2^2 - 4x_1x_2,$$

$$3.6. f = 50(x_1 - 2)^2 + 5(x_2 - x_1^2)^2.$$

4. Найти условный минимум методом штрафных функций в задачах:

$$4.1. f = (x_1 - 2)^2 + (x_2 - x_1^2)^2 \rightarrow \min,$$

$$x_2 = 0,1x_1^2 - 7.$$

$$4.2. f = x_1 + 4x_1^2 + 3x_2^2 - 4x_1x_2 \rightarrow \min,$$

$$x_2 \leq x_1 - 3.$$

$$4.3. f = (x_1 - 2)^4 + 20(x_1 - 2x_2)^2 \rightarrow \min,$$

$$x_2 \geq 2x_1 + 2.$$

Примечание. Задания 3 и 4 можно дополнить требованием представить графически траектории поиска из нескольких начальных точек на фоне линий уровня целевой функции (и ограничения для п.4).

Глава 9. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Динамическое программирование - один из мощных методов решения задач математического программирования, разработанный американским математиком Р. Беллманом в конце 50-х годов. Он изначально ориентирован на оптимизацию так называемых многошаговых (многостадийных, многоэтапных) процессов и использование ЭВМ.

Концепция метода проистекает из следующего свойства оптимального решения. Пусть оптимальный путь из точки A в точку E проходит через точки B , C и D (рис. 9.1). Тогда любая часть этого пути является оптимальным путем. Действительно, если бы, например, оптимальный путь из C в E не проходил через D , то и полный путь A, B, C, D, E не был бы оптимальным (его можно было бы улучшить за счет изменения перехода из C в E), что противоречит исходной посылке.

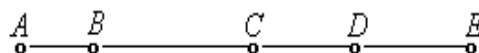


Рис. 9.1

Фундаментальной основой метода динамического программирования (ДП) является сформулированный Беллманом **принцип оптимальности**, согласно которому оптимальное управление определяется конечной целью управления и состоянием системы в рассматриваемый момент, независимо от того, каким образом она пришла в это состояние. Иначе говоря, при фиксированном состоянии системы последующее оптимальное решение не зависит от ее предыстории. Ввиду важности принципа приведем его полную формулировку.

Принцип оптимальности Беллмана. *Оптимальное поведение обладает тем свойством, что каковы бы ни были первоначальное состояние и решение в начальный момент, последующие решения должны составлять оптимальное поведение относительно состояния, получающегося в результате первого решения.*

Этот принцип является конструктивным, он позволяет существенно повысить эффективность решения определенного класса задач математического программирования путем их разложения на ряд задач значительно меньшей размерности. Как уже отмечалось выше, здесь имеются в виду задачи, которые могут быть представлены как многошаговые. Во многих случаях это свойство задачи очевидно, в других - требуется некоторое умение, чтобы выделить шаги. Такие задачи описываются математической моделью, в которой и критерий, и ограничения являются **составными**. Под составной понимается функция f , образованная частными функциями (подфункциями) f_i , к которым

применен один и тот же оператор вхождения (например, оператор сложения), т.е.

$$f = (\langle \text{оператор} \rangle \langle f_1, f_2, \dots, f_m \rangle).$$

При этом в одной модели могут быть составные функции с разными операторами. Частным случаем составной функции является *сепарабельная* функция. Метод предполагает, что подфункции зависят от значительно меньшего числа переменных, чем сама функция, и в каждом условии модели переменные частных функций не пересекаются. Количество шагов в задаче определяется числом подфункций критерия. Приведенное описание свойств модели не является строго математическим, но его усвоение облегчает оценку возможности применения метода ДП к конкретной задаче. Ниже мы будем использовать такую характеристику моделей рассматриваемых задач, что должно помочь лучшему пониманию изучаемого метода.

Очень важным понятием в ДП является ***состояние*** системы. Оно фигурирует в принципе оптимальности и на нем базируются вычислительные процедуры ДП. При первом знакомстве с методом ДП определение состояния вызывает, пожалуй, наибольшие трудности, но по мере решения задач они быстро уменьшаются.

При разбиении задачи на шаги состояние служит связующим звеном между смежными шагами. Состояние зависит от предшествующих решений, но зная его, можно принимать решения на последующих шагах без учета решений, предшествовавших данному состоянию. Обратим внимание на важный момент: чтобы принять очередное решение, достаточно знать состояние, а не решения, приведшие к этому состоянию. Так, например, при корректировке траектории некоторого летательного аппарата (в однородной среде) достаточно знать его координаты на момент принятия решения, и не нужны решения, приведшие его в данную точку пространства. Значит, координаты и определяют состояние системы. Если же выбор решения зависит также от количества топлива на борту аппарата, то состояние системы будет определяться как координатами, так и весом оставшегося топлива. Итак, состояние описывается теми переменными системы, которые зависят от решения на предшествующем шаге и знание которых достаточно для принятия решения на очередном шаге. Чтобы отличать переменные состояния от искомым переменных (управлений), в дальнейшем будем называть их *параметрами состояния*.

9.1. Как работает метод ДП

Для первоначального представления метода ДП не будем прибегать к математическим выкладкам и рассмотрим его сущность на описательном уровне. Представим себе лабиринт без петель, в котором между любыми

входами и выходами встречается одинаковое число развилок (узлов), например 3. Во всех узлах пересекается более одной дороги. Каждому переходу в лабиринте соответствует определенное время. Необходимо найти путь от одного из входов к любому выходу, на который требуется минимальное время.

Эту задачу можно решить методом полного перебора всех маршрутов, ведущих от фиксированного входа к выходу из лабиринта. При этом число вариантов может оказаться весьма большим, а изменение входа потребует повторного перебора. Принципиально другой подход дает метод ДП. Он предлагает конструировать оптимальный путь по частям, для чего нужно представить задачу как многошаговую. Разместим условно все входы на одной вертикальной прямой, все узлы, которые встречаются первыми на пути от входов к выходам, - на другой прямой. Также поступим со второй и другими группами узлов и, наконец, с выходами. Получим схему лабиринта в виде, показанном на рис.9.2.

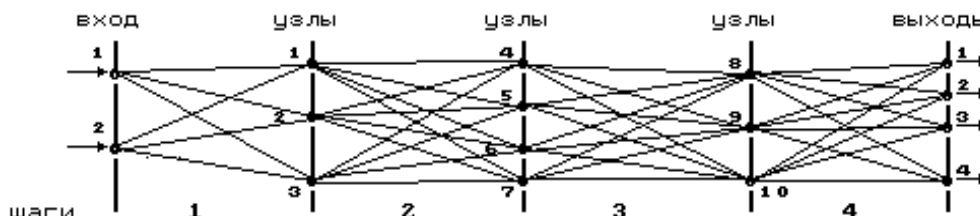


Рис. 9.2

В данном случае мы говорим, что задача представлена как 4-шаговая. Построение оптимального пути можно начинать с 1-го или 4-го шага, но предпочтительнее с 4-го. Такой порядок решения иногда называют **обратной прогонкой** в отличие от **прямой**, при которой идут с 1-го шага. Следует сразу заметить, что для некоторых задач возможен только один порядок прогонки, обусловливаемый спецификой задачи.

Итак, начнем с построения пути на 4-м шаге. Но мы тут же сталкиваемся с проблемой: как принимать решение, если не известен узел, в котором мы окажемся после оптимального прохождения первых трех этапов? Чтобы обойти эту проблему, будем искать оптимальное решение для каждого из узлов, в котором можем оказаться перед 4-м шагом (это узлы 8, 9 и 10). Фиксируем узел 8 и из четырех значений времени перехода из него к выходам выбираем наименьшее. Соответствующий переход может принадлежать оптимальному пути. Узлу 8 приписываем этот переход и найденное минимальное время, которое обозначим как t_8 . Аналогично поступаем, фиксируя узел 9, а затем 10. В результате получим t_9 и t_{10} соответственно и переходы, на которых достигаются эти минимальные значения времени. Тем самым завершается первый этап

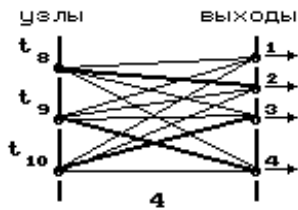


Рис. 9.3

построения оптимального пути. Результат представлен на рис.9.3, где жирными линиями выделены оптимальные переходы.

Теперь полагаем, что осталось совершить 3-й и 4-й переходы. Но как и ранее, мы не знаем узел, с которого будем двигаться. Это снова вынуждает нас искать решение не для одного, а для всех узлов, в которые можем прийти после первых двух шагов. Фиксируем узел 4 и определяем минимальный путь из него к выходам. При полном переборе пришлось бы сопоставлять 12 вариантов (столько путей связывает узел 4 с выходами). Однако нам достаточно сравнить только три: 1) время на переходе 4-8 плюс t_8 ; 2) время на переходе 4-9 плюс t_9 ; 3) время на переходе 4-10 плюс t_{10} . Минимальное значение приписываем узлу 4 (t_4) и выделяем жирной линией соответствующий переход на третьем шаге. Здесь уже проявилась принципиальная особенность ДП: оптимальный переход на шаге 3 определялся не как самый короткий среди переходов этого шага, а как такой, который обеспечивает минимум времени от данного узла к выходу. Это значит, что выбирая решение на 3-м шаге, мы учитывали его последствия. Точно так же находим решения для узлов 5, 6 и 7. Возможные результаты после второго этапа решения приведены на рис.9.4.

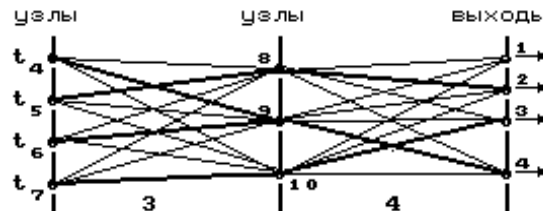


Рис. 9.4

Опираясь на них, можно переходить к третьему этапу построения оптимального пути, охватывающему 2, 3 и 4-й шаги. Рассуждения аналогичны вышеприведенным. Например, оптимальный переход на 2-м шаге из узла 1 определяется по минимуму из четырех значений:

$$\min \begin{bmatrix} (\text{время перехода } 1 - 4) + t_4 \\ (\text{время перехода } 1 - 5) + t_5 \\ (\text{время перехода } 1 - 6) + t_6 \\ (\text{время перехода } 1 - 7) + t_7 \end{bmatrix} = t_1.$$

Переход, которому соответствует t_1 , и будет наилучшим для узла 1. И снова отметим: выбор перехода на одном 2-м шаге проводился с учетом

влияния на последующие шаги вплоть до выхода! Нетрудно подсчитать, что при полном переборе путей из узла 1 к выходам пришлось бы сравнить 48 значений, каждое из которых вычисляется как сумма трех чисел. После завершения третьего этапа расчетов будут определены оптимальные пути из узлов 1-3 (рис.9.5).

Последний, четвертый, этап охватывает все шаги. Определение оптимального перехода на 1-м шаге для каждого из входов требует сравнения всего трех вариантов против 144 при полном переборе.

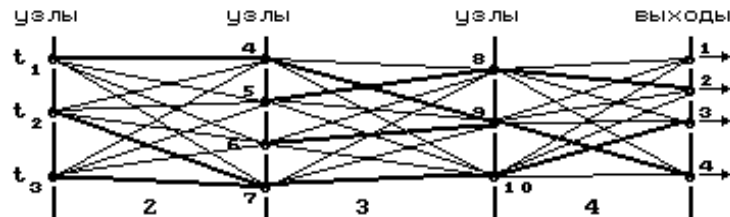


Рис. 9.5

Найдя решение для 1-го шага, мы тем самым завершаем построение оптимальных путей (рис.9.6). Непрерывная жирная линия, соединяющая заданный вход с выходом, и есть оптимальный путь, а соответствующее ему минимальное время равно $t_{\text{вх}}$. Двигаясь по нему от входа к выходу, то есть в **прямом** направлении (или, иначе говоря, в направлении, **обратном порядку расчета**), последовательно находим переходы, составляющие оптимальный путь.

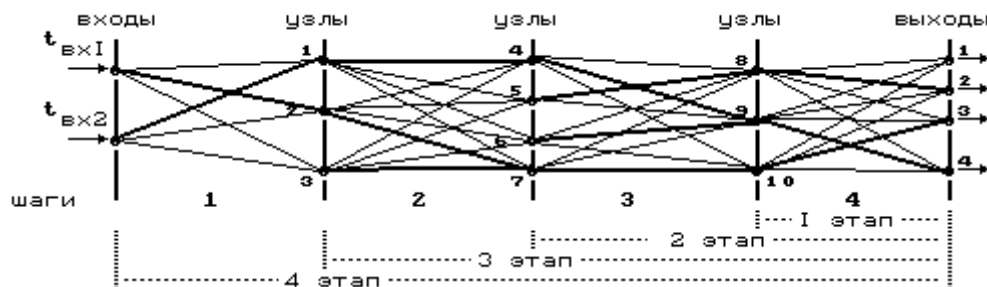


Рис. 9.6

Этот пример иллюстрирует основные черты метода ДП:

1. Вместо одной задачи решается 4 однотипных задачи меньшей размерности. Если каждому шагу поставить в соответствие одну переменную, а переходу - ее значение, то исходная задача - это задача оптимизации с четырьмя переменными. В рассмотренной процедуре расчета она заменяется последовательностью четырех задач с одной переменной, что существенно облегчает решение.

2. Решение каждой из этих задач опирается на результаты решения предшествующей (в смысле порядка расчета) задачи. Связь задач осуществляется через *состояние*, которое есть, с одной стороны, результат выбора конкретного перехода на данном шаге, с другой - исходная информация для извлечения результата решения непосредственно предшествующей задачи. Из такого смысла состояния следует, что каждая задача должна решаться для *всех состояний*, с которых может начаться рассматриваемый шаг. В нашем примере состояние полностью описывается номером узла или входа, и на каждом этапе мы искали решение для всех возможных узлов, так как иначе нельзя было бы приступить к следующему этапу расчета. Такую схему расчета называют *условной оптимизацией*. Это название подчеркивает, что на каждом шаге при выборе оптимального решения оговаривается условие, а именно, состояние, относительно которого ищется решение.

3. При заданном исходном состоянии, то есть для известного входа, выделение оптимальных переходов на каждом шаге не требует оговаривать условия: заданный вход определяет оптимальный переход на 1-м шаге, его результатом будет состояние накануне 2-го шага, по которому определяется оптимальный переход на 2-м шаге, приводящий в состояние, из которого совершается оптимальный переход на 3-м шаге и так до выхода. Определение оптимального решения для заданного исходного состояния по результатам условной оптимизации, осуществляемое в направлении, обратном условной оптимизации, в динамическом программировании называют *безусловной оптимизацией*.

4. Как видно из рис.9.6, значительная часть выделенных жирной линией переходов не входит в оптимальные пути, что на первый взгляд кажется существенным недостатком метода, так как соответствующие расчеты сделаны впустую. Однако более глубокий анализ показывает, что этот недостаток метода оборачивается его достоинством. Действительно, при реализации оптимальных решений нередко отклонения от первоначальных условий, что приводит систему в состояние, отличное от предусмотренного найденным решением. В этом случае при любом другом методе пришлось бы решать задачу заново, в то время как в результатах условной оптимизации ДП уже содержатся оптимальные решения для *всех возможных состояний* и поэтому достаточно выполнить лишь безусловную оптимизацию для нового состояния.

Уяснение рассмотренного примера и выводов по нему очень важно для понимания всего последующего материала.

9.2. Функциональное уравнение ДП

Теперь от описательного представления метода перейдем к формализованному. Рассмотрим достаточно общую многошаговую задачу,

к которой применим метод ДП. Чтобы не путаться с различной нумерацией шагов и этапов расчета, пронумеруем шаги в порядке проведения условной оптимизации, в данной задаче с конца к началу. Эффективность i -го шага описывается функцией $Z_i(S_i, U_i)$, где S_i - состояние к i -му шагу (точнее - вектор параметров состояния), U_i - управление на i -м шаге (точнее - вектор управляемых переменных или решение). Тогда структуру задачи можно представить так, как показано на рис.9.7.

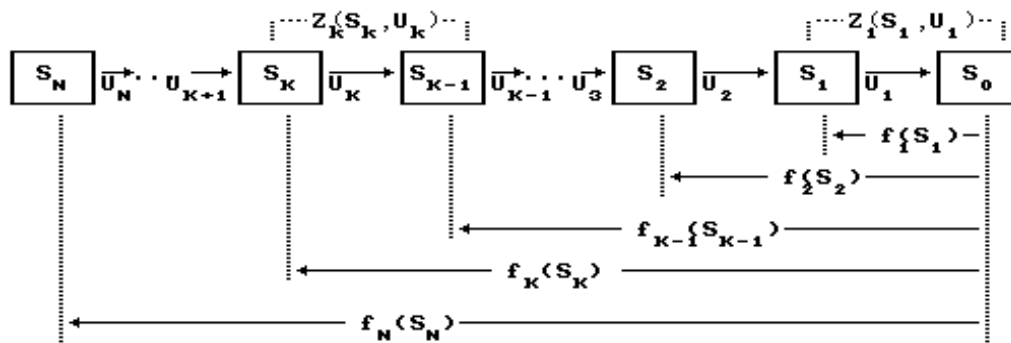


Рис. 9.7

Модель задачи ДП включает целевую функцию

$$Z = \sum_{i=1}^N Z_i(S_i, U_i) \Rightarrow \text{extr}, \quad (9.1)$$

описание допустимой области управлений D , а также уравнение состояния $S_{i-1} = \psi(S_i, U_i)$, связывающее между собой два последовательных состояния. Это уравнение легко выводится из условий задачи.

Формализация вычислительной процедуры метода ДП базируется на принципе оптимальности. Напомним, что суть его состоит в том, что последующие решения должны быть оптимальными относительно состояния, сложившегося в результате предшествующих, пусть и не оптимальных, решений. Для описания этого свойства введем последовательность функций $\{f_k(S_k)\}$, $k = \overline{1, N}$ так, что каждая из них есть зависимость экстремального значения критерия за k оставшихся шагов от состояния на начало k -го шага:

$$f_k(S_k) = \text{extr}_{\{U_i\}} \sum_{i=1}^k z_i(S_i, U_i). \quad (9.2)$$

Какие шаги охватывает каждая функция введенной последовательности, видно также на рис.9.7. Если бы мы использовали прямую нумерацию шагов, то в формуле (9.2) суммирование должно быть от k до N , и функция f_k охватывала бы $N-k$ шагов, что, на наш взгляд, менее предпочтительно при интерпретации ее смысла. Важно уяснить, что функции (9.2) зависят только от состояния и не могут зависеть от искомых переменных (управлений), так как по ним ищется экстремум. Отсюда

также следует еще один способ определения состояния: состояние - это то, от чего зависит *экстремум* критерия.

Теперь построим рассуждения на основе принципа оптимальности. Предположим, что осталось k шагов ($k \geq 2$), на которых предстоит принять решение, и S_k - состояние, сложившееся к k -му шагу (см. рис.9.7). Выделим один k -й шаг и из *допустимых управлений* на этом шаге возьмем произвольное U_k . Тогда эффективность шага составит $Z_k(S_k, U_k)$, а состояние, в которое придем в результате такого выбора, будет S_{k-1} . Согласно принципу оптимальности не имеет значения, как мы попали в это состояние - последующие решения, то есть решения на оставшихся $(k-1)$ шагах, должны быть оптимальны относительно этого состояния. Но при оптимальных решениях на $(k-1)$ шагах будет достигаться экстремальное значение критерия за все эти шаги, которое по определению (9.2) равно $f_{k-1}(S_{k-1})$. Следовательно, эффективность k шагов составит

$$z_k(S_k, U_k) + f_{k-1}(S_{k-1}). \quad (9.3)$$

Очевидно, что она не является экстремальной для k шагов, так как U_k взято произвольно. Если воспользоваться уравнением состояния, то (9.3) примет вид

$$z_k(S_k, U_k) + f_{k-1}(\psi(S_k, U_k)), \quad (9.4)$$

из чего следует, что при фиксированном состоянии S_k эффективность k шагов зависит только от управления на одном k -м шаге (здесь полезно найти аналогию с задачей о лабиринте). В этом главный смысл метода ДП. Значит, варьируя U_k в допустимой области, можно найти экстремум выражения (9.4). Но экстремальное значение за k шагов по определению (9.2) есть $f_k(S_k)$. Таким образом, окончательно получаем

$$f_k(S_k) = \text{extr}_{U_k \in D_k} [z_k(S_k, U_k) + f_{k-1}(\psi(S_k, U_k))]. \quad (9.5)$$

Выражение (9.5) называется *основным функциональным уравнением* динамического программирования. Чтобы подчеркнуть специфическую структуру этого выражения, его также называют *рекуррентным соотношением*. Действительно, согласно (9.5) одна функция последовательности вычисляется через другую, непосредственно ей предшествующую. Поэтому вычисления f_k возможны только после того, как будут известны значения f_{k-1} для всех возможных состояний S_{k-1} . Чтобы начать рекуррентные вычисления, необходимо иметь первую функцию последовательности. Она находится непосредственно по определению (9.2) для $k=1$:

$$f_1(S_1) = \text{extr}_{U_1 \in D_1} z_1(S_1, U_1). \quad (9.6)$$

После вычисления f_1 , используя (9.5), последовательно находим f_2, f_3, \dots, f_N . При этом каждая функция f_k ($k = \overline{1, N}$) вычисляется для всех возможных состояний S_k . Результаты вычислений представляются в таблицах одинаковой структуры, включающей состояние, условно оптимальные значения переменных (компоненты вектора U_k^*) и значение функции. Число строк в таблице равно числу состояний, возможных к данному шагу. Следует заметить, что в процессе расчета f_k допустимое множество D_k изменяется, так как оно зависит от состояния S_k , на что будет обращено внимание в следующих задачах.

Теперь по заданному состоянию S_N^{\wedge} входим в N -ю таблицу и из нее извлекаем U_N^* и $f_N(S_N^{\wedge})$. Зная S_N^{\wedge} и U_N^* , по уравнению состояния находим S_{N-1}^{\wedge} и по нему входим в $(N-1)$ -ю таблицу, откуда берем U_{N-1}^* и $f(S_{N-1}^{\wedge})$. Этот прямой ход, соответствующий безусловной оптимизации, продолжаем аналогичным образом вплоть до 1-й таблицы, из которой найдем U_1^* и $f_1(S_1^{\wedge})$. Очевидно, что $f_N(S_N^{\wedge})$ - это и есть оптимум нашей задачи, а соответствующие $U_1^*, U_2^*, \dots, U_N^*$ - ее оптимальное решение. Таким образом, метод ДП позволил заменить задачу с N векторами переменных N задачами с одним искомым вектором каждая, что существенно облегчило решение.

Исходя из всего вышерассмотренного, можно представить процедуру динамического программирования в виде следующей последовательности.

1. Имея описание и модель задачи, выделяем шаги и производим их нумерацию с конца.

2. Определяем параметры состояния и вводим последовательность функций $\{f_k(S_k)\}$, $k = \overline{1, N}$, в которой каждая функция $f_k(S_k)$ есть **наилучшее значение критерия за k оставшихся шагов относительно состояния S_k** .

3. На основе принципа оптимальности составляем функциональное уравнение ДП и отдельно выражение для f_1 .

9. Проводим условную оптимизацию, последовательно вычисляя f_1, f_2, \dots, f_N . При этом на каждом шаге для всех возможных значений состояния S_k запоминаются значения U_k^* и f_k (в таблице или файле).

5. Исходя из заданного состояния S_N^{\wedge} , проводим безусловную оптимизацию по схеме:

$$S_N^{\wedge} \rightarrow \text{табл. } N \rightarrow U_N^* \rightarrow \text{у.с.} \rightarrow S_{N-1}^{\wedge} \rightarrow \text{табл. } N-1 \rightarrow U_{N-1}^* \rightarrow \text{у.с.} \rightarrow \dots \rightarrow S_1^{\wedge} \rightarrow \text{табл. } 1 \rightarrow U_1^*,$$

где у.с. - уравнение состояния. Значение $f_N(S_N)$ из N -й таблицы есть оптимальное значение критерия задачи.

Отметим, что при прямой нумерации шагов рекуррентная формула (9.5) связывала бы f_k с f_{k+1} , а в процедуре ДП следовало бы заменить нумерацию элементов и таблиц на обратную.

В следующих разделах процедура ДП будет применяться к конкретным задачам исследования операций.

9.3. Распределение одного вида ресурса

Пусть необходимо распределить ресурс в количестве X между N предприятиями, если известно, что при выделении i -му предприятию ресурса x_i оно дает прибыль $r_i(x_i)$. При этом к функциям r_i не предъявляется каких-либо требований (например, дифференцируемости).

Запишем модель задачи:

$$R(x_1, x_2, \dots, x_n) = \sum_{i=1}^N r_i(x_i) \rightarrow \max, \quad \sum_{i=1}^N x_i = X, \quad \forall x_i \geq 0. \quad (9.7)$$

Очевидно, что условия применимости ДП выполняются: и критерий, и ограничение задачи (9.7) являются составными (в данном случае - аддитивными). Представим задачу как многошаговую. В качестве шага будем рассматривать выделение ресурса одному предприятию. Расположим предприятия в виде последовательности, в которой каждому предприятию приписано определенное место, и пронумеруем их справа налево. Порядок предприятий в последовательности шагов принципиального значения не имеет, но могут оказаться полезными следующие советы:

1. Если условие-ограничение имеет вид равенства, то на 1-е место целесообразно ставить предприятие с наиболее сложной функцией прибыли, а при неравенстве - наоборот, с самой простой, так как в этом случае несколько снизится трудоемкость расчетов.

2. Если нет уверенности, что некоторое предприятие останется в рассматриваемой системе распределения, его следует поставить последним (на самое левое место); тогда в случае его удаления не придется решать задачу заново.

Теперь необходимо определиться с состоянием. Для этого зададимся вопросом: что нужно знать, чтобы принять решение по распределению ресурса между заданным числом предприятий? Ответ очевиден - количество распределяемого ресурса. Тот же ответ последует и на вопрос: от чего зависит **максимальная** прибыль системы предприятий в нашей задаче? Следовательно, состояние определяется одним параметром - количеством распределяемого ресурса, которое обозначим через V . Оно может относиться к разному числу предприятий, поэтому $V \leq X$. В соответствии с процедурой ДП введем последовательность функций $\{f_k(V_k)\}$ так, что

$$f_k(V_k) = \max \sum_{i=1}^k r_i(x_i), \quad k = \overline{1, N} \quad (9.8)$$

есть максимальная прибыль k предприятий при распределении между ними ресурса в количестве V_k .

Если бы пришлось искать функции (9.8) непосредственно, то потребовалось бы решать семейство задач вида

$$R^k(x_1, x_2, \dots, x_k) = \sum_{i=1}^k r_i(x_i) \rightarrow \max, \quad \sum_{i=1}^k x_i = V, \quad \forall x_i \geq 0 \quad (9.9)$$

для $0 \leq V \leq X$, что многократно увеличивает трудоемкость решения. В динамическом программировании число задач будет таким же, однако их размерность не зависит от номера задачи и будет в N раз меньше размерности исходной задачи.

Перейдем к составлению рекуррентного соотношения. Допустим, что осталось распределить ресурс V_k между k предприятиями. Выделим из них две части: k -е предприятие и все остальные $k-1$ (рис.9.8).

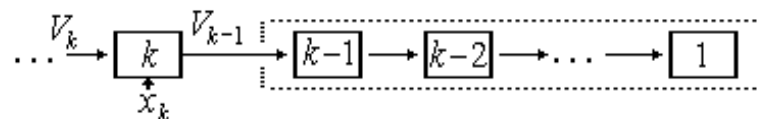


Рис. 9.8

Примем решение по k -му предприятию: из всего имеющегося ресурса V_k выделим ему x_k . Тогда оно даст прибыль $r_k(x_k)$, а система перейдет в состояние V_{k-1} , относительно которого следует искать решения на $k-1$ оставшихся шагах. Согласно принципу оптимальности эти решения должны составлять оптимальное поведение независимо от того, как система попала в состояние V_{k-1} . Такое поведение обеспечит максимальную прибыль от $k-1$ предприятий, то есть $f_{k-1}(V_{k-1})$. В результате, прибыль k предприятий составит

$$r_k(x_k) + f_{k-1}(V_{k-1}). \quad (9.10)$$

Как следует из модели задачи (9.9), уравнение состояния имеет простой вид

$$V_{k-1} = V_k - x_k, \quad (9.11)$$

что означает уменьшение распределяемого ресурса после выделения x_k ресурса k -му предприятию. Подставив (9.11) в (9.10), получим зависимость только от одной переменной x_k :

$$r_k(x_k) + f_k(V_k - x_k),$$

максимизация которой по x_k в допустимой области обеспечивает максимальную прибыль от всех k предприятий при распределении ресурса в количестве V_k . Но по определению (9.8) это есть $f_k(V_k)$. Таким образом, приходим к искомому рекуррентному соотношению

$$f_k(V_k) = \max_{0 \leq x_k \leq V_k} [r_k(x_k) + f_{k-1}(V_k - x_k)]. \quad (9.12)$$

Здесь видно, как допустимая область D_k , задаваемая диапазоном x_k , явно зависит от состояния V_k . Формула (9.12) применима для $k \geq 2$. Для первой функции последовательности в соответствии с ограничением модели (9.7) и определением (9.8) имеем

$$f_1(V_1) = r_1(V_1), \quad x_1^* = V_1,$$

так как весь имеющийся ресурс распределяется полностью. Поэтому 1-й шаг условной оптимизации состоит в прямом вычислении значений заданной функции r_1 , а в ряде случаев и этого не требуется.

Остановимся на вычислительной стороне формулы (9.12). В отдельных случаях при простых функциях $r_k(V_k)$ максимум в (9.12) может быть найден аналитически, однако в общем случае максимум находится одним из численных методов, в том числе полным перебором допустимых значений x_k . Тогда, если x_k и V_k - непрерывные величины, для вычислений по рекуррентному соотношению их необходимо дискретизировать. Шаг дискретности Δ выбирается по точности, с которой следует провести распределение ресурса. При этом, исходя из уравнения состояния, в данной задаче шаг дискретности должен быть одинаковым для x_k и V_k . С учетом этого рекуррентная формула (9.12) примет вид

$$f_k(v\Delta) = \max_{0 \leq \mu \leq v} [r_k(\mu\Delta) + f_{k-1}((v - \mu)\Delta)], \quad (9.13)$$

где $v = 1, 2, \dots, m$; $m\Delta = X$, $v\Delta = V_k$, $\mu\Delta = x_k$. Один из способов вычислений по формуле (9.13) состоит в том, что сначала фиксируется v , а затем ищется максимум правой части, и так для всех значений v . Результаты заносятся в таблицу вида

V_k	x_k^*	$f_k(V_k)$
Δ	$x_k^*(\Delta)$	$f_k(\Delta)$
2Δ	$x_k^*(2\Delta)$	$f_k(2\Delta)$
\vdots	\vdots	\vdots
$v\Delta$	$x_k^*(v\Delta)$	$f_k(v\Delta)$
\vdots	\vdots	\vdots
$m\Delta$	$x_k^*(m\Delta)$	$f_k(m\Delta)$

где $x_k^*(v\Delta)$ - значение x_k , доставляющее максимум в (9.13) при $V_k = v\Delta$. Все N таблиц, получаемых при последовательном расчете функций f_k , имеют одинаковую структуру. Решение исходной задачи (9.7) находится на этапе безусловной оптимизации следующим образом. По значению $V_N = X$ входим в таблицу N , из которой берем $f_N(X)$ - максимальную прибыль от N предприятий при распределении между ними ресурса в количестве X и $x_N^* = x_N^*(X)$ - количество ресурса, выделяемое N -му предприятию при оптимальном распределении ресурса X . По уравнению состояния (9.11)

находим $V_{N-1}=X-x_N^*$, входим по этому значению состояния в таблицу $N-1$ и из нее извлекаем $f_{N-1}(X-x_N^*)$ и x_{N-1}^* - максимальную прибыль от $N-1$ предприятий и количество ресурса, выделяемое $(N-1)$ -му предприятию при оптимальном распределении, соответственно. Снова пересчитываем состояние

$$V_{N-2}=V_{N-1}-x_{N-1}^*=X-x_N^*-x_{N-1}^*$$

и по нему входим в таблицу $N-2$, в которой найдем x_{N-2}^* , и так вплоть до таблицы 1, из которой получим x_1^* . В результате имеем оптимальное решение задачи распределения ресурса в количестве X .

На примере этой задачи отметим ряд достоинств метода ДП.

1. Задача содержит N переменных, которые после дискретизации могут принимать m значений. Поэтому порядок числа вариантов распределения определяется величиной m^N (точное значение можно получить, если убрать варианты, не удовлетворяющие условию в (9.7)). При расчете по рекуррентной формуле максимум ищется для m значений состояния, а поиск максимума путем просмотра всего диапазона переменной требует перебора от двух до $m+1$ вариантов, то есть в среднем $\sim m/2$. Значит, один шаг включает расчет $m^2/2$ вариантов, а вся задача в ДП - $Nm^2/2$. Очевидно, что $Nm^2/2 \ll m^N$. Кроме того, вариант в полном переборе требует выполнения большего числа операций (вычисления N функций $r(V)$ и $N-1$ сложений). Так, например, при $m=N=10$ полный перебор требует расчета $\sim 10^{10}$ вариантов, что даже при затратах машинного времени порядка 10^{-5} секунды на вариант потребует непрерывной работы ЭВМ в течение 277 часов. Для решения этой же задачи методом ДП достаточно рассчитать не более 500 вариантов (!).

2. В случае изменения количества распределяемого ресурса метод полного перебора и другие регулярные численные методы требуют решать задачу заново. В динамическом программировании, если новое значение не больше X , достаточно провести безусловную оптимизацию, взяв за исходное состояние новое значение ресурса. А, как показано выше, безусловная оптимизация требует минимума элементарных действий. Это же возможно и при увеличении ресурса, если с самого начала решения задачи методом ДП увеличить значение X до максимально ожидаемого уровня либо дополнить часть последних таблиц недостающими строками.

3. При использовании любого метода оптимизации, кроме ДП, исключение одного из предприятий из системы распределения приведет к необходимости решать изменившуюся задачу как новую. В динамическом программировании, если этому предприятию был присвоен номер N , то есть оно было последним по ходу условной оптимизации, решение измененной задачи находится сразу: по заданному количеству распределяемого ресурса входим не в N -ю, а в $(N-1)$ -ю таблицу и далее

действуем в соответствии со схемой безусловной оптимизации, которая была рассмотрена выше.

Свойства, приведенные в п.п.2,3, обусловлены тем, что несмотря на резкое сокращение числа рассчитываемых вариантов, метод ДП находит не один оптимум для заданных X и N , а оптимумы для всех возможных состояний в данной задаче для принятого порядка нумерации предприятий и соответствующие им оптимальные решения. Иначе говоря, метод ДП дает весь ансамбль оптимальных решений, свойственных задаче. Отсюда также следует, что получаемые результаты позволяют определить чувствительность критерия к изменениям X .

4. Метод ДП не накладывает каких-либо специальных требований на вид и форму представления функций, составляющих критерий. Так, в рассмотренной задаче функции $r(V)$ могут быть недифференцируемыми, могут быть заданы даже в виде таблицы или графика, или задаваться алгоритмически.

5. Отыскание глобального экстремума многомерной функции является очень сложной проблемой. Динамическое программирование снижает размерность решаемых задач в N раз (по числу переменных), что значительно облегчает нахождение глобального экстремума. Так в рассмотренной задаче на каждом шаге находится максимум функции одной переменной. Из свойства рекуррентного соотношения следует, что отыскание глобального оптимума на каждом шаге гарантирует достижение глобального оптимума исходной задачи.

6. Наложение специфических условий на переменные не «утяжеляет» решение задачи методом ДП (в отличие от многих других). Например, ограничения сверху и снизу или/и дискретность (в частности, целочисленность) переменных не затрудняют, а, наоборот, облегчают решение на каждом шаге.

7. Трудоемкость вычислений и объем памяти для хранения результатов на одном шаге не зависят от числа шагов. Поэтому увеличение числа шагов приводит только к пропорциональному возрастанию времени решения задачи, тогда как для других методов с увеличением размерности задачи трудоемкость ее решения растет гораздо быстрее.

Для лучшего усвоения материала решение следующих двух задач поясняется на численных примерах.

9.4. Задача организации выпуска m видов продукции

Фирма ежемесячно выпускает m видов продукции, на которые имеется равномерный спрос. По каждому виду продукции известны:

C_{1i} - затраты на хранение единицы продукции i -го вида в единицу времени;

C_{2i} - затраты на запуск в производство одной партии i -го вида;

R_i - месячный спрос на продукцию i -го вида.

Для выпуска одной партии требуется один цикл производства. Изменение числа партий приводит к изменению затрат на производство одного и того же количества продукции. В течение месяца фирма может организовать не более N циклов. В этих условиях нужно так организовать производство, чтобы при полном удовлетворении спроса обеспечить минимальные затраты фирмы.

При построении модели примем одно допущение, обусловленное отсутствием необходимых данных и не имеющее принципиального значения: время выпуска партии много меньше продолжительности цикла и поэтому им можно пренебречь. График изменения уровня готовой продукции на фирме представлен на рис.9.9,

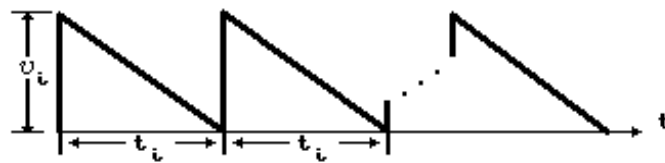


Рис. 9.9

где v_i - объем партии продукции i -го вида; t_i - продолжительность цикла по i -му виду продукции (в долях месяца).

Критерий, суммарные затраты на все виды продукции, определим следующим образом. Затраты на хранение и выпуск партии i -го вида продукции за время одного цикла составят

$$\frac{v_i}{2} C_{1i} t_i + C_{2i},$$

а так как количество циклов, необходимое для выпуска продукции в объеме R_i

$$u_i = \frac{1}{t_i} = \frac{R_i}{v_i},$$

то затраты на весь объем продукции i -го вида запишутся в виде

$$\left(\frac{v_i}{2} C_{1i} t_i + C_{2i} \right) \frac{1}{t_i} = \frac{C_{1i} R_i t_i}{2} + \frac{C_{2i}}{t_i} = \frac{C_{1i} R_i}{2u_i} + C_{2i} u_i = Q_i(u_i).$$

Суммируя затраты по всем видам продукции, получим критерий задачи

$$Q(U) = \sum_{i=1}^m \left(\frac{C_{1i} R_i}{2u_i} + C_{2i} u_i \right) \rightarrow \min_{\{u_i\}}. \quad (9.14)$$

Добавив к выражению критерия очевидные ограничения

$$\sum_{i=1}^m u_i \leq N, \quad \forall u_i \geq 0, \text{ int(целые)}, \quad (9.15)$$

завершаем построение модели задачи.

По структуре эта модель аналогична рассмотренной в предыдущем разделе, так что применимость метода ДП к задаче (9.14),(9.15) не требует дополнительных доказательств. В качестве параметра состояния здесь следует взять n - максимальное количество циклов, которое может быть организовано для выпуска продукции ($1 \leq n \leq N$), а за шаг примем организацию выпуска одного вида продукции. Построив из видов продукции последовательность, порядок следования в которой определяется из соображений, приведенных в предыдущем разделе, приходим к m -шаговой задаче.

Введем последовательность функций $\{f_k(n)\}$, $k = \overline{1, m}$ так, что каждая функция определяется как минимальные затраты на выпуск k видов продукции в заданных количествах при условии, что можно организовать до n циклов, то есть

$$f_k(n) = \min \sum_{i=1}^k \left(\frac{C_{1i} R_i}{2u_i} + C_{2i} u_i \right). \quad (9.16)$$

Здесь опущен индекс у параметра состояния n , так как он совпадает с индексом функции и поэтому не обязателен. Как и в предыдущей задаче, выделив из k видов ($k > 1$) k -й и все остальные $k-1$ видов продукции и применив принцип оптимальности, получим затраты на k видов

$$Q_k(u_k) + f_{k-1}(n - u_k).$$

В этом выражении новое состояние определяется через исходное n и число циклов u_k на выпуск k -го вида продукции: на $k-1$ видов остается $n - u_k$ циклов (имеем очевидное уравнение состояния $n_{k-1} = n_k - u_k$).

Так как для фиксированного n затраты зависят только от u_k , то минимизируя их по u_k в области допустимых значений, приходим к рекуррентному соотношению

$$f_k(n) = \min_{0 \leq u_k \leq n-k+1} [Q_k(u_k) + f_{k-1}(n - u_k)], \quad k \geq 2. \quad (9.17)$$

Почему же в этой формуле такое ограничение сверху на u_k ? Дело в том, что выпуск продукции оговорен величиной R_i , поэтому, чтобы обеспечить выпуск $k-1$ видов продукции, нужно иметь хотя бы по одному циклу на каждый вид i , следовательно, максимальное число циклов на k -й вид равно $n - (k-1)$.

Первая функция вычисляется непосредственно по определению (9.16):

$$f_1(n) = \min_{0 \leq u_1 \leq n} \left(\frac{C_{11} R_1}{2u_1} + C_{21} u_1 \right). \quad (9.18)$$

Для численного примера ограничимся тремя видами продукции и $N=7$. Остальные данные следующие:

$i(k)$	1	2	3
C_{1i}	1	2	1
C_{2i}	10	8	5
R_i	320	200	490

Условная оптимизация начинается с вычисления функции f_1 согласно формуле (9.18). При этом можно воспользоваться классическим методом, так как требуется найти минимум дифференцируемой функции на интервале. Приравняв первую производную выражения в скобках правой части (9.18) нулю, найдем стационарную точку

$$u_1^0 = \sqrt{C_{11}R_1 / 2C_{21}}.$$

С учетом ограничений на u_1 оптимальное решение будет иметь вид

$$u_1^* = \begin{cases} [u_1^0] & \text{при } n \geq u_1^0, \\ n & \text{при } n < u_1^0, \end{cases} \quad (9.19)$$

где $[u_1^0]$ - целая часть u_1^0 . По исходным данным получаем $u_1^0=9$. Подставив в (9.18) исходные данные и оптимальное u_1 , приходим к расчетной формуле

$$f_1(n) = \frac{160}{u_1^*} + 10u_1^*,$$

по которой с учетом (9.19) составляем табл. 9.1.

↓

Таблица 9. 1

n	1	2	3	4	5	6	7
n_1^*	1	2	3	4	4	4	4
$f_1(n)$	170	100	83,3	80	80	80	80

На втором шаге расчет ведем по рекуррентной формуле (9.17):

$$f_2(n) = \min_{0 \leq u_2 \leq n-1} [Q_2(u_2) + f_1(n - u_2)], \quad (9.20)$$

где

$$Q_2(U_2) = \frac{C_{12}R_2}{2u_2} + C_{22}u_2 = \frac{200}{u_2} + 8u_2.$$

Находить минимум в (9.20) будем перебором допустимых u_2 . Для удобства ручных расчетов в промежуточную таблицу (табл. 9.2п) предварительно внесем во второй столбец значения $Q_2(u_2)$ для всех возможных u_2 и во вторую строку значения $f_1(n)$, взятые из табл. 9.1. Для отыскания минимума сначала фиксируем состояние, то есть значение n , а затем ведем перебор u_2 . Начнем с $n=1$. Расчет выпуска двух видов продукции в этом случае теряет смысл, так как на 2-й вид не остается ни одного цикла - во всех клетках столбца с этим значением n ставим прочерк. Берем $n=2$. При

этом возможен только один вариант: на 2-й вид продукции отводится один цикл. Соответствующие затраты составят

$$Q_2(1) + f_1(2-1) = 208 + 170 = 378,$$

они записываются в клетку на пересечении строки с $u_2=1$ и столбца с $n=2$. Очевидно, что эти значения затрат и числа циклов являются оптимальными относительно состояния $n=2$ и поэтому их записываем в клетки нижних строк как значения u_2^* и $f_2(n)$. Теперь фиксируем $n=3$ и вычисляем затраты для двух допустимых значений u_2 :

$$\text{при } u_2=1 \quad Q_2(1) + f_1(3-1) = 208 + 100 = 308,$$

$$\text{при } u_2=2 \quad Q_2(2) + f_1(3-2) = 116 + 170 = 286.$$

Минимальными являются затраты при $u_2=2$, поэтому $f_2(3)=286$ и $u_2^*=2$, что и записываем в соответствующие клетки нижних строк.

В табл. 9.2п обнаруживается свойство, обусловленное структурой формулы (9.20), которое позволяет вести расчет по простой схеме прямо в таблице, не обращаясь каждый раз к формуле (9.20): для подсчета затрат в текущей клетке нужно складывать значение из 2-го столбца данной строки со значением, взятым из строки f_1 в клетке, в которую попадаем, двигаясь по диагонали от текущей клетки.

Таблица 9.2п

U_2	$Q_2(u_2)$	n						
		1	2	3	4	5	6	7
		$f_1(n)$						
		170	100	83,3	80	80	80	80
1	208	-	208 <u>170</u> 378	208 <u>100</u> 308	208 <u>83,3</u> 291,3	208 <u>80</u> 288	208 <u>80</u> 288	208 <u>80</u> 288
2	116		-	116 <u>170</u> 286	116 <u>100</u> 216	116 <u>83,3</u> 199,3	116 <u>80</u> 216	116 <u>80</u> 216
3	90,7			-	90,7 <u>170</u> 260,7	90,7 <u>100</u> 190,7	90,7 <u>83,3</u> 174,0	90,7 <u>80</u> 170,7
4	82				-	82 <u>170</u> 252	82 <u>100</u> 182	82 <u>83,3</u> 165,3
5	80					-	80 <u>170</u> 250	80 <u>100</u> 180
6	81,3						-	81,3 <u>170</u> 251,3
u_2^*		-	1	2	2	3	3	4
$f_2(n)$		-	378	286	216	190,7	174	165,3

Аналогично проводим расчет для n от 4 до 7 включительно, занося получающиеся значения в табл. 9.2п. Этим завершается расчет на втором шаге. Результаты сохраняются в виде табл. 9.2.

↓

Таблица 9.2

n	1	2	3	4	5	6	7
u_2^*	-	1	2	2	3	3	4
$f_2(n)$	-	378	286	216	190,7	174	165,3

Теперь переходим к третьему шагу расчета, которому соответствует рекуррентная формула

$$f_3(n) = \min_{0 \leq u_3 \leq n-2} [Q_3(u_3) + f_2(n - u_3)], \quad (9.21)$$

где

$$Q_3(U_3) = \frac{C_{13}R_3}{2u_3} + C_{23}u_3 = \frac{245}{u_3} + 5u_3.$$

Как и на предыдущем шаге, вычисления выполним в промежуточной таблице (табл. 9.3п), в которую переносятся значения f_2 из табл. 9.2 и предварительно рассчитанные значения $Q_3(u_3)$. Схема расчета полностью идентична 2-му шагу, но расчет по трем видам продукции приобретает смысл начиная с $n=3$.

Таблица 9.3п

U_3	$Q_3(u_3)$	n					
		2	3	4	5	6	7
		$f_2(n)$					
		378	286	216	190,7	174	165,3
1	250	-	250	250	250	250	250
			<u>378</u>	<u>286</u>	<u>216</u>	<u>190,7</u>	<u>174</u>
			628	536	466	440,7	424
2	132,5		-	132,5	132,5	132,5	132,5
				<u>378</u>	<u>286</u>	<u>216</u>	<u>190,7</u>
				510,5	418,5	348,5	323,2
3	96,7			-	96,7	96,7	96,7
					<u>378</u>	<u>286</u>	<u>216</u>
					474,7	382,7	312,7
4	81,2				-	81,2	81,2
						<u>378</u>	<u>286</u>
						459,2	367,2
5	74					-	74
							<u>378</u>
							452
u_3^*		-	1	2	2	2	3
$f_3(n)$		-	628	510,5	418,5	348,5	312,7

Как и выше, можно пользоваться формальным правилом подсчета затрат без обращения к рекуррентной формуле. Из табл. 9.3п получаем табл. 9.3, которую необходимо сохранить. На этом заканчивается расчет 3-го шага и весь этап условной оптимизации.

⇓ Таблица 9.3

n	1	2	3	4	5	6
u_3^*	1	2	2	2	2	3
$f_3(n)$	-	628	510,5	418,5	348,5	312,7

Далее следует безусловная оптимизация. Пусть для выпуска трех видов продукции можно организовать 5 циклов ($N=5$). Тогда по начальному состоянию $n=5$ входим в табл. 9.3 (показано стрелкой). Из нее устанавливаем, что минимальные затраты на три вида продукции составляют 418,5, при этом по 3-му виду должно быть организовано 2 цикла, то есть весь заказ выполняется двумя запусками. Согласно уравнению состояния на остальные два вида остается 3 цикла. Поэтому по состоянию $n=3$ входим в табл. 9.2, как указывает стрелка, и извлекаем из нее $u_2^*=2$, то есть весь заказ тоже делится на 2 партии. Соответствующая величина $f_2(3)=286$ означает затраты на 1-й и 2-й виды продукции при оптимальной организации производства. Новое состояние $n=3-2=1$ определяет вход в табл. 9.1, из которой берем $u_1^*=1$, и тем самым завершаем нахождение оптимального решения.

Действуя аналогично, нетрудно убедиться, что при $N=7$ оптимальное решение будет таким: минимальные затраты, равные 312,7, достигаются при $u_1^*=2$, $u_2^*=2$, $u_3^*=3$. Теперь ясно, что мы можем получить оптимальное решение для любых n в диапазоне от 3 до 7. Если же 3-й вид продукции будет исключен, то нам не потребуется делать полный пересчет: достаточно с заданным значением $n=N$ войти в табл. 9.2, а затем по пересчитанному состоянию в табл. 9.1, чтобы получить соответствующее оптимальное решение. Например, для $N=7$ будем иметь $u_2^*=4$, $u_1^*=3$, а минимальные затраты составят 165,3. Однако, если из системы будет исключен 2-й или 1-й вид продукции, без пересчета этапа условной оптимизации уже не обойтись. Что именно надо считать заново в каждом из этих случаев, предлагается определить самостоятельно.

Анализ табл. 9.3 показывает, что с увеличением N уменьшаются минимальные затраты. Поэтому интересно посмотреть, что будет при значениях N , превышающих 7. Если расширить табл. 9.1-9.3 до $n=20$ по тем же формулам (9.18)-(9.21), то увидим, что наименьшие минимальные затраты равны 230 и достигаются они при $N \geq 16$. Таким образом, увеличение возможного числа циклов с 7 до 16 могло бы привести к снижению затрат на 82,7. Естественно возникает вопрос: правомерно ли

рекомендовать производству перейти на организацию 16 циклов? На первый взгляд кажется, что такой вариант явно выгоден и потому правомерен. Однако произойдет ли при этом реальное снижение затрат? Чтобы ответить на эти вопросы, нужно обратиться не к рекуррентным соотношениям, а к модели задачи. При построении модели предполагалось, что в пределах возможного числа циклов N используемое число циклов не влияет на затраты. Очевидно, что организация циклов сверх N потребует дополнительных ресурсов (оборудования, оснастки и пр.), то есть увеличения затрат, чего модель не учитывает. Следовательно, модель адекватна только в пределах справедливости указанного допущения и делать по ней выводы за этими пределами неправомерно. Чтобы выяснить, возможно ли снижение затрат при переходе на большее число циклов, нужно ввести в критерий статью затрат, связанную с организацией дополнительного количества циклов, и заново решить задачу.

Второй численный пример, рассматриваемый ниже, близок к задаче о лабиринте, описанной в начале главы, но имеет и свои отличия от всех ранее приведенных задач.

9.5. Задача о кратчайшем пути

Дан сетевой график, в котором каждой дуге поставлена в соответствие ее длина L_{ij} (рис. 9.10). Порядок нумерации вершин не имеет значения, но в приведенной нумерации задача состоит в определении кратчайшего пути из вершины 1 в вершину 7. Подобные задачи возникают непосредственно при сетевом планировании и управлении, при определении стратегии аренды оборудования, прокладке маршрутов и в других случаях.

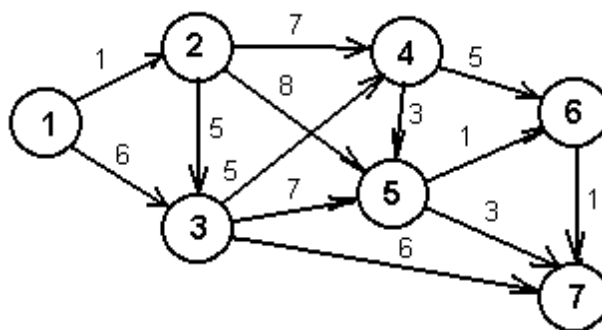


Рис.9.10

Модель задачи включает критерий - длину искомого пути

$$L = \sum_{ij \in \pi_1} L_{ij} \rightarrow \min, \quad (9.22)$$

где π_1 - путь от вершины 1 к вершине 7, и граф сети (или описывающую его матрицу). Применение метода ДП правомерно, так как задача представима как многошаговая: искомый путь есть допустимая графом последовательность дуг, а выбор дуги рассматриваем как один шаг задачи. В отличие от предыдущих задач здесь нет параметра состояния, состояние полностью определяется номером вершины, а число шагов от конкретной вершины до 7-й неоднозначно. Учитывая эти особенности, вводим последовательность функций $\{f_i\}$, $i=1,7$ так, что каждая функция есть минимальная длина пути от i -й вершины в 7-ю:

$$f_i = \min_{\pi_i \in \Pi_i} \sum_{j \in \pi_i} L_{ij}, \quad (9.23)$$

где Π_i - множество всех допустимых путей из i -й вершины в 7-ю.

Для составления функционального уравнения возьмем произвольную вершину i ($i \neq 7$) и будем определять путь из нее в вершину 7. Из этого пути выделим один шаг - выбор вершины, следующей за i -й. Множество дуг, выходящих из вершины i , обозначим I^- . Взяв произвольную дугу из множества I^- , окажемся в смежной вершине j , длина пути до которой равна L_{ij} . В соответствии с принципом оптимальности последующее поведение должно быть оптимальным, то есть путь от j -й вершины к 7-й должен иметь минимальную длину, которая согласно (9.23) есть f_j . В результате, длина пути от i -й вершины до 7-й будет равна $L_{ij} + f_j$. Так как она зависит только от j , то выбором j можно ее минимизировать. Но минимальная длина пути от i -й вершины к вершине 7 есть по определению функция f_i . Таким образом, приходим к рекуррентному соотношению задачи:

$$f_i = \min_j (L_{ij} + f_j), \quad i, j \in I^-. \quad (9.24)$$

Принципиальное отличие полученного соотношения состоит в том, что в нем нет однозначной последовательности шагов, которая выражалась в предыдущих задачах равенством $j=i-1$ (или $j=i+1$), и, следовательно, число шагов из фиксируемой вершины заранее определить нельзя (действительно, путь между двумя несмежными вершинами может состоять из разного числа дуг).

Рассуждения, которые привели к соотношению (9.24), подсказывают, что начинать условную оптимизацию следует с определения f_7 . Так как f_7 - минимальная длина пути из вершины 7 в саму себя, то $f_7=0$. Как показывает формула (9.24), вычислять можно те функции f_i , для которых уже известны все f_j , $ij \in I^-$. Поэтому следующей можно находить только функцию f_6 :

$$f_6 = \min (L_{67} + f_7) = 1 + 0 = 1.$$

Далее производим вычисления в порядке, диктуемом графом сети:

$$\begin{aligned}
 f_5 &= \min \begin{pmatrix} L_{57} + f_7 \\ L_{56} + f_6 \end{pmatrix} = \min \begin{pmatrix} 3 + 0 \\ 1 + 1 \end{pmatrix} = 2, \\
 f_4 &= \min \begin{pmatrix} L_{46} + f_6 \\ L_{45} + f_5 \end{pmatrix} = \min \begin{pmatrix} 5 + 1 \\ 3 + 2 \end{pmatrix} = 5, \\
 f_3 &= \min \begin{pmatrix} L_{37} + f_7 \\ L_{35} + f_5 \\ L_{34} + f_4 \end{pmatrix} = \min \begin{pmatrix} 6 + 0 \\ 7 + 2 \\ 5 + 5 \end{pmatrix} = 6, \\
 f_2 &= \min \begin{pmatrix} L_{24} + f_4 \\ L_{25} + f_5 \\ L_{23} + f_3 \end{pmatrix} = \min \begin{pmatrix} 7 + 5 \\ 8 + 2 \\ 5 + 6 \end{pmatrix} = 10, \\
 f_1 &= \min \begin{pmatrix} L_{12} + f_2 \\ L_{13} + f_3 \end{pmatrix} = \min \begin{pmatrix} 1 + 10 \\ 6 + 6 \end{pmatrix} = 11.
 \end{aligned}$$

В приведенных формулах подчеркнуты индексы, на которых достигается минимум. Из расчета видно, что длина кратчайшего пути из вершины 1 в вершину 7 равна 11. Сам оптимальный путь найдем, как обычно, просматривая результаты условной оптимизации в обратном порядке: из f_1 следует, что первая часть пути лежит на дуге 1-2, значит, новое состояние - это вершина 2; из f_2 находим следующую часть пути - дугу 2-5 и очередное состояние - вершину 5; поэтому далее обращаемся к f_5 и достраиваем оптимальный путь дугой 5-6 и, наконец, заканчиваем дугой 6-7. Таким образом, построен весь путь: $1 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 7$. При этом на этапе безусловной оптимизации просматривались не все функции f_i , что отличает данную задачу от ранее рассмотренных.

Очевидно, имея результаты условной оптимизации, можно легко найти кратчайшие пути из любой вершины сети в вершину 7, что характерно для метода ДП.

Для самостоятельного изучения предлагается такой вопрос: возможно ли применение метода ДП в случае, когда сеть содержит цикл, например, при добавлении дуги 6-2? Если нельзя, то почему, а если можно, то как построить расчет.

9.6. Задача с мультипликативным критерием

Все решаемые выше задачи имели критерий в виде аддитивной функции. Теперь возьмем задачу с иной целевой функцией и покажем применимость метода ДП для ее решения.

В качестве примера рассмотрим задачу о надежности некоторого устройства, состоящего из N последовательно соединенных блоков. Повышение надежности устройства обеспечивается включением дублирующих элементов в отдельные блоки. В общем случае ограничивающими факторами могут выступать затраты на дублирование, вес и/или объем устройства, надежность переключательных схем и др. Пусть основным ограничением являются затраты на дублирование Q и, кроме того, известны C_j - стоимость одного дублирующего элемента для блока j и $\varphi_j(m_j)$ - вероятность безотказной работы j -го блока с m_j дублирующими элементами. Задача состоит в определении оптимальной стратегии дублирования в пределах выделенных средств.

Очевидно, что в качестве критерия следует взять вероятность безотказной работы всего устройства P , которая при последовательном соединении блоков равна произведению вероятностей этих блоков. Поэтому модель задачи будет иметь вид

$$P = \prod_{j=1}^N \varphi_j(m_j) \rightarrow \max, \quad (9.25)$$

$$\sum_{j=1}^N C_j m_j \leq Q, \quad \forall m_j \geq 0, \text{ int}. \quad (9.26)$$

В этой модели целевая функция является *мультипликативной*, что, однако, не мешает применению метода ДП. Действительно, и в критерии, и в ограничении можно выделить составляющие их функции, описывающие отдельные блоки, хотя операторы, примененные к этим частным функциям в (9.25) и (9.26), разные. Поэтому задача представима как многошаговая, а шагом является определение числа дублирующих элементов для одного блока. В последовательности шагов расположение блоков может не соответствовать реальной схеме их соединения в устройстве. Порядок расположения фиксируется исходя из соображений, которые были приведены в разделе 9.3.

Пронумеруем шаги, как и ранее, справа налево. Очевидно, что для принятия решения нужно знать свои возможности, которые в данной задаче определяются средствами на дублирование. Поэтому в качестве параметра состояния следует взять допустимый уровень затрат на дублирование q , который на любом шаге может принимать любые значения из диапазона $[0, Q]$.

Теперь можно ввести последовательность функций $\{f_k(q)\}$, $k=1, N$. Каждая функция имеет смысл максимальной вероятности безотказной работы k оставшихся блоков при допустимом уровне затрат на дублирование q :

$$f_k(q) = \max \prod_{j=1}^k \varphi_j(m_j). \quad (9.27)$$

Для получения функционального уравнения рассмотрим k блоков, на дублирование которых можно затратить q средств. Если в k -й блок включить m_k дублирующих элементов, а оставшиеся после этого средства $(q - C_k m_k)$ использовать оптимально на дублирование $k-1$ блоков, то с учетом (9.27) и последовательного соединения k -го блока с остальными $k-1$ блоками вероятность безотказной работы k блоков будет равна $[\varphi_k(m_k)f_{k-1}(q - C_k m_k)]$. Максимизируя это выражение по m_k , приходим к искомому рекуррентному соотношению

$$f_k(q) = \max_{0 \leq m_k \leq [q/C_k]} [\varphi_k(m_k)f_{k-1}(q - C_k m_k)], \quad k \geq 2, \quad (9.28)$$

где $[q/C_k]$ означает целую часть от q/C_k .

Условная оптимизация начинается с вычисления первой функции последовательности

$$f_1(q) = \max_{0 \leq m_1 \leq [q/C_1]} \varphi_1(m_1)$$

и затем продолжается по формуле (9.28). Безусловная оптимизация проводится в обратном порядке, то есть от f_N к f_1 , с исходного состояния $q_N = Q$ и последующим пересчетом по уравнению состояния

$$q_{k-1} = q_k - C_k m_k.$$

Таким образом, мультипликативность целевой функции не изменяет процедуру динамического программирования. Отличие от ранее рассмотренных задач лишь в том, что выражение в правой части рекуррентного соотношения не аддитивное, а мультипликативное. Очевидно, что мультипликативность ограничения или одновременно ограничения и целевой функции также не вызовет осложнений.

9.7. Усложненная задача

Выше рассматривались задачи, в которых метод ДП позволял переходить от исходной задачи с N переменными к N одномерным задачам. Однако могут быть и более сложные случаи. Покажем это на примере задачи распределения ассигнований в некоторой крупной корпорации.

Корпорация включает N предприятий, производящих продукцию с номенклатурой m_j каждое ($j = \overline{1, N}$). Прибыль, приносимая i -й продукцией j -го предприятия, определяется по известной зависимости $r_i^j(x_{ij})$, где x_{ij} - объем выделенных ассигнований. Задача состоит в оптимальном распределении ассигнований в размере A между предприятиями и видами продукции при условии, что предприятию может быть выделено на все виды продукции не более B_j средств.

Критерием оптимальности здесь является суммарная прибыль корпорации. Прибыль j -го предприятия

$$R_j(x_{1j}, x_{2j}, \dots, x_{m_jj}) = \sum_{i=1}^{m_j} r_i^j(x_{ij}). \quad (9.29)$$

Тогда модель задачи запишется в виде:

$$R = \sum_{j=1}^N R_j(x_{1j}, x_{2j}, \dots, x_{m_jj}) \rightarrow \max, \quad (9.30)$$

$$\sum_{j=1}^N \sum_{i=1}^{m_j} x_{ij} \leq A, \quad (9.31)$$

$$\sum_{i=1}^{m_j} x_{ij} \leq B_j, \quad j = \overline{1, N}, \quad (9.32)$$

$$\forall x_{ij} \geq 0. \quad (9.33)$$

Ограничения (9.32) связывают переменные, относящиеся к одному предприятию. Поэтому в качестве шага берем выделение ассигнований одному предприятию, и задача становится N -шаговой. При этом состояние определяется только величиной ассигнований a ($a \leq A$), распределяемых между рассматриваемыми предприятиями. Ассигнования B_j не могут быть параметрами состояния, так как их значения на последующих шагах не зависят от решения на текущем шаге. Таким образом, функции последовательности будут зависеть от одного параметра состояния:

$$f_k(a) = \max \sum_{i=1}^k R_j(x_{1j}, x_{2j}, \dots, x_{m_jj}), \quad k = \overline{1, N} \quad (9.34)$$

и представлять собой максимальную прибыль k предприятий при суммарных ассигнованиях в пределах a . Рассуждения на основе принципа оптимальности приводят к следующему рекуррентному соотношению:

$$f_k(a) = \max_{\sum_{i=1}^{m_k} x_{ik} \leq \min(B_k, a)} [R_k(x_{1k}, x_{2k}, \dots, x_{m_kk}) + f_{k-1}(a - \sum_{i=1}^{m_k} x_{ik})]. \quad (9.35)$$

Существенное отличие этого соотношения от рассмотренных в предыдущих разделах состоит в том, что правая часть зависит более чем от одной переменной (в данном случае от m_k переменных) и поэтому придется искать условный максимум функции многих переменных. В результате решение задачи значительно усложняется.

Облегчить решение можно с помощью декомпозиции задачи подобно тому, как было сделано в примере на применение множителей Лагранжа (гл.3). Согласно такому подходу сначала найдем параметрические

оптимальные решения для каждого предприятия в отдельности, а затем рассмотрим их совместно.

На уровне предприятия имеем такую модель:

$$R_j(x_{1j}, x_{2j}, \dots, x_{m_jj}) \rightarrow \max,$$

$$\sum_{i=1}^{m_j} x_{ij} \leq B_j, \quad \forall x \geq 0.$$

Очевидно, что данная задача может решаться также методом ДП. Она представима как m_j -шаговая с одним параметром состояния b_j ($b_j \leq B_j$). Используя представление (9.29), введем последовательность функций

$$F_v^j(b_j) = \max \sum_{i=1}^v r_i^j(x_{ij}), \quad v = \overline{1, m_j},$$

которая подчиняется рекуррентному соотношению

$$F_v^j(b_j) = \max_{0 \leq x_{vj} \leq b_j} [r_v^j(x_{vj}) + F_{v-1}^j(b_j - x_{vj})], \quad v \geq 2, \quad (9.36)$$

выводимому из принципа оптимальности. Первая функция этой последовательности вычисляется непосредственно:

$$F_1^j(b_j) = \max_{0 \leq x_{1j} \leq b_j} r_1^j(x_{1j}). \quad (9.37)$$

Для каждого предприятия по формулам (9.37) и (9.36) находится последовательность функций $F_v^j(b_j)$. Зная $F_{m_j}^j(b_j)$ – максимальную прибыль j -го предприятия от всех видов продукции, рекуррентное соотношение (9.35) можно представить в виде

$$f_k(a) = \max_{b_k \leq \min(B_k, a)} [F_{m_k}^k(b_k) + f_{k-1}(a - b_k)], \quad (9.38)$$

для которого

$$f_1(a) = \max_{b_1 \leq \min(B_1, a)} F_{m_1}^1(b_1). \quad (9.39)$$

Расчет проводим в следующем порядке. Сначала выполним вычисления по формулам (9.38) и (9.39). Имея результаты условной оптимизации (например, в виде таблиц), переходим к этапу безусловной оптимизации на верхнем уровне (на уровне корпорации). По исходному состоянию $a=A$ входим в таблицу функции $f_N(a)$ и находим максимальную прибыль от всех N предприятий и оптимальный объем ассигнований b_N^* , выделяемый предприятию N . В результате состояние изменяется с A на $A - b_N^*$. С этим значением параметра состояния входим в таблицу функции $f_{N-1}(a)$ и извлекаем b_{N-1}^* , снова пересчитываем состояние и продолжаем движение по таблицам f_k вплоть до получения b_1^* . После этого проводится безусловная оптимизация на нижнем уровне. При этом каждое предприятие рассматривается независимо от других. Для примера возьмем

j -е предприятие. По найденному выше значению b_j^* из таблицы функции $F_{m_j}^j(b_j)$ берем прибыль j -го предприятия при оптимальном распределении ассигнований и оптимальные ассигнования $x_{m_j j}^*$ на продукцию m_j . По новому состоянию $(b_j^* - x_{m_j j}^*)$ входим в таблицу функции $F_{m_j-1}^j(b_j)$ и находим $x_{m_j-1, j}^*$.

Точно так же последовательно проходим по остальным таблицам функций F^j , в результате чего определяются все x_{ij}^* . Повторив аналогичную процедуру для всех предприятий, получим оптимальное решение исходной задачи.

Как видно из формул (9.36)-(9.39), на каждом шаге условной оптимизации как верхнего, так и нижнего уровня, отыскивался максимум функции одной переменной. Такое упрощение решения явилось следствием декомпозиции исходной задачи.

9.8. Многомерные задачи динамического программирования

В рассмотренных выше задачах вектор состояния S имел размерность 1 или 0 (нулевая в задаче о кратчайшем пути). Если размерность вектора S (число параметров состояния) больше 1, то говорят, что задача **многомерна** в смысле динамического программирования. Многомерные задачи порождают определенные проблемы при реализации вычислительной схемы метода ДП. Покажем на примерах, как возникают такие задачи и с чем приходится сталкиваться при их решении.

Задача 1. Необходимо распределить два вида ресурсов в объеме X и Y соответственно между N производствами при известных функциях прибыли $R_j(x_j, y_j)$, $j=1, N$. Здесь x_j, y_j - количество ресурса 1-го и 2-го вида, потребляемое j -м производством.

Запишем модель задачи:

$$R = \sum_{j=1}^N R_j(x_j, y_j) \rightarrow \max, \quad (9.40)$$

$$\sum_{j=1}^N x_j = X, \quad (9.41)$$

$$\sum_{j=1}^N y_j = Y, \quad (9.42)$$

$$\forall x_j, y_j \geq 0. \quad (9.43)$$

Очевидно, что задача представима как N -шаговая (по числу производств). Для принятия решения по распределению ресурсов нужно знать их количество и число производств, участвующих в распределении. При

выделении ресурсов одному из производств изменяется объем ресурсов, направляемых на остальные производства. Поэтому состояние характеризуется двумя параметрами: количеством ресурса 1-го вида V и 2-го вида U ($V \leq X$, $U \leq Y$). Введем последовательность функций:

$$f_k(V, U) = \max_{j=1}^k R_j(x_j, y_j), \quad k = \overline{1, N} \quad (9.44)$$

и рассмотрим k оставшихся производств (шагов), между которыми нужно распределить ресурсы в количестве V и U . Приняв решение о выделении k -му производству произвольного допустимого количества ресурсов x_k и y_k , будем иметь прибыль от этого производства $R_k(x_k, y_k)$, а на остальные $k-1$ шагов останется ресурсов $V-x_k$ 1-го вида и $U-y_k$ 2-го вида. Следуя принципу оптимальности, распределим оставшиеся ресурсы оптимальным образом, что обеспечит максимальную прибыль от $k-1$ производств, то есть $f_{k-1}(V-x_k, U-y_k)$. Прибыль же от всех k шагов составит

$$R_k(x_k, y_k) + f_{k-1}(V-x_k, U-y_k).$$

Так как она зависит от двух переменных, то варьируя их в допустимой области, можно получить максимальную прибыль, которая по определению (9.44) есть $f_k(V, U)$. Таким образом, приходим к функциональному уравнению динамического программирования:

$$f_k(V, U) = \max_{\substack{0 \leq x_k \leq V \\ 0 \leq y_k \leq U}} [R_k(x_k, y_k) + f_{k-1}(V-x_k, U-y_k)], \quad k \geq 2, \quad (9.45)$$

Задача 2. Распределению подлежит один вид ресурса, но в системе имеются ограничения, связанные с его использованием. Это могут быть ограничения на общий объем, вес, габариты, стоимость и др. Рассмотрим случай двух ограничений, согласно которым фактические значения учитываемых показателей использования ресурса не могут превышать величин A и B . Тогда модель задачи можно представить в виде:

$$Z = \sum_{i=1}^N Z_i(x_i) \rightarrow \text{extr}, \quad (9.46)$$

$$\sum_{i=1}^N \varphi_i(x_i) \leq A, \quad (9.47)$$

$$\sum_{i=1}^N \psi_i(x_i) \leq B, \quad (9.48)$$

$$\forall x_i \geq 0, \quad (9.49)$$

где x_i - количество ресурса, выделяемое i -му потребителю;

$Z_i(x_i)$ - показатель эффективности i -го потребителя;

$\varphi_i(x_i)$, $\psi_i(x_i)$ - функции ограничиваемых показателей.

Для применения метода ДП необходимо, как всегда, определить параметры состояния. Нетрудно видеть, что информация, достаточная для принятия решения на текущем шаге, должна содержать значения ограничивающих показателей. Они же непосредственно влияют на оптимальное значение эффективности системы. Кроме того, изменение ресурса на i -м шаге влечет изменение этих показателей к следующему шагу. Следовательно, в процедуре ДП ограничивающие показатели a и b выступают в качестве параметров состояния ($a \leq A, b \leq B$).

Определим функции последовательности как

$$f_k(a, b) = \text{extr} \sum_{i=1}^k Z_i(x_i), \quad k = \overline{1, N}. \quad (9.50)$$

Не повторяя известных рассуждений, опирающихся на принцип оптимальности ДП, приведем в окончательном виде рекуррентное соотношение для рассматриваемой задачи:

$$f_k(a, b) = \text{extr}_{\substack{\varphi_k(x_k) \leq a \\ \psi_k(x_k) \leq b}} [Z_k(x_k) + f_{k-1}(a - \varphi_k(x_k), b - \psi_k(x_k))]. \quad (9.51)$$

Проанализируем полученные выражения (9.45) и (9.51). При их выводе не возникло принципиальных затруднений. Хотя число распределяемых ресурсов разное, в обеих задачах функции последовательности зависят от **двух параметров состояния**. Нетрудно понять теперь, что размерность вектора состояния связана с числом ограничений, включающих переменные всех шагов. Трудоемкость решения сравниваемых уравнений разная, так как различно число переменных, по которым ищется экстремум. При нелинейных функциях φ_k и ψ_k может усложниться перебор допустимых x_k в (9.51). Однако эти различия не столь существенны, большее значение имеет одинаковая размерность вектора состояния.

Покажем на задаче 1, к чему приводит увеличение числа параметров состояния. Решение уравнения (9.45) проводится для всех возможных состояний. Число таких состояний зависит от шага дискретности и числа параметров состояния. Пусть m_x и m_y - число возможных значений ресурсов X и Y соответственно, тогда число возможных состояний будет равно $m_x m_y$. Для каждого из них в результирующей таблице решения уравнения (9.45) на k -м шаге должны быть представлены V, U, x_k^*, y_k^* и $f_k(V, U)$. Для вещественного числа требуется хотя бы 6 байт памяти, поэтому одна строка результирующей таблицы будет занимать не менее 30 байт. При $m_x = m_y = 100$ число возможных состояний, а значит, и число строк составит 10^4 . Следовательно, для запоминания одной таблицы потребуется около 300 Кбайт. Для эффективного расчета нужно иметь в **оперативной памяти** две таблицы - k -ю и $(k-1)$ -ю. Вынос этих таблиц во внешнюю память сделает метод ДП практически малоприменимым, так как

затраты времени на решение задачи многократно увеличатся. Теперь нетрудно оценить, насколько будут возрастать требования к памяти с увеличением числа параметров состояния до 3, 4 и более.

Таким образом, решение многомерных задач наталкивается на проблему реализации вычислений по рекуррентным соотношениям. Сама идея метода ДП настолько привлекательна, что обнаружив трудно преодолимое препятствие применению метода в виде отмеченной проблемы, Беллман назвал ее "*проклятием размерности*".

Естественно, появилась необходимость найти пути решения возникшей проблемы. К предложенным средствам можно отнести использование множителей Лагранжа, аппроксимацию в области стратегий, принципиально иной способ решения рекуррентного уравнения, применение идей последовательного анализа вариантов и др. Однако ни одно из них не является универсальным, а в ряде случаев не работают все. Ниже рассматривается один из способов преодоления проблемы размерности, нашедший практическое применение.

9.9. Снижение размерности с помощью множителей Лагранжа

Покажем на примере задачи 1, каким образом введение множителей Лагранжа позволяет заменить исходную задачу рядом задач с меньшей размерностью вектора состояния.

В задаче 1 состояние описывалось двумя параметрами, что обуславливалось наличием двух ограничений. Для уменьшения размерности на 1 достаточно из модели (9.40)-(9.43) убрать одно из ограничений, что можно сделать, если удаляемое ограничение включить в критерий задачи с неопределенным множителем Лагранжа λ . Тогда модель измененной задачи примет вид:

$$\bar{R} = \sum_{j=1}^N R_j(x_j, y_j) - \lambda \sum_{j=1}^N y_j \rightarrow \max_{\{x_j, y_j\}}, \quad (9.52)$$

$$\sum_{j=1}^N x_j = X, \quad (9.53)$$

$$\forall x_j, y_j \geq 0. \quad (9.54)$$

Как будет доказано ниже, задача (9.52)-(9.54) при определенных условиях эквивалентна исходной задаче (9.40)-(9.43). Так как ограничения (9.53), (9.54) не связывают между собой переменные y_j , то есть они стали независимыми, то справедлива следующая цепочка равенств:

$$\begin{aligned} \max_{\{x_j, y_j\}} \bar{R} &= \max_{\{x_j\}} \max_{\{y_j\}} \sum_{j=1}^N [R_j(x_j, y_j) - \lambda y_j] = \max_{\{x_j\}} \sum_{j=1}^N \max_{y_j} [R_j(x_j, y_j) - \lambda y_j] = \\ &= \max_{\{x_j\}} \sum_{j=1}^N h_j(x_j) \quad \text{при } \lambda = \text{const}, \end{aligned}$$

где

$$h_j(x_j) = \max_{y_j \geq 0} [R_j(x_j, y_j) - \lambda y_j], \quad j = \overline{1, N}. \quad (9.55)$$

Функции $h_j(x_j)$ имеют смысл, если максимум в (9.55) достигается при конечных значениях y_j , что всегда будет, когда

$$\frac{R_j(x_j, y_j)}{y_j} \rightarrow 0 \quad \text{при } y_j \rightarrow \infty. \quad (9.56)$$

Это условие ограничивает применение данного способа, но в рассматриваемой задаче оно, очевидно, выполняется, так как при неограниченном возрастании ресурса у рост прибыли будет замедляться.

Как видно из (9.55), вычисление функции $h_j(x_j)$ при фиксированном значении λ заключается в нахождении максимума функции одной переменной для всех возможных значений x_j , что не вызывает особых затруднений (для дифференцируемых $R_j(x_j, y_j)$ максимум можно найти аналитически). При известных $h_j(x_j)$, $j=1, N$ задача (9.52)-(9.54) сводится к следующей:

$$\sum_{j=1}^N h_j(x_j) \rightarrow \max, \quad (9.57)$$

$$\sum_{j=1}^N x_j = X, \quad (9.58)$$

$$\forall x_j \geq 0. \quad (9.59)$$

Получили уже знакомую нам задачу распределения одного ресурса. Для решения ее методом ДП введем последовательность функций

$$f_k(V) = \max \sum_{j=1}^k h_j(x_j),$$

где V - параметр состояния, значения которого не превосходят X . Для них справедливо рекуррентное соотношение:

$$f_k(V) = \max [h_k(x_k) + f_{k-1}(V - x_k)], \quad (9.60)$$

в котором $f_1(V) = h_1(V)$, $x_1^* = V$.

Вычисления по формуле (9.60) проводятся, как обычно, от f_1 к f_N , затем в обратном порядке - безусловная оптимизация, начиная с $V=X$, которая дает значения x_1^* . По последним из функций $h_j(x_j)$ находятся значения y_j^* . Теперь следует вспомнить об условии (9.42), которое не вошло в измененную задачу. Если $\sum y_j^* = Y$, то найденное решение x_j^* , y_j^* является оптимальным решением задачи (9.40)-(9.43). В противном случае придется продолжить расчеты.

Нетрудно увидеть, что оптимальное решение измененной задачи зависит от принятого значения множителя λ . Поэтому при невыполнении

условия (9.42) нужно изменить значение λ и повторить весь расчет, начиная с вычисления функций $h_j(x_j)$. В данной задаче при изменении λ можно воспользоваться очевидным свойством: с увеличением λ будет монотонно убывать $\sum y_j^*$ и наоборот. В более сложных ситуациях можно воспользоваться одним из методов одномерного поиска нелинейного программирования. Таким образом, равенство (9.42) может быть выполнено с любой заданной точностью.

Чтобы нагляднее представить весь расчет с использованием множителей Лагранжа, приведем его алгоритм в виде блок-схемы (рис.9.11). Как видно из алгоритма, функции $f_k(V)$ и $h_j(x_j)$, участвующие в расчете, зависят от одного параметра состояния, и, следовательно, поставленная цель достигнута.

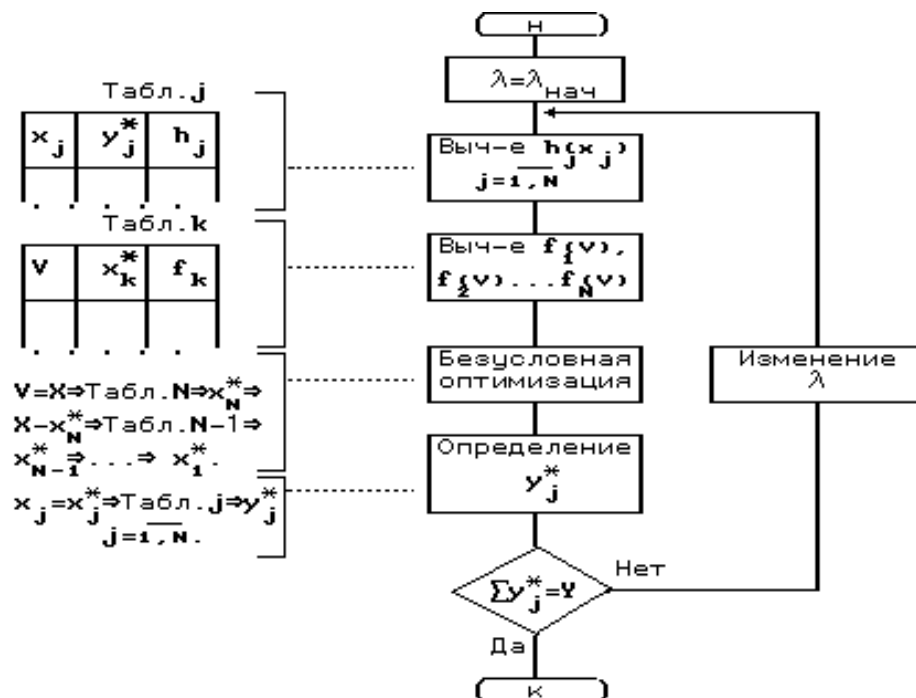


Рис.9.11

Теперь покажем эквивалентность задач (9.40)-(9.43) и (9.52)-(9.54), понимая под этим совпадение решений. Следуя Беллману, доказательство проведем от противного. Имея оптимальное решение измененной задачи x_j^*, y_j^* , предположим, что оптимальное решение исходной задачи иное, а именно, \hat{x}_j, \hat{y}_j . Тогда для критерия исходной задачи должно выполняться неравенство

$$\sum_{j=1}^N R_j(\hat{x}_j, \hat{y}_j) > \sum_{j=1}^N R_j(x_j^*, y_j^*). \quad (9.61)$$

Так как $\sum \hat{y}_j = Y$ по условию исходной задачи, а $\sum y_j^* = Y$ по алгоритму решения измененной задачи, то $\sum \hat{y}_j = \sum y_j^*$. Вычитание одной и той же величины, умноженной на λ , из левой и правой частей выражения (9.61) не меняет знак неравенства:

$$\sum_{j=1}^N R_j(\hat{x}_j, \hat{y}_j) - \lambda \sum_{j=1}^N \hat{y}_j > \sum_{j=1}^N R_j(x_j^*, y_j^*) - \lambda \sum_{j=1}^N y_j^*. \quad (9.62)$$

Но здесь и слева, и справа имеем выражение критерия измененной задачи, по которому оптимальным является решение x_j^*, y_j^* . Таким образом, неравенство (9.62), вытекающее из допущения существования разных решений, противоречит исходной посылке и потому такое допущение неверно, что доказывает совпадение решений исходной и измененной (эквивалентной) задач.

Для задачи 2 применение метода множителей Лагранжа реализуется проще. Модель измененной задачи можно записать по аналогии с вышеприведенным случаем в виде:

$$\bar{Z} = \sum_{i=1}^N Z_i(x_i) - \lambda \sum_{i=1}^N \psi_i(x_i) \rightarrow \text{extr}, \quad \sum_{i=1}^N \varphi_i(x_i) \leq A, \quad \forall x_i \geq 0.$$

Для функций последовательности, определенных как

$$f_k(a) = \text{extr}_{\{x_i\}} \sum_{i=1}^k [Z_i(x_i) - \lambda \psi_i(x_i)], \quad k = \overline{1, N},$$

справедливо следующее рекуррентное соотношение

$$f_k(a) = \text{extr}_{\varphi_k(x_k) \leq a} [Z_k(x_k) - \lambda \psi_k(x_k) + f_{k-1}(a - \varphi_k(x_k))]. \quad (9.63)$$

Как видно, здесь нет дополнительных функций h_j и вычисления можно проводить сразу по рекуррентной формуле (9.63), задавшись предварительно значением λ . После нахождения решения проверяется условие (9.48) - $\sum \psi_j(x_j^*) \leq B$ и, если оно не выполняется, то необходимо изменить значение λ и повторить расчет. Таким способом достигается эквивалентность исходной и измененной задач и получение оптимального решения с помощью последовательности функций, зависящих только от одного параметра состояния.

В общем случае, когда вектор состояния исходной задачи имеет размерность m , можно использовать q множителей Лагранжа ($q < m$), что позволит снизить размерность вектора состояния измененной задачи до $m - q$. При этом выполнение исключенных из условий исходной задачи q ограничений может быть обеспечено управлением таким же числом множителей Лагранжа. Однако увеличение размерности вектора состояния

и соответственно числа множителей Лагранжа ведет к значительно более быстрому росту трудоемкости решения измененной задачи. Поэтому проблема "проклятия размерности" остается, ограничивая применение метода ДП задачами с небольшим числом параметров состояния.

Несмотря на указанный недостаток метод динамического программирования находит широкое применение для решения многих задач исследования операций, в том числе задач распределения, замены, кратчайшего пути, упорядочения и др.

9.10. Задания для самостоятельной работы

Работа заключается в решении одной оптимизационной задачи методом динамического программирования. Содержание задачи по вариантам и соответствующие исходные данные приводятся ниже.

Указания. Целесообразно придерживаться следующего порядка решения задачи:

1. Составить модель задачи.
2. Представить задачу как многошаговую: выделить и пронумеровать шаги. Эта нумерация должна сохраниться до конца решения.
3. Определить параметры состояния.
9. Составить уравнение состояния.
5. Ввести последовательность функций, четко определив их смысл и математическое представление.
6. Записать первую функцию последовательности и вывести рекуррентное соотношение с указанием переменных и области их изменения, по которым определяется экстремум в соотношении.
7. Провести условную оптимизацию. При решении функциональных уравнений там, где возможно, следует использовать отыскание оптимума в общем виде (классическим методом). Существенное значение имеет обоснованный выбор диапазона значений параметров состояния и шага дискретизации.
8. Используя результаты условной оптимизации, провести безусловную оптимизацию для всех заданных значений исходного состояния.
9. Выписать оптимальное решение и все соответствующие ему показатели, сделать выводы по результатам решения.
10. Осмыслить полученные результаты.

В некоторых вариантах используется процедура снижения размерности. Для них перечисленные действия недостаточны, так как может быть необходим еще один этап оптимизации, а условная и безусловная оптимизации могут повторяться несколько раз. Однако в целом схема решения аналогична.

В большинстве вариантов один из параметров задан в диапазоне,

например 50–60. Это означает, что нужно найти решения для значений этого параметра от 50 до 60.

При решении задачи методом динамического программирования совершенно естественно применение компьютера, особенно в вариантах с большим объемом вычислений.

При использовании компьютера следует приводить укрупненную блок-схему алгоритма, текст программы с краткими комментариями, оформленные в таблицы конечные результаты расчета по шагам (или полные результаты, включая промежуточные при поиске экстремума).

При представлении конечных результатов необходимо также привести значения критерия на каждом шаге для возможных значений переменных. (Например, при распределении ресурса X по трем предприятиям это доходы $r_1(x_1)$, $r_2(x_2)$ и $r_3(x_3)$).

Безусловная оптимизация проводится вручную с соответствующими пояснениями действий.

Варианты 1.1 - 1.3

В каждый месяц планового периода известен спрос на машины $D_i(t=1,...,T)$, которые выпускает предприятие. Запас машин на складе предприятия составляет i_0 единиц на начало периода планирования. Общие затраты предприятия складываются из затрат на производство машин $C(X)$ и затрат на их содержание на складе.

Пусть затраты на производство одной машины определяются выражением

$$C(x)=C_0-C_1X^2,$$

а затраты на хранение одной машины в течение месяца равны h . Месячное производство машин ограничено величиной B , а на складе можно хранить не более M машин. Отправка потребителю производится в конце месяца, а выпуск машин - равномерно в течение каждого месяца.

Требуется определить оптимальный план выпуска машин, а также вычислить максимальные потери от неоптимальности.

Значения параметров по вариантам даны в табл. 1.

Таблица 1

Вариант	D_1	D_2	D_3	D_4	D_5	C_0	C_1	h	B	M	i_0
1.1	7	5	6	4	7	12	0,04	2	10	12	0-8
1.2	10	13	12	11	8	10	0,03	1	15	18	5-8
1.3	11	14	15	15	13	20	0,07	3	16	18	2-12

Варианты 2.1 – 2.3

Характеристики оборудования зависят от его возраста t :

$r(t)$ - стоимость ежегодно производимой продукции;

$u(t)$ - годовые эксплуатационные затраты;

$s(t)$ - остаточная стоимость (выручка от продажи оборудования).

На начало планового периода из N лет оборудование имеет возраст $t=t_0$. В начале любого года оборудование можно сохранить или продать и купить такое же новое по цене P (включая установку и пр.). Продолжительность замены много меньше года.

Необходимо разработать оптимальную политику замены оборудования и сравнить её с наихудшей. Исходные данные приведены в табл. 2.

Таблица 2

Ва- риант	Функ- ции	t											P
		0	1	2	3	4	5	6	7	8	9	10	
2.1	$r(t)$	29	27	27	25	24	23	23	21	20	18	16	20
	$u(t)$	7	8	9	10	11	11	12	14	15	15	16	
2.2	$r(t)$	25	24	23	21	20	20	19	19	18	17	17	16
	$u(t)$	9	10	10	11	12	13	13	14	15	16	17	
2.3	$r(t)$	33	32	32	30	29	28	27	26	24	22	20	23
	$u(t)$	10	12	12	14	14	15	16	17	17	18	19	

$$S(t) = P \cdot 2^{-t}; \quad t_0=0...6; \quad N=10.$$

Примечание: рассматривается замена одной единицы оборудования.

Варианты 3.1 – 3.3

Автомобиль должен пройти 4 этапа длиной l_i километров каждый. Известны зависимость скорости движения автомобиля V_i (км/ч) от расхода горючего q (л/км) для каждого этапа. Скорость движения на трех этапах ограничена. Эти данные представлены в табл. 3.

Необходимо определить скорости движения, обеспечивающие минимальное время пробега при заданном количестве горючего Q (л). Показать, как отразится на решении изменение Q на 10% в обе стороны.

Таблица 3

Ва- риант	Пара- метры	1 этап	2 этап	3 этап	4 этап	Q
3.1	l_i	100	200	150	250	6000
	$V_i(q)$	$160(1-e^{-0,2q})$	$8q$	$120(1-e^{-0,1q})$	$30q-1,5q^2$	
	$V_{i \max}$	130	100	50	-	
3.2	l_i	120	180	250	200	5500
	$V_i(q)$	$130(1-e^{-0,1q})$	$30q-1,5q^2$	$8q$	$160(1-e^{-0,2q})$	
	$V_{i \max}$	50	-	100	130	
3.3	l_i	150	200	200	180	5000
	$V_i(q)$	$8q$	$170(1-e^{-0,2q})$	$110(1-e^{-0,1q})$	$30q-1,5q^2$	
	$V_{i \max}$	90	140	60	-	

Варианты 4.1 – 4.3

Планируется деятельность предприятия на 5 лет. Известны функции дохода $r_i(x_i)$ предприятия в i -м году (x_i - количество средств, вкладываемых в i -м году). Требуется распределить имеющиеся средства в количестве Q условных единиц по годам планируемого периода, если часть дохода P текущего года вкладывается в последующие годы, и вкладывать в год нужно не менее двух единиц средств. Как изменится решение, если снять последнее условие? Все исходные данные представлены в табл. 9.

Таблица 4

Вариант	Q	$r_1(x)$	$r_2(x)$	$r_3(x)$	$r_4(x)$	$r_5(x)$	P, %
4.1	13-16	$10\sqrt{x}$	$1,4x$	$30x-1,5x^2$	$8\sqrt{x}$	$1,6x$	8
4.2	15-18	$2,5x$	$6\sqrt{x}$	$1,5x$	$20x-3x^2$	$10\sqrt{x}$	6
4.3	16-19	$3\frac{5+2x}{20-x}$	$1,2x$	$20x-2x^2$	$10\sqrt{x}$	$3\sqrt{2x}$	10

Варианты 5.1 – 5.3

Средства Q распределяются между тремя предприятиями одной фирмы, которые связаны технологическим циклом производства так, что продукция 1-го предприятия является полуфабрикатом для второго, а продукция первых двух предприятий служит полуфабрикатом для 3-го предприятия. Известны соответствующие функции дохода $r_1(x_1)$, $r_2(x_2, x_1)$, $r_3(x_3, x_1+x_2)$, где x_i – вкладываемые в i -е предприятие средства (табл. 5-7). Необходимо найти оптимальное распределение.

Таблица 5

Вариант	Q	$r_1(x_1)$ при x_1								
		1	2	3	4	5	6	7	8	9
5.1	7-8	2,2	3,3	4,4	5,6	5,9	6,0	6,0		
5.2	8-9	2,5	2,9	3,5	4,2	5,5	5,7	5,6	5,5	
5.3	9-10	2,1	2,9	3,9	5,0	5,8	6,7	6,8	6,7	6,5

Таблица 6

Вариант	x_1	$r_2(x_1, x_2)$ при x_2							
		1	2	3	4	5	6	7	8
5.1	1	2,2	2,8	3,1	4,3	6,0	6,5		
	2	3,1	4,2	5,3	7,1	8,0			
	3	3,3	4,5	6,1	7,3				
	4	3,5	4,8	6,7					
	5	4,7	5,9						
	6	5,5							

Вариант	x_1	$r_2(x_1, x_2)$ при x_2							
		1	2	3	4	5	6	7	8
5.2	1	2,7	3,1	3,6	4,1	4,5	4,8	4,7	
	2	3,0	3,5	4,1	4,7	4,9	5,1		
	3	3,1	3,6	4,3	4,9	5,3			
	4	3,5	3,9	4,5	5,1				
	5	3,7	4,1	4,9					
	6	4,0	4,6						
	7	4,4							
5.3	1	2,7	3,8	4,7	5,3	5,9	6,5	6,4	6,3
	2	2,9	3,9	4,9	5,5	6,1	6,7	6,6	
	3	3,0	3,9	5,0	5,7	6,5	6,5		
	4	3,2	4,1	5,1	5,8	6,5			
	5	3,3	4,2	5,3	6,0				
	6	3,5	4,6	5,8					
	7	3,6	4,8						
	8	4,3							

Таблица 7

Вариант	x_1+x_2	$r_3(x_3, x_1+x_2)$ при x_3							
		1	2	3	4	5	6	7	8
5.1	1	3,4	3,8	4,2	5,0	5,2	5,0		
	2	3,7	4,1	4,5	5,3	5,1			
	3	3,7	4,2	4,6	5,5				
	4	4,0	4,5	4,8					
	5	4,2	4,8						
	6	4,6							
5.2	1	3,1	3,5	3,9	4,3	4,6	4,5	4,3	
	2	3,3	3,6	3,9	4,4	4,7	4,6		
	3	3,4	3,8	4,1	4,6	4,4			
	4	3,6	3,9	4,3	4,7				
	5	3,7	4,0	4,5					
	6	3,9	4,3						
	7	4,2							
5.3	1	3,2	4,3	4,9	5,7	6,6	6,8	6,6	6,5
	2	3,4	4,5	5,2	5,9	6,9	6,8	6,7	
	3	3,6	4,8	5,5	6,4	7,2	7,1		
	4	3,9	5,2	5,9	6,8	7,0			
	5	4,1	5,5	6,4	7,0				
	6	4,5	5,8	6,7					
	7	4,7	6,5						
	8	5,4							

Варианты 6.1 – 6.3

Предприятие выпускает 4 вида продукции, используя два вида сырья в пределах выделенных фондов B_1 и B_2 . Нормы расхода сырья на единицу продукции приведены в табл. 8.

Таблица 8

Сырье	Нормы расхода на единицу продукции (по вариантам)				B
	1-й вид	2-й вид	3-й вид	4-й вид	
I	2; 1; 3	4; 3; 2	1; 2; 2	2; 3; 1	18
II	4; 2; 1	2; 4; 3	3; 1; 1	3; 2; 5	24

Зависимость прибыли от объема производства каждого вида продукции задана аналитически (табл. 9).

Таблица 9

Продукция	Вариант 6.1	Вариант 6.2	Вариант 6.3
1-й вид	$30\sqrt[3]{x/5}$	$5 \cdot x$	$10x - x^2$
2-й вид	$6x$	$9x - x^2$	$0,8x^2$
3-й вид	$1,2x^2$	$25\sqrt[3]{x/4}$	$20(1 - e^{-0,3x})$
4-й вид	$8x - x^2$	$1,1x^2$	$7x$

Определить оптимальный план производства, рассматривая выпуск продукции только в целых единицах.

Варианты 7.1 – 7.3

Для объединения, включающего два предприятия, формируется план на 4 года. Известны функции дохода i -го предприятия в j -м году $g_{ij}(x_{ij})$, где x_{ij} - количество средств, вкладываемое в i -е предприятие в j -м году. Вложенные средства уменьшаются за год до величины $\varphi_i(x)$. Оставшиеся в конце года средства используются предприятием в последующие годы.

Требуется распределить имеющиеся средства в количестве R единиц по предприятиям и годам планового периода.

$$g_{1j} = k_{1j}(1 - e^{-\alpha_j x_{1j}}); \quad g_{2j} = k_{2j}\sqrt{x_{2j}}; \quad \varphi_i(x) = [\beta_i x].$$

Квадратные скобки означают взятие целого по правилам округления. Значения всех параметров даны в табл. 10 и 11.

Таблица 10

J	K_{1j}	K_{2j}
1	10	3,8
2	8,5	4,1
3	9,1	3,9
4	11,6	4,3

Таблица 11

Вариант	α_1	α_2	α_3	α_4	β_1	β_2	R
7.1	0,25	0,3	0,3	0,4	0,26	0,3	13-15
7.2	0,2	0,25	0,3	0,35	0,3	0,2	10-14
7.3	0,3	0,3	0,32	0,35	0,4	0,45	11-14

Варианты 8.1 – 8.3

Автомобиль должен пройти 4 этапа длиной l_i (км) каждый. Известны зависимости скорости движения автомобиля V (км/ч) от расхода горючего q (л/км) для каждого этапа. Скорость движения на трех этапах ограничена сверху. Соответствующие данные приведены в табл. 12.

Необходимо определить скорости движения, обеспечивающие минимальный расход горючего при заданном времени пробега T (ч). Показать, как изменится решение при увеличении или уменьшении T на 10%.

Таблица 12

Вариант	Параметр	1-й этап	2-й этап	3-й этап	4-й этап	T , ч
8.1	l_i	100	200	150	250	8,0
	$V_i(q)$	$160(1-e^{-0,2q})$	$8q$	$120(1-e^{-0,1q})$	$30q-1,5q^2$	
	$V_{i \max}$	130	100	50	-	
8.2	l_i	120	180	250	200	8,5
	$V_i(q)$	$130(1-e^{-0,1q})$	$30q-1,5q^2$	$8q$	$160(1-e^{-0,2q})$	
	$V_{i \max}$	50	-	100	130	
8.3	l_i	150	200	200	180	9,5
	$V_i(q)$	$8q$	$170(1-e^{-2q})$	$110(1-e^{-0,1q})$	$30q-1,5q^2$	
	$V_{i \max}$	90	140	60	-	

Варианты 9.1 – 9.3

Характеристики оборудования зависят от его возраста t :

$r(t)$ - стоимость ежегодно производимой продукции;

$U(t)$ - годовые эксплуатационные затраты;

$S(t)$ - остаточная стоимость (выручка от продажи оборудования).

На начало планового периода из N лет работающее оборудование имеет возраст $t=t_0$. В начале любого года оборудование можно сохранить или продать и купить новое. Продолжительность замены много меньше года. Пусть τ - число лет с начала планового периода. Тогда при $\tau < 5$ оборудование может заменяться новым, такого же типа с характеристиками $r_1(t)$, $U_1(t)$, $S_1(t)$ и ценой P_1 , а при $\tau \geq 5$ может заменяться таким же либо другого типа с характеристиками $r_2(t)$, $U_2(t)$, $S_2(t)$ и ценой P_2 .

Необходимо определить оптимальную политику замены оборудования для $N=9$ и $t_0=0 \div 6$. Исходные данные приведены в табл.13.

Таблица 13

Вариант	Функции	t										S	P
		0	1	2	3	4	5	6	7	8	9		
9.1	$r_1(t)$	25	24	23	21	20	20	19	19	18	17	5	17
	$U_1(t)$	9	10	10	11	12	13	14	15	16	17		
9.2	$r_1(t)$	21	20	19	19	18	18	17	16	16	15	2	9
	$U_1(t)$	11	11	11	12	12	13	13	13	14	15		
9.3	$r_1(t)$	28	27	27	26	25	25	24	23	22	21	3	11
	$U_1(t)$	16	16	17	17	18	18	19	20	20	21		
Для всех	$r_2(t)$	30	28	28	27	25	25	23	21	20	19	7	25
	$U_2(t)$	10	12	12	14	14	15	16	17	19	22		

Варианты 10.1–10.3

Планируется деятельность объединения, включающего 2 предприятия, на 4 года. Известны функции дохода i -го предприятия в j -м году $g_{ij}(x_{ij})$, где x_{ij} - количество ресурса, вкладываемое в i -е предприятие в j -м году. Вложенные ресурсы за год уменьшаются до величины $\varphi_i(x)$.

Оставшийся в конце года ресурс возвращается в централизованный фонд объединения и снова полностью распределяется.

Определить оптимальное распределение ресурсов по предприятиям и годам, если начальный централизованный фонд ресурсов равен R , а зависимости имеют вид:

$$g_{1j} = k_{1j} \sqrt{x_{1j}}; \quad g_{2j} = k_{2j} (1 - e^{-\alpha_j x_{2j}}); \quad \varphi_i(x) = [\beta_i x].$$

Квадратные скобки означают взятие целого по правилам округления.

Значения коэффициентов и R даны в табл. 14, 15.

Таблица 14

j	k_{1j}	k_{2j}	j	k_{1j}	k_{2j}
1	4	7,2	3	3,5	6,5
2	3	7,8	4	5,1	7,3

Таблица 15

Вариант	α_1	α_2	α_3	α_4	β_1	β_2	R
10.1	0,1	0,15	0,2	0,2	0,3	0,2	19-21
10.2	0,12	0,17	0,25	0,35	0,26	0,22	17-19
10.3	0,15	0,17	0,22	0,3	0,4	0,3	13-17

Варианты 11.1 – 11.3

Пусть непилотируемый летательный аппарат, запускаемый с земли, должен за время T , кратное δt , достигнуть высоты H . Сигнал коррекции траектории поступает через интервалы δt и мгновенно отрабатывается. Между корректировками полет идет под одним углом к горизонту. Известны зависимости:

$$q = a_0 + a_1 \alpha^2 - a_2 h; \quad v = b_0 - b_1 \alpha,$$

где q - расход горючего, кг/с; h - высота аппарата относительно земли, м; v - скорость полета, км/ч; α - угол подъема (спуска), град;

Требуется найти оптимальную траекторию полета с точностью не хуже 4% от H и соответствующее ей (оптимальное) значение T ; построить графики траектории в координатах “высота - время” и “высота - расстояние по земле от точки старта”.

Значения параметров приведены в табл. 16.

Таблица 16

Вариант	H , м	δt , с	a_0 , кг/с	a_1 , кг/(с·град ²)	a_2 , кг/(с·м)	b_0 , км/ч	b_1 , км/(ч·град)
11.1	9600	10	12	10^{-2}	10^{-3}	3000	30
11.2	4800	5	12	10^{-2}	10^{-3}	3000	30
11.3	12000	20	12	10^{-2}	10^{-3}	3000	30

Варианты 12.1 – 12.3

На кондитерской фабрике организуется выпуск трех видов продукции из двух видов сырья. Известны нормы расхода сырья на изготовление единицы продукции и количество имеющегося сырья (табл. 17).

Зависимость прибыли от объема производства каждого вида продукции дана в табл. 18.

Требуется определить оптимальный план производства продукции.

Указание: использовать прием снижения размерности на основе множителей Лагранжа.

Таблица 17

Вид сырья	Расход на единицу продукции (по вариантам)			Кол-во сырья (по вариантам)
	1-й вид	2-й вид	3-й вид	
I	2; 5; 3	3; 1; 2	1; 2; 4	10; 9; 8
II	4; 2; 3	2; 4; 5	3; 2; 1	12; 10; 11

Таблица 18

Вид продукции	Вариант 12.1	Вариант 12.2	Вариант 12.3
I	$8\sqrt{x}$	$20(1-e^{-0,3x})$	$12x-x^2$
II	$1,2x^2$	$9x-x^2$	$7x$
III	$8x-x^2$	$3x$	$1,5x^2$

Варианты 13.1 – 13.3

Два вида средств в количестве P и Q , выделенных предприятию на квартал, необходимо распределить по месяцам. Прибыль за месяц зависит от величины использованных средств, как показано в табл. 19, где x и y - количества 1-го и 2-го вида средств, используемых в соответствующий месяц. На каждый месяц должно быть выделено не менее одной единицы первого вида средств.

Указание: использовать прием снижения размерности на основе множителей Лагранжа. При этом невязка по сумме средств не должна быть больше 3%.

Таблица 19

Вариант	1-й месяц	2-й месяц	3-й месяц	P	Q
13.1	$1,5x+10\sqrt{y}$	$5\sqrt{xy}$	$6\sqrt{x}+10y-y^2$	5	4
13.2	$5\sqrt{x}+2\sqrt{2y}$	$2x\sqrt{y}$	$0,5x(12y-2y^2)$	6	5
13.3	$10\sqrt{xy}$	$\sqrt{12x^3}+10\sqrt{y}$	$5\sqrt{x}+8y-y^2$	5	6

Варианты 19.1 – 19.3

Обработка информации осуществляется пятью последовательно включенными вычислительными устройствами (ВУ). Известна продолжительность однократного счета на каждом из ВУ - t_i (табл. 20).

Для повышения достоверности обработки применяется повторный счет на отдельных ВУ. Зависимость вероятности получения правильного результата от числа повторностей счета $P(\kappa)$ дана в аналитическом виде (табл. 21).

Определить вариант обработки информации, обеспечивающий вероятность получения правильного результата не хуже ψ за минимальное время счета.

Вывести функциональное уравнение для случая максимизации ψ при заданном общем времени счета.

Таблица 20

Вариант	t_1	t_2	t_3	t_4	t_5	ψ
19.1	7	2	5	3	4	0,7-0,9
19.2	3	8	6	2	7	0,8-0,95
19.3	5	1	9	8	6	0,9-0,97

Таблица 21

Вариант	$P_1(\kappa)$	$P_2(\kappa)$	$P_3(\kappa)$	$P_4(\kappa)$	$P_5(\kappa)$
19.1	$1 - 0,1^\kappa$	$1 - 0,5^\kappa$	$1 - 0,8^{2\kappa}$	$1 - e^{-\kappa}$	$1 - 0,2^\kappa$
19.2	$1 - 2 \cdot 0,6^\kappa + 0,8^{\kappa \cdot \kappa}$	$1 - e^{-2\kappa}$	$1 - 0,1^\kappa$	$1 - 0,2^\kappa$	$1 - 0,1^{\sqrt{\kappa}}$
19.3	$1 - 0,5^{\kappa \cdot \kappa}$	$1 - 0,2^\kappa$	$1 - e^{-1,5 \kappa}$	$1 - 4 \cdot 0,1^\kappa$	$1 - e^{-\kappa}$

Варианты 15.1 – 15.3

Проектируется строительство дороги с четырьмя перегонами (табл. 22). Капитальные (приведенные) C_i^k и эксплуатационные $C_i^э$ затраты зависят от длины перегона l_i : с уменьшением l_i первые возрастают из-за увеличения объема земляных работ, а вторые снижаются.

Спроектировать дорогу общей длиной не более L с минимальными затратами.

Записать функциональное уравнение для случая минимизации длины при заданном уровне затрат.

Примечание: для упрощения расчетов считать, что эксплуатационные затраты даны на весь период эксплуатации дороги.

Таблица 22

Вариант	Перегоны				L , км
	1	2	3	4	
	$C^k = 200 / \sqrt{l}$	$C^k = 50 + 400/l$	$C^k = 120 - 3l$	$C^k = 20l - 0,5l^2$	40-50
15.1	$C^э = 40 + 2l$	$C^э = 60(1 - e^{-0,1l})$	$C^э = 70 + 8\sqrt{l}$	$C^э = 40\sqrt{l}$	
	$5 \leq l \leq 10$	$10 \leq l \leq 16$	$4 \leq l \leq 10$	$20 \leq l \leq 26$	

Ва- риант	Перегоны				L, км
	1	2	3	4	
15.2	$C^K=180-2l$	$C^K=580/\sqrt{l}$	$C^K=630/l$	$C^K=120/(1-e^{-0,1l})$	60-70
	$C^9=90(1-e^{-0,2l})$	$C^9=110+2l$	$C^9=80+9\sqrt{l}$	$C^9=40+2e^{\sqrt{l}-6}$	
	$10 \leq l \leq 16$	$21 \leq l \leq 27$	$14 \leq l \leq 20$	$7 \leq l \leq 12$	
15.3	$C^K=20+15l-0,5l^2$	$C^K=200/(2+0,1l)$	$C^K=70+350/l$	$C^K=170-10\sqrt{l}$	55-65
	$C^9=240l/(l+36)$	$C^9=30+12\sqrt{l}$	$C^9=30+5l$	$C^9=130(1-e^{-0,1l})$	
	$18 \leq l \leq 25$	$10 \leq l \leq 16$	$8 \leq l \leq 14$	$16 \leq l \leq 23$	

Варианты 16.1 – 16.3

На период в T дней известен объем погрузочно-разгрузочных работ, выражаемый в ежедневной потребности в рабочих - $d_t, t=1,...,T$ (табл. 23). Рабочих можно ежедневно нанимать и увольнять. При нехватке рабочих прибегают к сверхурочным работам и затраты возрастают на C_1 за каждого недостающего рабочего, расходы на содержание одного незанятого рабочего составляют C_2 , а на найм одного рабочего - C_3 . Увольнение требует расходов C_4 .

Составить оптимальный план регулирования численности рабочих на T дней, если исходное количество рабочих равно R .

Таблица 23

Вариант	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	R	C_1	C_2	C_3	C_4
16.1	8	5	3	10	7	11	13	-	5-10	20	10	14	6
16.2	12	10	8	10	6	8	9	12	10-13	25	13	17	8
16.3	10	12	13	9	10	11	12	-	9-12	40	15	20	10

Варианты 17.1 – 17.3

Пусть непилотируемый летательный аппарат, запускаемый с земли, должен за время T , кратное δt , достигнуть высоты H . Сигнал коррекции траектории поступает через интервалы δt и мгновенно отрабатывается. Между корректировками полет идет под одним углом к горизонту. Известны зависимости:

$$q=a_0+a_1\alpha^2-a_2h, \quad v=b_0-b_1\alpha,$$

где q - расход горючего, кг/с; h - высота аппарата относительно земли, м; v - скорость полета, км/ч; α - угол подъема (спуска), град.

Значения всех величин приведены в табл. 24

Таблица 24

Вариант	$H, \text{м}$	$T, \text{с}$	$\delta t, \text{с}$	$a_0, \text{кг/с}$	$a_1, \text{кг/(с·град)}^2$	$a_2, \text{кг/(с·м)}$	$v_0, \text{км/ч}$	$v_1, \text{км/(ч·град)}$
17.1	6000	40	10	12	10^{-2}	10^{-3}	3000	30
17.2	8000	40	10	12	10^{-2}	10^{-3}	3000	30
17.3	10000	60	15	12	10^{-2}	10^{-3}	3000	40

Определить оптимальную траекторию полета с точностью не хуже 4% от H , построить графики траекторий в координатах “высота - время” и “высота - расстояние по земле от точки старта”.

Варианты 18.1 – 18.3

Для восстановления дороги проводятся текущие и капитальные ремонты. Ремонт может производиться не более 1 раза в год и только в начале года (длительность ремонта много меньше года). Текущий ремонт требует C_T затрат, а капитальный - C_K (табл. 25). После текущего ремонта годовые эксплуатационные затраты равны $C_3^T(t)$, а после капитального - $C_3^K(t)$, где t - число лет, прошедшее после ремонта на начало рассматриваемого года.

Определить оптимальный план ремонта дороги на T лет, если на начало этого периода прошло $t_{T(K)}^H$ лет после текущего (капитального) ремонта.

Таблица 25

Вариант	C_T	C_K	$C_3^T(t)$	$C_3^K(t)$	T	t^H
18.1	8	13	$9+2t$	$6+t$	11	$t_T^H = 0...5$
18.2	10	15	$10 + t\sqrt{t}$	$7+1,1t$	10	$t_K^H = 0...7$
18.3	13	18	$8+0,5t^2$	$5 + t\sqrt{t}$	10	$t_K^H = 3...8$

Варианты 19.1 – 19.3

На космическом корабле действует N различных приборов. Прибор типа i весит w_i кг, а суммарный вес приборов не должен превышать W кг. Известны вероятности $P_i(t)$, с которыми i -й прибор может функционировать в течение t единиц времени (табл. 26). Как только запас какого-либо типа прибора станет равным нулю, космический корабль должен вернуться на Землю. Число приборов и их максимальный суммарный вес даны в табл. 27.

Определить запасы приборов, обеспечивающие наибольшее

математическое ожидание времени полета космического корабля.

Таблица 26

t	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
1	0,05	0,1	0,2	0,15	0,03	0,3	0	0,4	0,25
2	0,1	0,15	0,3	0,2	0,07	0,4	0,1	0,3	0,2
3	0,2	0,3	0,2	0,4	0,1	0,2	0,2	0,15	0,2
4	0,3	0,2	0,15	0,2	0,2	0,06	0,4	0,1	0,15
5	0,2	0,15	0,1	0,05	0,4	0,04	0,2	0,05	0,1
6	0,15	0,1	0,05	0	0,2	0	0,1	0	0,1
w_i	2,5	3	4,2	1,5	2,7	3,8	5,6	6,5	5

Таблица 27

Вариант	W , кг	N	Исключаемые типы приборов
19.1	56-66	7	3,5
19.2	60-73	6	2,6,9
19.3	45-58	7	7,8

Варианты 20.1 – 20.3

Продукты доставляются на грузовике с полезным объемом V . Потребность в I видах продуктов описывается непрерывным равномерным распределением с плотностью

$$P_i(m_i) = \begin{cases} 1/b_i, & 0 \leq m_i \leq b_i, \\ 0, & \text{иначе;} \end{cases} \quad i = 1, 2, 3, \dots, I,$$

где m_i - масса i -го продукта.

Единица массы продукта i занимает объем v_i (табл. 28). Определить, как загрузить грузовик, чтобы минимизировать математическое ожидание неудовлетворенного за рейс спроса. Как изменится решение, если потребовать минимизации нереализованной стоимости при известной цене продуктов C_i , руб/кг.

$$C_1=1,5; \quad C_2=3,2; \quad C_3=0,8; \quad C_4=2,5.$$

Таблица 28

Вариант	V	b_1	b_2	b_3	b_4	v_1	v_2	v_3	v_4
20.1	21,3	10	11	17	7	2,5	4	1,5	3,6
20.2	15,75	10	14	15	18	1	5/3	3	10,3
20.3	23,8	7	10	12	15	2,0	1,2	1,8	4,5

Примечание: рассматривать только целые значения массы продукта.

Варианты 21.1 – 21.3

Пять предприятий получают сырье с двух складов. Затраты на перевозку единицы груза от склада к предприятию зависят от количества перевозимого груза:

$$C_{ij}(x) = C_{ij}^0 + C_{ij}^1 \sqrt{x}.$$

Пусть b_j - потребность j -го предприятия в сырье, a_i - количество сырья на i -м складе, при этом общий запас на складах равен суммарной потребности предприятий.

Определить оптимальный план перевозок сырья для данных, приведенных в табл. 29.

Таблица 29

Вариант	C_{ij}^0	C_{ij}^1	a_1	a_2	b_1	b_2	b_3	b_4	b_5
21.1	$(i+j)/i$	$i/(i+j)$	20	12	5	7	10	4	6
21.2	$(i+j)/j$	$j/(i+j)$	15	18	6	9	5	6	7
21.3	$(3+j)/j$	$j/(3+i)$	10	22	8	5	3	10	6

Варианты 22.1 – 22.3

Прокладывается автомобильная дорога, которая должна связать пять пунктов. Длина каждого из 4 участков дороги L_i ограничена сверху и снизу. Капитальные C_i^k и эксплуатационные C_i^e затраты зависят от длины участка: с увеличением L_i первые уменьшаются (сокращается объем земляных и мостостроительных работ), а вторые возрастают (табл. 30).

Определить вариант дороги, обеспечивающий общую минимальную длину в пределах выделенных средств S . Записать функциональное уравнение для случая минимизации средств при заданной длине.

Таблица 30

№	1	2	3	4	S
22.1	$C^k=580/\sqrt{L}$	$C^k=630/L$	$C^k=180-2L$	$C^k=120/(1-e^{-0,1L})$	890-900
	$C^e=110+2L$	$C^e=40+2e^{\sqrt{L-6}}$	$C^e=90(1-e^{-0,2L})$	$C^e=80+9\sqrt{L}$	
	$21 \leq L \leq 27$	$7 \leq L \leq 12$	$10 \leq L \leq 16$	$14 \leq L \leq 20$	
22.2	$C^k=170-10\sqrt{L}$	$C^k=20+15L-0,5L^2$	$C^k=200/(2+0,1L)$	$C^k=70+350/L$	750-760
	$C^e=130(1-e^{-0,1L})$	$C^e=240L/(L+36)$	$C^e=30+12\sqrt{L}$	$C^e=30+5L$	
	$16 \leq L \leq 23$	$18 \leq L \leq 25$	$10 \leq L \leq 16$	$8 \leq L \leq 14$	
22.3	$C^k=120-3L$	$C^k=20L-0,5L^2$	$C^k=200/\sqrt{L}$	$C^k=50+400/L$	820-830
	$C^e=70+8\sqrt{L}$	$C^e=40\sqrt{L}$	$C^e=40+2L$	$C^e=60(1-e^{-0,1L})$	
	$4 \leq L \leq 10$	$20 \leq L \leq 26$	$5 \leq L \leq 10$	$10 \leq L \leq 16$	

Примечание: для упрощения расчетов считать, что эксплуатационные

затраты даны на весь период эксплуатации дороги.

Варианты 23.1 – 23.3

Пусть непилотируемый летательный аппарат, запускаемый с земли, должен за время T , кратное δt , достигнуть высоты H . Сигнал коррекции траектории поступает через интервалы δt и мгновенно отрабатывается. Между корректировками полет идет под одним углом к горизонту. Известны зависимости:

$$q = a_0 + a_1 \alpha^2 - a_2 h, \quad v = b_0 - b_1 \alpha,$$

где q - расход горючего, кг/с; h - высота аппарата относительно земли, м; v - скорость полета, км/ч; α - угол подъема (спуска), град.

Кроме того, известны стоимость горючего C_r и затраты на одну корректировку C_k (табл. 31).

Таблица 31

Вариант	H , м	T , с	a_0 , кг/с	a_1 , кг/(с·град ²)	a_2 , кг/(с·м)	b_0 , км/ч	b_1 , км/(ч·град)	C_r , руб/кг	C_k , руб/к
23.1	7000	36	12	10^{-2}	10^{-3}	3000	30	0,5	3
23.2	6500	36	12	10^{-2}	10^{-3}	3500	40	0,5	2
23.3	6000	36	12	10^{-2}	10^{-3}	3500	40	0,5	2

Требуется найти оптимальные траектории полета при 1, 2 и 3 корректировках за полет, построить графики траекторий в координатах “высота–время” и “высота–расстояние по земле от точки старта”, определить наилучшее число корректировок среди указанных значений. Точность определения траектории - не хуже 4% от H .

Варианты 24.1 – 24.3

Самолет имеет максимальную грузоподъемность $G(m)$. Предметы, загружаемые в самолет, имеют различные веса и стоимости: $w_i(m)$, C_i (руб.) соответственно (табл. 32).

Определить оптимальную загрузку самолетов с грузоподъемностью от G до $1,3G$. Показать, как изменится решение, если j -х предметов нужно погрузить не меньше k .

Таблица 32

№	G	W_1	C_1	W_2	C_2	W_3	C_3	W_4	C_4	W_5	C_5	W_6	C_6	j	k
24.1	20	0,8	12	3,2	60	4,0	72	6,5	11	2,8	50	2,1	15	1; 6	2
24.2	12	1,1	9,2	2,3	15	0,7	5,5	3,1	12	1,5	11	0,5	4	2; 4	1
24.3	15	2	4,5	1,2	3	0,6	2	1,5	5	3,0	4,2	1,1	3	3; 5	2

Варианты 25.1 – 25.3

На обработку информации выделено T единиц времени. Обработка производится пятью последовательно включенными вычислительными устройствами (ВУ). Известна продолжительность однократного счета на каждом из ВУ - t_i (табл. 33). Для повышения достоверности обработки применяется повторный счет на отдельных ВУ. Зависимость вероятности искажения информации от числа повторностей счета $P_i(k)$ приведена в табл. 34.

Определить оптимальный вариант обработки информации.

Вывести функциональное уравнение для случая, когда задана общая допустимая вероятность искажения и необходимо минимизировать время обработки.

Таблица 33

Вариант	t_1	t_2	t_3	t_4	t_5	T, c
25.1	2	1	5	3	7	45-50
25.2	6	3	1	8	4	52-60
25.3	4	2	7	3	5	50-55

Таблица 34

k	$P_1(k)$	$P_2(k)$	$P_3(k)$	$P_4(k)$	$P_5(k)$
1	0,1	0,07	0,08	0,1	0,12
2	0,05	0,03	0,04	0,07	0,08
3	0,02	0,015	0,02	0,05	0,06
4	0,01	0,01	0,012	0,02	0,03
5	0,004	0,002	0,005	0,01	0,012
6	0,001	0,0005	0,002	0,003	0,005

Варианты 26.1 – 26.3

Потерпевший аварию самолет может находиться в одном из n районов с вероятностями p_1, \dots, p_n . Если на поиск в i -м районе

затрачивается t единиц времени, то вероятность обнаружения самолета равна $1 - e^{-\alpha t}$ (если самолет окажется в данном районе).

Определить оптимальную стратегию поиска при условии ограничения времени поиска величиной T . Необходимые данные приведены в табл. 35.

Таблица 35

Вариант	Параметры	Район					T
		1	2	3	4	5	
26.1	P_i	0,1	0,4	0,2	0,1	0,2	20-25
	α_i	0,2	0,1	0,5	0,4	1,0	
26.2	P_i	0,25	0,1	0,3	0,2	0,15	3-6
	α_i	2	1	0,5	2,5	0,4	
26.3	P_i	0,3	0,1	0,2	0,15	0,25	15-20
	α_i	0,1	0,2	0,5	0,4	0,3	

Варианты 27.1 – 27.2

Отрасль выпускает n типов станков. Станок типа L выполняет L -ю, $L+1$ -ю, ..., n -ю операции. Пусть b_L - потребность в станках, выполняющих L -ю операцию, $L=1,2,3,\dots,n$, а $C_L(x)$ - функция стоимости x станков типа L .

Определить оптимальный вариант приобретения станков, если $C_L(x) = \alpha_L x^\beta$. Все данные приведены в табл. 36.

Таблица 36

Вариант	b_1	b_2	b_3	b_4	b_5	α_1	α_2	α_3	α_4	α_5	β
27.1	1	2	4	3	6	2000	1800	1500	1100	800	0,85
27.2	2	1	3	6	4	900	820	700	500	280	0,7

Варианты 28.1 – 28.3

Для уничтожения n целей имеется m ракет. Значимость i -й цели оценивается величиной C_i , а вероятность поражения цели при использовании одной ракеты равна P_i (табл. 37).

Определить оптимальное распределение ракет по целям.

Таблица 37

Вариант	m	C_1	C_2	C_3	C_4	C_5	P_1	P_2	P_3	P_4	P_5
28.1	9-15	10	5	15	8	3	0,5	0,2	0,3	0,4	0,6
28.2	10-16	25	15	20	10	12	0,2	0,3	0,4	0,4	0,5
28.2	12-17	3	8	6	11	9	0,4	0,3	0,5	0,2	0,3

Варианты 29.1 – 29.3

Для оснащения нового печатного цеха, имеющего производственную площадь S (м²), выделено B млн. руб. Можно приобретать станки четырех типов с одинаковыми функциями. Станок i -го типа стоит b_i млн. руб., занимает площадь S_i (м²) и имеет производительность m_i (табл. 38).

Определить оптимальный вариант оснащения цеха.

Таблица 38

Вариант	B	S	b_1	b_2	b_3	b_4	S_1	S_2	S_3	S_4	m_1	m_2	m_3	m_4
29.1	25	265	5,2	0,8	2,3	1,5	90	20	70	50	4,5	1,2	3	2
29.2	12	135	1,6	0,9	1	0,5	40	15	30	10	1,5	0,7	1,2	0,4
29.3	19	205	2	1,5	3	2,4	52	35	20	42	2,1	1,8	2,7	2,2

Варианты 30.1 – 30.2

Ежемесячно требуется поставлять со склада a_i деталей i -го типа стоимостью C_i . Общая стоимость деталей, хранимых на складе, не должна превышать величину C (табл. 39).

Определить количество хранимых деталей каждого типа, наименее уклоняющееся от потребности в них. Решить для двух показателей (раздельно), характеризующих меру уклонения.

Таблица 39

Вариант	a_1	a_2	a_3	a_4	a_5	C_1	C_2	C_3	C_4	C_5	C
30.1	6	1	7	3	5	80	100	150	70	60	1600
30.2	10	3	5	8	-	50	80	60	120	-	1800

Глава 10. МНОГОКРИТЕРИАЛЬНЫЕ ЗАДАЧИ ПРИНЯТИЯ РЕШЕНИЙ

10.1. Основы многокритериальной оптимизации

Вся сознательная жизнь человека связана с принятием решений. Одни решения касаются только самого принимающего решения, другие относятся к небольшому кругу людей, третьи затрагивают интересы целой организации, региона и даже страны. Чем выше уровень, тем серьезнее могут быть последствия, тем выше ответственность принимающих решения. Усложнение ситуаций, в которых приходится принимать решения, вызвало потребность в научной поддержке, что привело к развитию нового подхода, получившего название исследование операций. Однако до начала семидесятых годов в рамках исследования операций рассматривались в основном задачи, в которых эффективность решения оценивалась одним критерием. В то время считалось, что требования, предъявляемые к решению, можно выразить одним показателем качества. Методы математического программирования, интенсивно развиваемые в исследовании операций, изначально ориентировались на решение однокритериальных задач.

Со временем росло понимание неадекватности такого подхода реальным процессам принятия решений. Все яснее становилась необходимость учитывать существование более одного показателя эффективности, оптимальные решения по которым не совпадают. С этого периода началось бурное развитие многокритериальных методов принятия решений и, в частности, методов многокритериального математического программирования.

Многокритериальность может быть обусловлена одной из трех причин:

1. Цель не может быть адекватно представлена (покрыта) одним критерием.
2. Принимающий решения ставит более одной цели, которые связаны общими активными средствами.
3. Решения принимаются группой лиц с несовпадающими интересами.

Так для характеристики цели «Повысить уровень жизни народа» требуется целый ряд показателей. При выборе номенклатуры и количества выпускаемых изделий начинающая фирма может преследовать как тактическую цель – получение высокой прибыли в ближайшее время, так и стратегическую – закрепление на рынке сбыта и его расширение. В качестве примера третьей ситуации можно привести переговоры России и Казахстана по космодруму Байконур.

В этой главе изложение затронутых проблем будет ограничено в основном многокритериальными задачами математического программирования. Естественно, что круг задач принятия решений при многих критериях существенно шире.

10.1.1. Многокритериальная задача математического программирования

В формальном представлении критерии (целевые функции), по которым оценивается решение X , будут записываться в виде $f_i(X)$, $i = \overline{1, m}$. Критерий f_i называют также частными. Для удобства рассуждений примем, что для всех i чем больше значение критерия, тем лучше. Тогда задача многокритериального математического программирования запишется в виде:

$$\begin{aligned} & \max \{f_1(X)=y_1\}, \\ & \max \{f_2(X)=y_2\}, \\ & \dots\dots\dots \\ & \max \{f_m(X)=y_m\}, \\ & X \in D, \end{aligned}$$

где D – множество допустимых решений. Иначе говоря, задача состоит в максимизации вектора критериев $\mathbf{f}(X)=Y$ по $X \in D$.

Существенное отличие этой задачи от традиционной однокритериальной состоит в понятии оптимальности. В однокритериальной задаче под оптимальным понимается решение, обеспечивающее максимальное значение критерия. При многих критериях увеличение одних критериев приводит к уменьшению других (редкие исключения не представляют практического интереса) и поэтому понятие оптимальности требует принципиальных уточнений. Очевидно, что без дополнительной информации о предпочтениях ЛПР бессмысленно говорить об оптимальном решении и тем более формализованно искать его.

Допустимое множество D строится в n -мерном пространстве переменных. Каждое решение $X \in D$ полностью характеризуется соответствующими значениями всех частных критериев, т.е. вектором Y . Числовое m -мерное пространство E^m , координатами которого являются $y_i=f_i(X)$, называется **критериальным пространством**. Очевидно, что каждому X можно поставить в соответствие точку в критериальном пространстве. Если же решение X допустимо, то соответствующая точка в E^m , определяемая вектором Y , является **достижимой**. Множество таких точек в критериальном пространстве называется множеством достижимости (достижимых векторов). Таким образом, векторная функция $f(X)$ отображает допустимое множество D на множестве достижимости G :

$$G = \{Y \in E^m \mid Y = \mathbf{f}(X), X \in D\}$$

и задача состоит в выборе вектора из этого множества, наилучшего с точки зрения ЛПР.

В общем случае построение множества G для реальных задач весьма проблематично, но для задач с «хорошими» свойствами, например, линейных, множество достижимости может быть построено.

10.1.2. Где искать оптимальное решение

Как отмечалось выше, без установления принципа оптимальности, отражающего предпочтение ЛПР, невозможно формально распознать оптимальное решение (как в сказке: "ищи то, не знаю, что"). Однако учитывая стремление ЛПР к увеличению значений всех частных критериев, можно формальными методами исключить из множества G (и D) заведомо не перспективные точки и тем самым облегчить решение задачи.

Для наглядности рассуждений рассмотрим пример с двумя критериями (рис.10.1). Независимо от предпочтений ЛПР, вектор

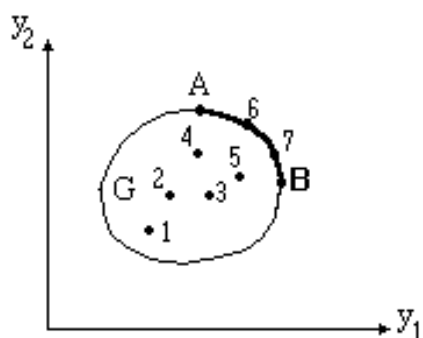


Рис. 10.1

критериев, соответствующий точке 2, лучше, чем в точке 1. Аналогично, точка 3 лучше точки 2, а 4 лучше 3. Но точки 4 и 5 оказываются не сравнимыми, так как по первому критерию лучше точка 5, а по второму – точка 4. Как для точки 5, так и для 4 на множестве G можно найти лучшую точку, например 6. Нетрудно убедиться в том, что для любой точки Y внутри G найдется точка, которая ее доминирует, т.е. лучше хотя бы по одному частному критерию и не хуже по всем другим. В то же время для точек 6 или 7 этого сделать нельзя. Более того, не найдется вектора из G , который доминировал бы точку, принадлежащую северо-восточной границе AB множества G . Таким образом, векторы на AB являются недоминируемыми (неулучшаемыми). Одновременно они являются несравнимыми между собой (например, в точках 6 и 7), поэтому отдать предпочтение одному из них без ЛПР невозможно. Такие точки (векторы критериев и соответствующие решения) называют эффективными или оптимальными по Парето. Их совокупность образует множество Парето (паретовское множество).

Это наименование произошло от фамилии итальянского экономиста и социолога В.Парето (1848-1923), который проводил математические исследования процесса рыночного обмена товаров. Рассматривалась модель чистого обмена, в которой каждый участник стремится составить себе набор товаров наибольшей ценности. Эффективным является такое состояние, которое не может быть улучшено путем перераспределения товаров ни для одного из участников без ущемления интересов некоторых других участников. Значит, эффективное состояние соответствует экономическому

равновесию, а неэффективное состояние побуждает проводить перераспределение (торговать), которое ведет к установлению равновесия.

Теперь очевидно, что оптимальное решение следует искать только среди эффективных точек. При групповом принятии решений множество эффективных точек называют также переговорным, подчеркивая тем самым, что только их и нужно рассматривать в качестве претендентов на компромиссное решение. Если эффективная точка одна (А на рис.10.2), что возможно в тривиальном случае непротиворечивости критериев, то она и является искомым оптимумом. В задачах с конечным числом точек G (дискретные задачи) выделение эффективного множества часто настолько уменьшает число вариантов, что выбор из них наилучшего не вызывает затруднений у ЛПР.

Однако при непрерывном и тем более невыпуклом множестве G паретовское множество имеет сложную структуру и его исследование требует специальных методов

Ввиду особой важности парето-оптимальности для решения многокритериальных задач приведем более строгие определения, связанные с этим понятием.

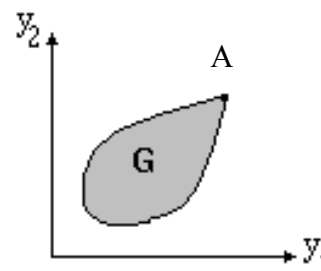


Рис. 10.2

10.1.3. Определения

Для описания предпочтений используют бинарные отношения, вводимые на множестве A сравниваемых объектов. В многокритериальной задаче роль таких объектов играют X или Y на множествах D и G соответственно.

Если из двух объектов a и b ЛПР выбирает a , то говорят, что a предпочтительнее b . Все пары вида (a,b) , где $a,b \in A$, для которых a предпочтительнее b , образуют множество, называемое отношением строгого предпочтения на A . Такое отношение обозначают символом \succ ($a \succ b$ или aPb , где P – первая буква английского слова preference – предпочтение).

Объекты a и b неразличимы для ЛПР, если они одинаковы по предпочтительности. Это значит, что не выполняется ни отношение $a \succ b$, ни $b \succ a$. Множество всех неразличимых пар (a,b) называют отношением неразличимости или безразличия и обозначают символом \sim ($a \sim b$ или aIb , где I происходит от indifference – безразличие).

Очевидно, что для любой пары $a,b \in A$ выполняется только одно из трех соотношений: $a \succ b$, $b \succ a$, $a \sim b$. Объединение P и I дает отношение нестрогого предпочтения, обозначаемого символом \succsim ($a \succsim b$ или aRb). Отношение $a \succsim b$ означает, что a не менее предпочтительно, чем b .

В соответствии с этими определениями решение $X^* \in D$ (вектор $Y^* \in G$) называют оптимальным по отношению \succsim на множестве D (G), если не существует другого решения $X \in D$ (вектора $Y \in G$), для которого справедливо соотношение $X \succ X^*$ ($Y \succ Y^*$). Если для любых $X \in D$ ($Y \in G$) выполняется соотношение $X^* \succsim X$ ($Y^* \succsim Y$), то $X^* \in D$ ($Y^* \in G$) называется оптимальным решением (вектором) по отношению \succsim .

При сравнении по предпочтительности векторов $Y=f(X)$ наиболее просто сопоставлять те вектора, которые отличаются лишь одной компонентой. Однако в общем случае частные критерии $y_i=f_i(X)$ могут по-разному соотноситься по предпочтительности в зависимости от того, на каких уровнях зафиксированы остальные критерии. Так если вектор $(\alpha, y_2', \dots, y_m')$ предпочтительнее вектора $(\beta, y_2', \dots, y_m')$, а вектор $(\alpha, y_2'', \dots, y_m'')$ менее предпочтителен, чем $(\beta, y_2'', \dots, y_m'')$, то какое из значений первого критерия, α или β , предпочтительнее сказать нельзя без знания значений остальных критериев. Так, например, чем выше потолок комнаты, тем лучше, но справедливо это до определенных соотношений высоты, ширины и длины комнаты. Чаше, однако, все значения частного критерия можно упорядочить по предпочтению без учета значений других критериев. Такие критерии называют независимыми по предпочтению от остальных. Примерами могут служить прибыль, издержки и т.п.

Задачи, в которых все критерии независимы по предпочтению, а отношением строгого предпочтения R является отношение \geq (не меньше) называются многокритериальными задачами максимизации (аналогично при отношении «не больше» – задачами минимизации).

Напомним, что R включает (объединяет) P и I . На множестве G (или D) отношение строгого порядка P задают неравенством $Y \geq Y'$ (т.е. $Y \geq Y'$ и $Y \neq Y'$) или $Y > Y'$ (т.е. $y_i > y_i'$ для $\forall i$). Наконец, равенство $=$ порождает отношение безразличия.

Вектор (решение), оптимальный по отношению \geq на множестве G (D), называется эффективным или парето-оптимальным. Значит, вектор $Y^* \in G$ является парето-оптимальным (оптимумом Парето), если не существует вектор $Y \in G$ такой, что $Y \geq Y^*$. Множество таких векторов обозначают через $P(Y)$ и называют множеством Парето (эффективным множеством). Множество эффективных решений обозначают через $P(X)$.

Вектор, оптимальный по отношению $>$, называют слабо эффективным, слабо оптимальным по Парето (слабым оптимумом Парето). Значит, вектор $Y^* \in G$ слабо парето оптимальный, если не существует $Y \in G$ такой, что $Y > Y^*$. Множество таких векторов называют слабо эффективным и обозначают через $S(Y)$. Соответствующее множество слабо эффективных решений имеет обозначение $S(X)$. Если в G не найдётся $Y \geq Y^*$, то не существует и $Y > Y^*$. Следовательно, всякий эффективный вектор

одновременно является и слабо эффективным, т.е. $P(Y) \subseteq S(Y)$. Аналогично $P(X) \subseteq S(X)$.

Различие эффективного и слабо эффективного множеств хорошо видно на рис.10.3. Множество $P(Y)$ состоит из частей границы множества G : кривых bc , de (исключая точки d и e) и gh , а $S(Y)$ – из кривой $abcde$ (включая точку e) и кривой ghk . Точка d не входит в $P(Y)$, т.к. она доминируется точкой c . Точно также точка e менее предпочтительна, чем g .

Геометрическое определение множеств $P(Y)$ и $S(Y)$ основано на том, что все точки $Y \in E^m$, для которых выполняется неравенство $Y \geq Y^0$, образуют ортант (для $m=2$ – прямой угол), стороны которого параллельны координатным осям, а вершиной является точка Y^0 .

Поэтому, если весь угол (ортант), построенный на некоторой точке $Y^* \in G$, расположен вне множества G , то Y^* парето-оптимальна. Если кроме вершины Y^* пересечение ортанта и G содержит только точки, лежащие на одной из сторон ортанта, то Y^* слабо парето-оптимальна, при этом $Y^* \notin P(Y)$, т.е. не является эффективной.

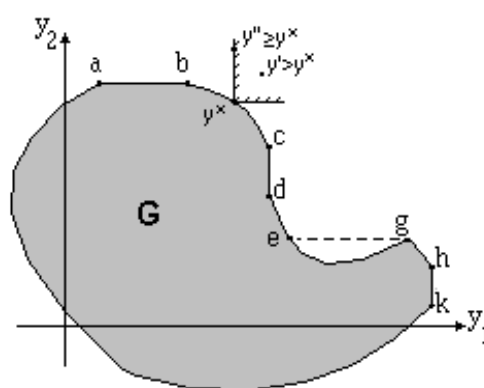


Рис. 10.3

Понятие слабой эффективности оказывается полезным и в случае, когда приходится сокращать первоначальный набор критериев. Нередко на первых этапах исследования трудно определить минимально необходимый набор критериев и поэтому начинают с возможно более полного набора. По мере изучения свойств задачи выявляются несущественные критерии, которые исключаются из дальнейшего рассмотрения. В [30] показано, что множество слабо эффективных решений, выделяемое на полном наборе критериев, содержит все исходные решения, эффективные по сокращенному набору критериев.

10.1.4. Условия оптимальности

Здесь рассмотрим наиболее важные с точки зрения приложений необходимые и достаточные условия оптимальности. Они позволяют строить методы отыскания эффективных решений и способы проверки эффективности найденных решений.

Наиболее общий случай необходимых условий содержит следующая теорема.

Теорема 1. Пусть $Y^* \in G$ и все $y_i > 0$. Вектор Y^* слабо эффективен тогда и только тогда, когда найдутся такие числа $\mu_i > 0, i = \overline{1, m}, \sum_{i=1}^m \mu_i = 1$, что

$$\min_i \mu_i y_i^* = \max_{Y \in G} \min_i \mu_i y_i. \quad (10.1)$$

Условие $y_i > 0$ не ограничивает применимость теоремы, так как его всегда можно обеспечить добавлением к $f_i(X)$ положительной константы $C > \max |f_i(X)|$.

При оговариваемых свойствах D и $f(X)$ справедливы теоремы 2 и 3.

Теорема 2. Пусть D выпукло, а $f_i, i = \overline{1, m}$, вогнуты и положительны на D . Тогда решение X^* слабо эффективно в том и только в том случае, если существуют такие числа $\mu_i \geq 0, i = \overline{1, m}, \sum_{i=1}^m \mu_i = 1$, что

$$\prod_{i=1}^m [f_i(X^*)]^{\mu_i} = \max_{X \in D} \prod_{i=1}^m [f_i(X)]^{\mu_i}. \quad (10.2)$$

Теорема 3. Пусть D выпукло, а f вогнуто. Для слабой эффективности точки $X^* \in D$ необходимо и достаточно, чтобы существовали числа $\mu_i \geq 0, i = \overline{1, m}, \sum_{i=1}^m \mu_i = 1$, при которых

$$\sum \mu_i f_i(X^*) = \max_{X \in D} \sum \mu_i f_i(X). \quad (10.3)$$

Требование вогнутости f существенно, так как его невыполнение может привести к тому, что не для всех слабо эффективных решений найдутся μ_i , удовлетворяющие (10.3). Например, для критериев $f_1(x) = x$ и $f_2(x) = e^{-x}$ (f_2 выпукла) на $D=[0,1]$ множество $S(X)=D$. Максимум функции $\mu_1 x + \mu_2 e^{-x}$ достигается только на одном из концов интервала $[0,1]$ и поэтому ни при каких неотрицательных μ_1 и μ_2 максимизация этой функции не даст слабо оптимальную точку, лежащую внутри D .

Терема 4. Вектор $Y^* \in G$ эффективен тогда и только тогда, когда для каждого $i = \overline{1, m}$

$$y_i^* = \max_{Y \in G_i} y_i, \quad (10.4)$$

где

$$G_i = \{Y \in G \mid y_j \not\geq y_j^*, j = \overline{1, m}, j \neq i\}. \quad (10.5)$$

Если $Y^* \in G$ эффективна, то она является единственной в G точкой, удовлетворяющей (10.4) при каждом $i = \overline{1, m}$.

Достаточные условия, приведенные ниже, основаны на свойствах возрастающей функции многих переменных. Поэтому сначала дадим определение такой функции. Числовая функция $F(Y)$, определённая на

множестве G , является возрастающей по отношению \geq , если из выполнения неравенства $Y \geq Y'$ для векторов $Y, Y' \in G$ всегда следует справедливость неравенства $F(Y) > F(Y')$. Аналогично, $F(Y)$ – функция, возрастающая по отношению $>$, если из $Y > Y'$ всегда следует $F(Y) > F(Y')$.

Теорема 5. Пусть функция $F(Y)$ определена на множестве G . Для того чтобы точка $Y^* \in G$ была эффективной (слабо эффективной), достаточно, чтобы она являлась точкой максимума на множестве G функции $F(Y)$, возрастающей по отношению \geq (по отношению $>$).

Теорема легко доказывается от противного. Пусть $Y^* \in G$ и

$$F(Y^*) \geq F(Y) \text{ для всех } Y \in G. \quad (10.6)$$

Предположим противное, т.е. что существует $Y' \in G$, для которого верно неравенство $Y' \geq Y^*$. Так как функция F возрастающая по отношению \geq , то противоречит (10.6). Аналогично доказываются достаточные условия слабой эффективности.

Теорема 5 играет важную роль в решении многокритериальных задач. Её применение основано на максимизации возрастающих функций многих переменных. Поэтому целесообразно рассмотреть примеры таких функций.

1). Функция $F(Y) = \sum_{i=1}^m \mu_i y_i$, где $\mu_i > 0$, является возрастающей по каждой переменной y_i на числовой оси и потому возрастает по \geq на E^m . Поэтому любая точка максимума $F(Y)$ на G эффективна. Эта же функция при $\mu_i \geq 0$ и хотя бы одном из них положительном является возрастающей по отношению $>$ и, значит, максимизация такой функции на G дает слабо эффективную точку.

2). Функция $F(Y) = [\sum_{i=1}^m \mu_i y_i^s]^{\frac{1}{s}}$, при $s > 0$ и $\mu_i > 0$ является возрастающей по каждой переменной на множестве неотрицательных чисел и потому возрастает по \geq на E_{\geq}^m (т.е. в пространстве E^m где все $y_i \geq 0$). Если же $s < 0$ и $\mu_i > 0$, то эта функция возрастает по \geq на $E_{>}^m$ (т.е. в области положительных y_i). Точка максимума такой функции эффективна.

3). Функция $F(Y) = -[\sum_{i=1}^m \mu_i (\hat{y}_i - y_i)^s]^{\frac{1}{s}}$, где $s > 0$, $\mu_i > 0$, а $\hat{y}_i = \sup_{y \in G} y_i$,

$i = \overline{1, m}$, возрастает по \geq на G . Поэтому любая её точка максимума на G эффективна. Отсюда, в частности, следует, что минимизация широко применяемой функции $\sum_{i=1}^m (\hat{y}_i - y_i)^2$ дает эффективную точку.

4). Функция $F(Y) = \prod_{i=1}^m y_i^{\mu_i}$ при $\mu_i > 0$ возрастает по каждой переменной y_i на множестве положительных чисел и поэтому является возрастающей по \geq на E^m . Если же $\mu_i \geq 0$ и есть среди них положительные, то эта функция будет возрастающей по отношению $>$ на E^m .

5). Возьмём функцию $F(Y) = \min_i \mu_i y_i$ при $\mu_i > 0, i = \overline{1, m}$. Если $y_i > y'_i$ для всех i , то и $\mu_i y_i > \mu_i y'_i$ для всех i . Поэтому справедливо неравенство

$$\min_i \mu_i y_i > \min_i \mu_i y'_i$$

и, значит, приведённая функция возрастает по отношению $>$ на E^m . Следовательно, любая её точка максимума на G слабо эффективна.

10.2. Методы многокритериальной оптимизации

Как видно из предыдущих разделов, математические методы оптимизации позволяют находить эффективные решения. Однако из-за несравнимости эффективных решений не удаётся решить задачу со многими критериями до конца без привлечения ЛПР. Исходя из своих представлений об оптимальности решения ЛПР в конечном итоге отдаёт предпочтение одной из множества возможных альтернатив.

В специальной литературе предложены различные способы вовлечения ЛПР в процесс принятия решений. В зависимости от того, на какой стадии процесса выявляются и используются предпочтения ЛПР, можно выделить три группы многокритериальных методов принятия решений:

- 1) основаны на том, что ЛПР может выразить свои предпочтения до начала процесса многокритериальной оптимизации;
- 2) интерактивные (диалоговые) методы;
- 3) методы построения множества эффективных решений с последующим представлением его ЛПР.

В методах первой группы используются различные способы свёртки критериев, лексикографическое упорядочение критериев, установление желаемых уровней критериев и др. Вторая группа методов основана на непосредственном участии ЛПР в процессе оптимизации, когда на каждой итерации компьютер предлагает решения, а ЛПР их оценивает, и с учётом этих оценок компьютер ищет новые решения. Методы третьей группы отличаются друг от друга различными способами построения и представления множества эффективных решений.

Прежде чем перейти к рассмотрению конкретных методов, сделаем одно замечание. На практике далеко не всегда принимаемое решение должно быть эффективным, так как оно соответствует (по определению) предельным возможностям системы. Например, желание иметь некоторые

резервы может привести к выбору решения, которое не будет принадлежать множеству Парето. Однако и такой выбор должен основываться на знании эффективных решений с тем, чтобы правильно оценить границы возможного, а значит и сами резервы.

10.2.1. Методы первой группы

ЛПР может выразить свои предпочтения в различной форме. Это зависит от особенностей самого ЛПР, новизны задачи, типа и числа критериев и других факторов. Поэтому методы данной группы отличаются тем, что используют разные представления предпочтений и способы их формализации. Однако все они в конечном итоге сводят многокритериальную задачу к одной или ряду задач с одним (иногда обобщенным) критерием.

10.2.1.1. Функция полезности

В XIX веке экономисты высказали предположение о существовании у каждого индивидуума определённого количественного измерителя счастья или полезности – “пользомера”. Единицы измерения этого прибора называли “утилиями” (от английского *utility* – полезность). Согласно этой модели потребительского поведения каждый потребитель выбирает так товары и услуги, чтобы максимизировать свою полезность в пределах средств, которыми он располагает. Эта идея перенесена на многокритериальные задачи и интенсивно разрабатывается в теории полезности, ставшей самостоятельным направлением прикладной математики.

Применительно к многокритериальной задаче в качестве товаров и услуг выступают критерии, а в качестве потребителя – ЛПР. При этом предполагается существование на множестве значений критериев y_1, y_2, \dots, y_m скалярной оценки предпочтений ЛПР, называемой полезностью.

Приведём строгое определение этого понятия. Функция U , которая каждой точке Y критериального пространства ставит в соответствие действительное число $U(Y)$, называется функцией полезности (ценности) ЛПР, если

$$\begin{aligned} Y' \sim Y'' &\Leftrightarrow U(Y') = U(Y''), \\ Y' \succ Y'' &\Leftrightarrow U(Y') > U(Y''). \end{aligned}$$

Таким образом, функция полезности представляет собой математическую модель предпочтений ЛПР. Если функция полезности известна, то многокритериальная задача сводится к стандартной задаче оптимизации: найти вектор $X \in D$, максимизирующий $U[Y(X)]$. Множество

точек критериального пространства, одинаковых по предпочтительности (для которых $U(Y)=\text{Const}$), образует гиперповерхность равного уровня функции полезности. Гиперповерхности равного уровня $U(Y)$ называются кривыми безразличия, а семейство всех кривых безразличия – картой безразличия. Такая терминология связана с тем, что для любых двух альтернатив Y' и Y'' , лежащих на одной кривой безразличия, $U(Y')=U(Y'')$, т.е. ЛПР всё равно, достигнет он Y' или Y'' . Пример карты безразличия некоторой функции полезности и нахождение по ней оптимального решения показаны на рис. 10.4.

Очевидно, что наибольшие затруднения при практическом применении рассматриваемого подхода вызывает построение функции полезности, адекватно отражающей предпочтения ЛПР.

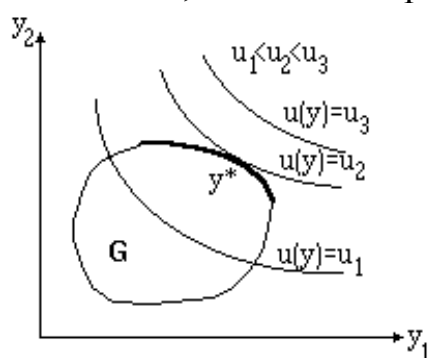


Рис. 10.4

Чтобы построить $U(Y)$, прежде всего необходимо установить её вид, который определяется структурой предпочтения ЛПР. Выявление структуры предпочтений – самый ответственный этап построения функции полезности. Следует однако заметить, что если функция U однозначно определяет всю структуру предпочтений, то обратное неверно. Это значит, что одна и та же структура может быть представлена разными функциями полезности, которые являются

стратегически эквивалентными. Функции полезности $U_1(Y)$ и $U_2(Y)$ стратегически эквивалентны, если они приводят к одинаковому упорядочению по предпочтению. Так любые U_1 и U_2 , связанные какой-либо монотонно возрастающей функцией $T(\bullet)$, являются эквивалентными. Действительно, максимизация $U_1(Y)$ и $U_2(Y)=T[U_1(Y)]$ приведет к одному результату, т.е. обе функции одинаково отражают структуру предпочтений ЛПР.

В благоприятных случаях удастся описывать предпочтения ЛПР функцией полезности, имеющей сравнительно простой вид:

$$U(y_1, y_2, \dots, y_m) = F[U_1(y_1), U_2(y_2), \dots, U_m(y_m)], \quad (10.7)$$

т.е. многомерная функция определяется через одномерные функции полезности значений одного отдельно взятого критерия. Типичные примеры таких функций для $m=2$:

$$U(Y) = C_1 y_1 + C_2 y_2, \quad C_1 > 0, C_2 > 0,$$

$$U(Y) = y_1^\alpha y_2^\beta, \quad \alpha > 0, \beta > 0.$$

Формализация структуры предпочтений основана на исследовании возможности взаимной компенсации значений различных критериев. Это проблема замещения по полезности. Возможные замещения на наборе критериев может дать только ЛПР, а выявить их у ЛПР и формально описать – задача аналитика. Далее будем предполагать,

что критерии независимы по предпочтению (см. раздел 10.1.3), и рассмотрим простейшие случаи построения $U(Y)$ для $m=2$.

Предельный коэффициент замещения критерия y_1 на критерий y_2 в точке (y'_1, y'_2) равен α , если ЛПР согласен уступить $\alpha\Delta$ единиц критерия y_1 за Δ единиц критерия y_2 , где Δ - достаточно малая величина (строго говоря, при $\Delta \rightarrow 0$). В общем случае предельные коэффициенты замещения зависят от значений y_1 и y_2 . Возвращаясь к рис.10.4 нетрудно понять, что предельный коэффициент есть тангенс угла наклона касательной к кривой безразличия в точке (y'_1, y'_2) , взятый с обратным знаком. Отсюда ясно, что определение коэффициентов замещения дает представление о карте безразличия, а следовательно, и о виде функции полезности.

Если коэффициент замещения не зависит от значений y_1 и y_2 , то это означает, что кривые безразличия – прямые, имеющие вид

$$y_1 + \lambda y_2 = \text{Const},$$

а предпочтения ЛПР могут быть представлены функцией

$$U(Y) = y_1 + \lambda y_2. \quad (10.8)$$

Если предельный коэффициент замещения зависит от значения y_2 , но не зависит от y_1 , то подходящей составной функцией полезности может быть

$$U(Y) = y_1 + U_2(y_2). \quad (10.9)$$

Возможный путь получения $U_2(y_2)$ состоит в следующем. Примем произвольно $U_2(y_2^0) = 0$, что дает точку отсчета значений U_2 . Тогда $U_2(y'_2)$ есть сумма в единицах y_1 , которую ЛПР согласен "заплатить" за переход от y_2^0 к y'_2 . Выявив у ЛПР эти суммы для ряда значений y_2 , получим график функции $U_2(y_2)$ как, например, на рис.10.5.

Карта безразличия, соответствующая структуре предпочтений (10.9), имеет вид семейства кривых, которые получаются простым сдвигом по горизонтали (по оси y_1) одной из них. Аналогичный подход применяется и в случае, когда α зависит от y_1 , но не зависит от y_2 .

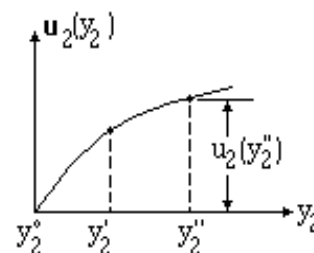


Рис. 10.5

Из ситуаций, когда коэффициент замещения зависит от значений обоих критериев, рассмотрим самую простую: структура предпочтений ЛПР аддитивна. Обратимся к рис 10.6. Если вместо знака вопроса (?) ЛПР поставит d , т.е. он согласен заплатить d единиц y_1 за увеличение y_2 на c в точке D и это остается в силе при любых значениях a , b , c и d и в любых точках A , B , C и D , образующих прямоугольник, то имеет место условие соответственных замещений.

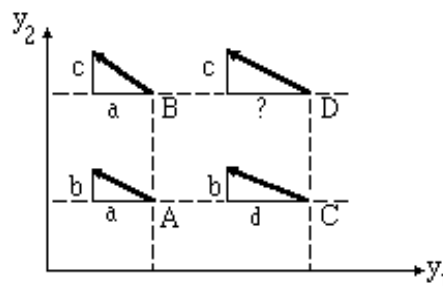


Рис. 10.6

Доказано, что структура предпочтений аддитивна и, следовательно, описывается функцией полезности вида

$$U(Y) = U_1(y_1) + U_2(y_2) \quad (10.10)$$

тогда и только тогда, когда выполняется условие соответственных замещений.

Одна из процедур построения функции (10.10), которую рассмотрим ниже, использует эквивалентность замещений в разных диапазонах одного из критериев. Приведем необходимые определения. Пара (y_1^a, y_1^b) эквивалентна по разности полезности паре (y_1^c, y_1^d) , где $y_1^a < y_1^b$ и $y_1^c < y_1^d$, если для любого исходного значения критерия y_2 ЛПР согласен "заплатить" одно и то же количество единиц y_2 за увеличение y_1 как от y_1^a до y_1^b , так и от y_1^c до y_1^d . Средней по полезности точкой y_1^c интервала $[y_1^a, y_1^b]$ значений критерия y_1 называется точка, которая образует эквивалентные по разности полезности пары $[y_1^a, y_1^c]$ и $[y_1^c, y_1^b]$. Заметим, что из условия соответственных замещений следует независимость значения средней точки для данного интервала y_1 от значений критерия y_2 , хотя "плата" за изменение y_1 (в единицах y_2) будет зависеть от уровня y_2 .

Для построения функции полезности предварительно устанавливаем область возможных значений критериев: $\underline{y}_i \leq y_i \leq \overline{y}_i$. Полагая, что структура предпочтений ЛПР аддитивна (на основе соответствующих предварительных исследований), функцию полезности представим в виде

$$U(y_1, y_2) = \lambda_1 U_1(y_1) + \lambda_2 U_2(y_2), \quad (10.11)$$

где

$$U_i(\underline{y}_i) = 0, U_i(\overline{y}_i) = 1, \quad (10.12)$$

$$\lambda_1 > 0, \lambda_2 > 0 \text{ и } \lambda_1 + \lambda_2 = 1. \quad (10.13)$$

В процедуре отыскания U в виде (10.11), приводимой ниже, одинаковость пар и значения средних точек определяет ЛПР в диалоге с аналитиком.

I. Строим U_1 в следующей последовательности:

- находим среднюю по полезности точку $y_1^{0.5}$ интервала $[\underline{y}_1, \overline{y}_1]$ и полагаем $U_1(y_1^{0.5}) = 0.5$;
- находим среднюю по полезности точку $y_1^{0.75}$ интервала $[y_1^{0.5}, \overline{y}_1]$ и полагаем $U_1(y_1^{0.75}) = 0.75$;
- находим среднюю по полезности точку $y_1^{0.25}$ интервала $[\underline{y}_1, y_1^{0.5}]$ и принимаем $U_1(y_1^{0.25}) = 0.25$;

- проверяем согласованность результатов: является ли $y_1^{0.5}$ средней по полезности точкой интервала $[y_1^{0.25}, y_1^{0.75}]$? Если нет, то придется корректировать эти точки до достижения согласованности.
 - по пяти определенным точкам (или большему числу, если продолжить дробление интервалов) строится график функции $U_1(y_1)$.
2. Таким же образом находим $U_2(y_2)$.
3. Определяем коэффициенты шкалирования λ_1 и λ_2 . Для этого выбираем любые две одинаковые по предпочтительности пары (y_1, y_2) . Пусть, например, это пары (y'_1, y'_2) и (y''_1, y''_2) . Тогда

$$U(y'_1, y'_2) = U(y''_1, y''_2)$$

или

$$\lambda_1 U_1(y'_1) + \lambda_2 U_2(y'_2) = \lambda_1 U_1(y''_1) + \lambda_2 U_2(y''_2). \quad (10.14)$$

Значения U_1 и U_2 в точках y'_i и y''_i определяются по построенным графикам. Добавив к (10.14) равенство (10.13), находим значения λ_1 и λ_2 .

Соответствие полученной функции полезности структуре предпочтений ЛПР можно дополнительно проверить, предложив ему несколько пар значений критериев (отличных от y'_i и y''_i) с одним значением U . Если ЛПР сочтет их примерно одинаковыми по предпочтительности, то построенная функция приемлема. В противном случае следует повторить 3-й этап процедуры с тем, чтобы уточнить значения λ_1 и λ_2 (их можно получить более надежно усреднением по нескольким парам с равной предпочтительностью).

С увеличением размерности критериального пространства трудоемкость построения функции полезности даже в аддитивном виде резко возрастает. А при более сложной структуре предпочтений ЛПР отыскание адекватной $U(Y)$ становится весьма проблематичным.

Несмотря на то что имеется целый ряд хорошо разработанных процедур построения функции полезности (например, Р.Кини и Х.Райфа) рассмотренный подход к решению многокритериальных задач находит ограниченное применение. И прежде всего это связано с необходимостью длительной и напряженной работы с ЛПР. А как известно, руководители - народ занятой, да и далеко не все из них могут высказывать непротиворечивые суждения. Проблема многократно усложняется в ситуациях группового принятия решений. В то же время следует отметить главное достоинство метода: функция полезности наиболее полно и адекватно отражает систему ценностей ЛПР и позволяет относительно просто находить решение, наиболее предпочтительное (и в этом смысле оптимальное) для ЛПР с помощью одной стандартной задачи оптимизации.

10.2.1.2. Решение на основе лексикографического упорядочения критериев

Как и в предыдущем подходе, предпочтения ЛПР выявляются до поиска наилучшего решения. Метод применим, если для ЛПР приемлемо ранжирование критериев по важности и при этом предпочтительным является то решение, в котором лучше значение более важного критерия независимо от значения всех менее важных критериев.

Лексикографическое отношение \succ^{lex} определяется следующим образом. Для двух векторов $\mathbf{Y} = (y_1, y_2, \dots, y_m)$ и $\mathbf{Y}' = (y'_1, y'_2, \dots, y'_m)$ имеет место отношение $\mathbf{Y} \succ^{lex} \mathbf{Y}'$ тогда и только тогда, когда выполняется одно из условий:

$$\begin{aligned} 1) & y_1 > y'_1 \\ 2) & y_1 = y'_1, y_2 > y'_2 \\ & \dots\dots\dots \\ m) & y_i = y'_i, i = \overline{1, m-1}, y_m > y'_m. \end{aligned} \quad (10.15)$$

В этом случае говорят, что вектор \mathbf{Y} лексикографически больше вектора \mathbf{Y}' .

Множество векторов (решений), оптимальных по отношению \succ^{lex} на G (соответственно на D), называют множеством лексикографически оптимальных точек ($Opt_{lex}Y$). Так как для любых двух векторов \mathbf{Y}, \mathbf{Y}' либо один лексикографически больше другого, либо они равны, то множество $Opt_{lex}Y$, если оно не пустое, содержит только один элемент. Так на множестве G , изображенном на рис.10.3, лексикографически оптимальной является только точка h .

Лексикографически-оптимальное решение достигается в процессе решения следующей последовательности задач:

- 1) находим $y_1^* = \max f_1(\mathbf{X})$ при условии $\mathbf{X} \in D$;
- 2) находим $y_2^* = \max f_2(\mathbf{X})$ при условии $\mathbf{X} \in D, f_1(\mathbf{X}) = y_1^*$;
-
- m) находим $y_m^* = \max f_m(\mathbf{X})$ при условии $\mathbf{X} \in D, f_i(\mathbf{X}) = y_i^*, i = \overline{1, m-1}$.

Процесс решения прекращается, как только очередная задача из этой последовательности дает единственное решение. Нетрудно показать, что такая процедура приводит к решению многокритериальной задачи, которое принадлежит парето-оптимальному множеству. В то же время, если остановиться на задаче, имеющей не единственное решение, то нельзя гарантировать, что полученное решение является эффективным (оно может быть слабо эффективным).

В случае линейной модели решение последовательности отдельных задач можно объединить в один симплекс-процесс, что значительно

снижает трудоемкость решения. Для этого применяют лексикографический вариант симплекс-метода.

В этом методе каждому критерию соответствует своя строка относительных оценок Δ_j^i (i -индекс критерия, j -индекс переменной). Строки располагаются в порядке убывания приоритетов критериев. Сначала симплекс-преобразования выполняются по Δ^1 . При достижении оптимального решения по 1-му критерию ($\forall \Delta_j^1 \geq 0$) выявляют нулевые оценки небазисных переменных. Если таких нет, то решение единственное и лексикографическая оптимизация завершается. Если они есть, то в строке Δ^2 в столбцах с выделенными нулевыми Δ^1 ищут отрицательные оценки. Небазисная переменная x_s , для которой $\Delta_s^1 = 0$, а $\Delta_s^2 < 0$, вводится в базисное решение, улучшая значение 2-го критерия без ухудшения значения 1-го критерия. Этот процесс продолжается, пока будут выявляться такие перспективные переменные. Если они исчерпались или их не было, то переходят к решению по 3-му критерию, т.е. к выбору перспективных переменных по строке Δ^3 . Введение в решение небазисной переменной x_p улучшит значение 3-го критерия без изменения первых двух, если $\Delta_p^1 = \Delta_p^2 = 0$, а $\Delta_p^3 < 0$. Процесс решения многокритериальной задачи завершается, когда на последующих этапах не находятся перспективные переменные.

10.2.1.3. Метод главного критерия

Суть метода состоит в том, что ЛПР выделяет главный критерий (далее $f_1(X)$), а на остальные критерии накладывает требования, что они были не меньше задаваемых им минимальных (пороговых) значений t_i . Тогда многокритериальная задача сводится к однокритериальной задаче

$$\begin{aligned} f_1(X) &\Rightarrow \max, \\ f_i(X) &\geq t_i, \quad i = \overline{2, m}, \\ X &\in D. \end{aligned} \quad (10.16)$$

Если эта задача разрешима, то ее решение всегда является слабо эффективным, а если оно единственно, то и эффективным. Заметим, что этот вывод не зависит от выбора главного критерия. Рис.10.7 иллюстрирует случай единственного решения задачи (10.16), а рис.10.8 – множества оптимальных решений, лежащего на границе ab , из которого только точка a является эффективной.

Практически задачу (10.16) решают для нескольких наборов значений $\{t_i\}$, и затем на основании анализа полученных (слабо) эффективных решений ЛПР определяет наиболее предпочтительное из них.

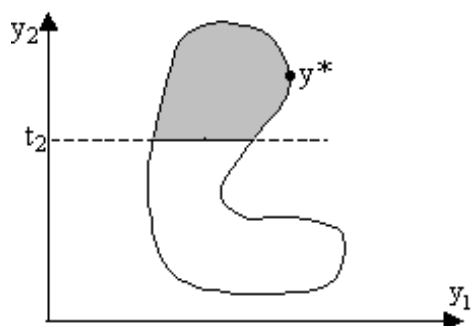


Рис. 10.7

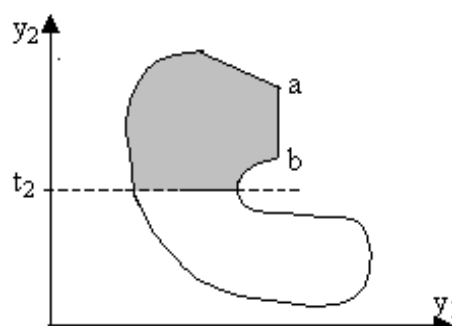


Рис. 10.8

Рассмотренный метод целесообразно применять, когда ЛПР может обоснованно назначить значения t_i или указать узкие пределы для них.

10.2.1.4. Линейная свертка

Если ЛПР может не только ранжировать критерии, но и дать сравнительную количественную оценку значимости (важности) критериев, решение многокритериальной задачи сводится к обычной задаче с одним критерием, в качестве которого берется обобщенный показатель вида

$$F(\mathbf{X}) = \sum_{i=1}^m C_i f_i(\mathbf{X}), \quad (10.17)$$

где C_i – положительные числа, отражающие веса критериев в структуре предпочтений ЛПР. При групповом ЛПР C_i находятся по индивидуальным весам одним из методов обработки экспертных оценок. Обычно значения C_i нормируются так, чтобы $\sum_{i=1}^m C_i = 1$. Как следует из теоремы 5, точка максимума функции (10.17) при положительных C_i является эффективной.

Данный способ решения многокритериальной задачи имеет существенные недостатки. Во-первых, большие затруднения возникают при определении весов. Одно дело – расположить критерии по важности, и совсем другое – оценить на сколько или во сколько один критерий важнее другого. Во-вторых, неизвестна связь между значениями весов и значениями критериев в точке максимума $F(\mathbf{X})$. Очень часто эта зависимость оказывается существенно нелинейной (даже в линейных задачах), включая зоны нечувствительности значений f_i к изменению C_i . Поэтому для получения решения, удовлетворяющего ЛПР, приходится максимизировать $F(\mathbf{X})$ для нескольких наборов C_i . Наконец, заметим, что в свертке (10.17) целесообразно все критерии приводить к одним единицам измерения. С этой целью лучше представлять критерии в относительных единицах, беря за базовое максимальное или желаемое значение. Достоинство метода – в стандартности задачи, к которой сводится исходная многокритериальная проблема.

Пример 10.1. Рассмотрим задачу линейного программирования с

тремя критериями: максимизировать

$$f_1(X) = -3x_1 + 2x_2,$$

$$f_2(X) = 4x_1 + 3x_2,$$

$$f_3(X) = 2x_1 - 5x_2$$

при условиях

$$2x_1 + 3x_2 \leq 18,$$

$$2x_1 + x_2 \leq 10,$$

$$x_1, x_2 \geq 0.$$

Допустимая область и линии равного уровня критериев показаны на рис.10.9. Максимальное значение функции $f_1(X)$ равно 12 и достигается в точке $A(0,6)$, при этом $f_2^A=18$, $f_3^A=-30$; $\max f_2(X)=24$ в точке $B(3,4)$, где $f_1^B=-1$ и $f_3^B=-14$; $\max f_3(X)=10$ в точке $C(5,0)$, в которой $f_1^C=-15$ и $f_2^C=20$. Если взять свертку с равными весами, то есть

$$F(X) = \frac{1}{3}f_1(X) + \frac{1}{3}f_2(X) + \frac{1}{3}f_3(X) = x_1,$$

то результат максимизации $F(X)$, как легко убедиться, совпадает с максимизацией одной функции $f_3(X)$. Таким образом, при равных весах решение по линейной свертке дает наилучшее значение f_3 и наихудшее для f_1 . Используя параметрическое программирование, можно определить

диапазон значений C_i (зону нечувствительности), в котором оптимальное решение по $F(X)$ будет оставаться в точке C .

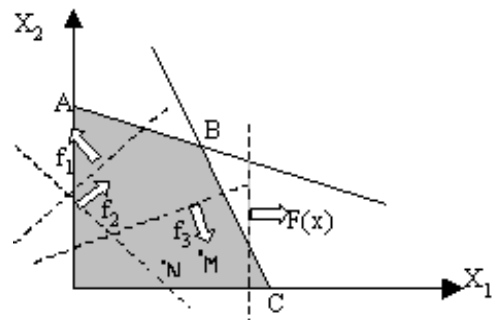


Рис. 10.9

10.2.1.5. Максиминная свертка

Как и в предыдущем подходе, ЛПР должен задать веса C_i всем критериям, но обобщенный критерий записывается в виде

$$F(X) = \min_i C_i f_i(X). \quad (10.18)$$

Тогда многокритериальная задача сводится к максимизации $F(X)$ на $X \in D$. Если ввести новую переменную x_0 , то эта задача преобразуется к виду

$$\begin{aligned} F(X) = x_0 &\Rightarrow \max; \\ C_i f_i(X) &\geq x_0, \quad i = \overline{1, m}; \\ X &\in D, \end{aligned} \quad (10.19)$$

который более удобен для решения. В частности, если в исходной задаче все функции линейны, то и задача в виде (10.19) будет обычной задачей линейного программирования.

Функция (10.18) подпадает под действие теоремы 5, что гарантирует получение, по крайней мере, слабо эффективного решения

многокритериальной задачи.

Максиминная свертка имеет те же недостатки, что и предыдущая, но отличается тем, что максимально увеличивает минимальное слагаемое в (10.17), способствуя относительному сближению значений критериев.

Пример 10.2. Применим рассмотренный подход к задаче из примера 10.1. Снова возьмем равные веса. В соответствии с (10.19) к исходным условиям задачи добавятся ограничения

$$-3x_1 + 2x_2 \geq 3x_0$$

$$4x_1 + 3x_2 \geq 3x_0$$

$$2x_1 - 5x_2 \geq 3x_0$$

и новый критерий $x_0 \Rightarrow \max$. Оптимальное решение этой задачи достигается в точке $x_0^* = x_1^* = x_2^* = 0$, в которой f_1, f_2 и f_3 тоже равны нулю.

Сравним с результатами примера 10.1. При решении по критерию (10.17) получили $\min_i f_i = -15$, а по критерию (10.18) $\min_i f_i = 0$. Однако это решение доставляет наихудшее значение критерию f_2 и ни одному из критериев не обеспечивает максимального значения.

10.2.1.6. Метод идеальной точки

Идеальной или точкой абсолютного максимума называют точку в критериальном пространстве, в которой все критерии достигают своих максимальных значений: $\hat{f}_i = \max_x f_i(\mathbf{X})$.

Если эта точка принадлежит достижимому множеству G , то все эффективное (паретовское) множество состоит из этой единственной точки (рис. 10.10) и проблемы как таковой в этом случае нет. Однако идеальная точка обычно лежит вне множества G и поэтому нереализуема. В связи с

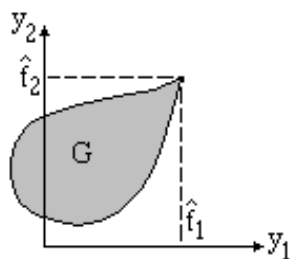


Рис. 10.10

этим ее иногда называют также утопической. Идея метода состоит в том, чтобы на множестве G найти точку, наиболее близкую к идеальной. Мерой близости выступает некоторая функция расстояния $\rho(f(\mathbf{X}), \hat{f})$, в качестве которой используют в общем случае взвешенные L_p -метрики

$$\rho_p = \left[\sum_{i=1}^m (\lambda_i |f_i(\mathbf{X}) - \hat{f}_i|)^p \right]^{\frac{1}{p}},$$

где p может быть любым целым положительным числом и ∞ . Так как возведение в степень $\frac{1}{p}$ является монотонным преобразованием, то на положение экстремума оно не влияет. Таким образом, многокритериальная задача сводится к минимизации функции

$$\sum_{i=1}^m \lambda_i [\hat{f}_i - f_i(\mathbf{X})]^p, \quad (10.20)$$

где λ_i - веса отклонений, задаваемые ЛПР ($\sum \lambda_i = 1, \lambda_i > 0$). На практике чаще используют значение $p=2$. В соответствии с теоремой 5 минимизация такой функции приводит к эффективному решению.

Как и ранее, целесообразно использовать отклонения в относительных единицах, для чего выражение в квадратных скобках в (10.20) можно разделить на \hat{f}_i .

Пример 10.3. Найдем решение для модели из примера 10.1 при одинаковых λ_i и $p=2$. Так как $\hat{f}_1=12, \hat{f}_2=24$ и $\hat{f}_3=10$, обобщенный критерий запишется в соответствии с (10.20) в виде

$$\rho_2 = \frac{1}{3}[12 - (-3x_1 + 2x_2)]^2 + \frac{1}{3}[24 - (4x_1 + 3x_2)]^2 + \frac{1}{3}[10 - (2x_1 - 5x_2)]^2.$$

После отбрасывания общего множителя (1/3), свободного члена (820) и простых преобразований приходим к выражению

$$\rho_2 = 29x_1^2 + 38x_2^2 - 8x_1x_2 - 160x_1 - 92x_2.$$

Используя метод квадратичного программирования, получаем: $x_1^*=2,97, x_2^*=1,52$ (точка М на рис.10.9). В этой точке $f_1=-5.87, f_2=16.44, f_3=-1.66$.

Метрика с $p=\infty$ дает уже рассмотренный ранее подход: критериальная функция определяется как максимальное отклонение

$$\max_i \lambda_i [\hat{f}_i - f_i(\mathbf{X})], \quad (10.21)$$

которое следует минимизировать по $\mathbf{X} \in D$

Пример 10.4. Если воспользоваться сверткой (10.21) для модели из примера 10.1, то получим решение: $x_1^*=1,52, x_2^*=1,37$ (точка N на рис. 10.9), в котором $f_1=-1,82, f_2=10,19, f_3=-3,81$.

Пример 10.5. Пусть требуется максимизировать два критерия

$$\begin{aligned} f_1(\mathbf{X}) &= -2x_1 + x_2, \\ f_2(\mathbf{X}) &= 4x_1 - x_2 \end{aligned}$$

при условиях

$$\begin{aligned} x_1 + x_2 &\leq 8, \\ -x_1 + x_2 &\leq 2, \\ 0 \leq x_1 &\leq 6, \quad 0 \leq x_2 \leq 4. \end{aligned}$$

Так как задача содержит две переменные и два критерия, множества D и G представимы на плоскости, что позволяет наглядно сравнить результаты рассмотренных подходов (рис.10.11).

Решения этой задачи, полученные при различных способах свертки, сведены в

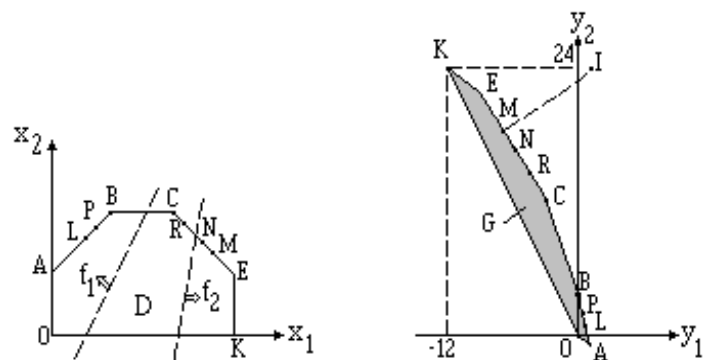


Рис. 10.11

следующую таблицу.

№	Обобщенный критерий	Решение				
		Рис. 10.11	X ₁	X ₂	Y ₁	Y ₂
1	$f_1(\mathbf{X}) \Rightarrow \max$	A	0	2	2	-2
2	$f_2(\mathbf{X}) \Rightarrow \max$	K	6	0	-12	24
3	$\sum C_i f_i(\mathbf{X}) \Rightarrow \max,$ $C_1 = C_2 = \frac{1}{2}$	[K,E]	6	[0,2]	[-12,-10]	[24,22]
4	$\min C_i f_i \Rightarrow \max,$ $C_1 = C_2 = \frac{1}{2}$	L	1	3	1	1
5	$\min \frac{C_i f_i}{\hat{f}_i - f_i} \Rightarrow \max,$ $C_1 = C_2 = \frac{1}{2}$	P	1,176	3,176	0,824	1,53
6	$\sum \lambda_i (\hat{f}_i - f_i(\mathbf{X}))^2 \Rightarrow \min,$ $\forall \lambda_i = \frac{1}{2}$	M	$5\frac{4}{17}$	$2\frac{13}{17}$	-7,7	18,2
7	$\max \lambda_i (\hat{f}_i - f_i(\mathbf{X})) \Rightarrow \min,$ $\forall \lambda_i = \frac{1}{2}$	N	4,75	3,25	-6,25	15,75
8	$\max \lambda_i \frac{\hat{f}_i - f_i(\mathbf{X})}{\hat{f}_i - \bar{f}_i} \Rightarrow \min,$ $\forall \lambda_i = \frac{1}{2}$	R	4,08	3,92	-4,24	12,4

Здесь квадратными скобками обозначены интервалы, соответствующие множеству решений, оптимальных по данному обобщенному критерию. Как видно из таблицы, линейная свертка с равными весовыми коэффициентами (строка 3) дает весьма однобокий результат: значения второго критерия лежат в области максимума, а первого – в области минимума. Максиминная свертка дала равные абсолютные значения критериев, но второй критерий сильнее, чем первый, отличается от своего максимально возможного значения (строка 4). Очевидно, если использовать в этой свертке относительные величины критериев, взяв за базу, например, разность ($\max f_i - \min f_i$), то можно ожидать более "справедливого" соотношения значений критериев в оптимальном решении. Действительно, максимизация минимальной относительной величины критерия с весовым коэффициентом приводит к увеличению f_2 и уменьшению f_1 (строка 5). Следующие два решения, представленные в 6 и

7 строках таблицы, минимизируют отклонения от идеальной точки I. Результат, соответствующий минимуму суммы квадратов отклонений, можно получить геометрически. Так при одинаковых значениях λ_i , как в нашем случае, линии равного уровня обобщенного критерия представляют собой окружности с центром в идеальной точке. Точка минимума есть точка касания линии равного уровня и границы области достижимости G , а так как у нас линии - окружности, то это будет основание перпендикуляра, опущенного из идеальной точки на ближайшую границу G (точка M). Использование минимаксного отклонения приводит к выравниванию отклонений критериев: если в точке M отклонения составляют 9,7 и 5,8, то в точке N - 8,25 для обоих критериев. Решение по максимальному относительному отклонению представлено в строке 8 таблицы и точкой R на рис 10.11.

Таким образом, все способы свертки дают решения, принадлежащие паретовскому множеству, которое лежит на ломаной КЕСВА (рис.10.11).

10.2.1.7. Целевое программирование (ЦП)

Целевое программирование применяется в основном для решения линейных многокритериальных задач, но может быть использовано и в нелинейных задачах.

Принципиальное отличие ЦП от вышерассмотренных подходов – в изменении концепции цели. Вместо максимизации (минимизации) критериев ставится задача оптимального приближения к желаемым значениям критериев, которые называют также уровнями притязаний ЛПР. Таким образом, эти значения, обозначаемые далее как \bar{y}_i , и представляют собой цель, к которой следует стремиться. Если в методе главного критерия ограничения на критерии (10.16) могут приводить к неразрешимости задачи, то в ЦП, как будет показано дальше, желаемые значения \bar{y}_i , какими бы они ни были, не могут явиться причиной неразрешимости.

Притязания ЛПР могут быть выражены по-разному в зависимости от смысла критерия:

- 1) не меньше \bar{y}_i ;
- 2) не больше \bar{y}_i ;
- 3) равно \bar{y}_i ;
- 4) принадлежать диапазону $[\bar{y}_i', \bar{y}_i'']$.

Соответственно по-разному эти требования учитываются в математической модели задачи.

Как правило, множество решений, на котором достигаются одновременно все уровни притязаний, не пересекается с допустимым множеством. В таких случаях оно называется **утопическим**. Заметим, что

утопическое множество решений не обязательно должно быть непустым. В то же время утопическое множество в критериальном пространстве пустым быть не может. Приведем поясняющие примеры. На рис.10.12 показаны утопические множества для случая, когда притязания ЛПР заданы в виде

$$y_1 = f_1(\mathbf{X}) \geq \bar{y}_1, \quad y_2 = f_2(\mathbf{X}) \in [\bar{y}_2', \bar{y}_2''].$$

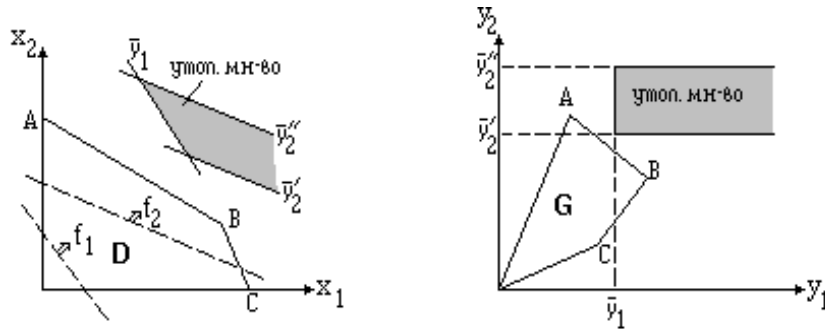


Рис. 10.12

На рис.10.13 представлен случай, когда ЛПР ставит цель в виде: $y_i = f_i(\mathbf{X}) = \bar{y}_i$, $i=1,2,3$. При этом утопическое множество решений оказывается пустым, так как нет точек, в которых одновременно все критерии достигают уровней притязаний. Однако совершенно очевидно, что в пространстве критериев утопическое множество не пусто: оно состоит из одной точки с координатами $(\bar{y}_1, \bar{y}_2, \bar{y}_3)$.

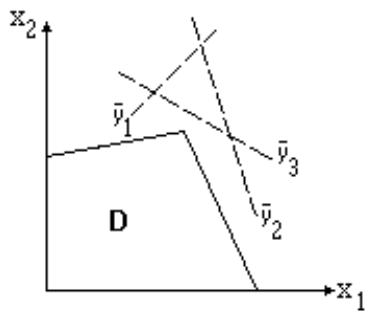


Рис. 10.13

При целевом программировании изменяется модель задачи:

- к исходным условиям задачи добавляются так называемые целевые ограничения, отражающие уровни притязаний;
- с целевыми ограничениями в модель вводятся новые переменные, имеющие смысл отклонений от желаемых значений исходных критериев;
- критерий в модели ЦП строится как функция новых переменных.

Пусть, например, исходная задача содержит 4 критерия и ЛПР выдвигает по ним разные варианты притязаний:

$$\begin{aligned} f_1(\mathbf{X}) &\geq \bar{y}_1, \\ f_2(\mathbf{X}) &\leq \bar{y}_2, \\ f_3(\mathbf{X}) &= \bar{y}_3, \\ f_4(\mathbf{X}) &\in [\bar{y}_4', \bar{y}_4'']. \end{aligned}$$

Тогда целевые ограничения будут иметь вид:

$$\begin{aligned} f_1(\mathbf{X}) + d_1^- &\geq \bar{y}_1, \\ f_2(\mathbf{X}) - d_2^+ &\leq \bar{y}_2, \\ f_3(\mathbf{X}) - d_3^+ + d_3^- &= \bar{y}_3, \\ f_4(\mathbf{X}) + d_4^- &\geq \bar{y}'_4, \\ f_4(\mathbf{X}) - d_4^+ &\geq \bar{y}''_4, \\ \forall d_i &\geq 0. \end{aligned}$$

де d_i^- – переменные-отклонения, характеризующие недостижение \bar{y}_i , d_i^+ – переменные-отклонения, означающие превышение \bar{y}_i . Все эти отклонения нежелательны. Поэтому в модели ЦП цель выражается минимизацией переменных-отклонений. Так как число этих переменных больше единицы, мы снова имеем многокритериальную задачу, в которой роль критериев играют переменные d_i . Очевидно, что для ее решения могут быть применены способы, описанные выше:

- лексикографическое упорядочение d_i ;
- линейная свертка

$$\sum_{i=1}^4 (c_i^- d_i^- + c_i^+ d_i^+) \Rightarrow \min;$$

- минимаксная свертка

$$\max_i (c_i^- d_i^-, c_i^+ d_i^+) \Rightarrow \min;$$

- квадратичная свертка (аналог (10.20))

$$\sum_{i=1}^4 (c_i^- d_i^{-2} + c_i^+ d_i^{+2}) \Rightarrow \min.$$

Если исходная модель задачи линейная, то и модели ЦП во всех случаях, кроме последнего, также линейны.

Принципиальной особенностью целевых Ограничений является то, что они не сужают исходную область, а наоборот, расширяют, переводя ее в пространство решений большей размерности (за счет переменных d_i). Поэтому они не могут быть причиной неразрешимости задачи. Последнее свойство следует также из того, что на переменные-отклонения не накладывается требование равенства нулю, а значит, всегда найдутся такие неотрицательные d_i , которые обеспечат выполнение целевых ограничений.

10.2.2. Интерактивные методы

Главный недостаток рассмотренных выше методов состоит в том, что в большинстве реальных ситуаций ЛПР не располагает до решения задачи информацией о ее свойствах, достаточной для надежного назначения

требуемых от него величин. Поэтому, как правило, этими методами не удастся за один раз получить приемлемое решение. А многократное применение с корректировкой назначаемых величин фактически переводит их в разряд интерактивных, но не приспособленных для дружественного общения с ЛПР, методов.

Интерактивный процесс решения многокритериальной задачи реализуется путем диалога ЛПР с компьютером. При этом происходит чередование этапов вычислений, выполняемых компьютером, и корректировки и принятия решений ЛПР. Такая процедура позволяет ЛПР более полно и глубоко оценить взаимосвязи критериев и возможности оптимизируемой системы. Более того, в интерактивном процессе может развиваться формирование предпочтений, компромиссов и даже системы ценностей. Все это облегчает ЛПР нахождение решения, наилучшего с его точки зрения, и повышает уверенность в правильности выбора. Поэтому такая технология оказывается более реалистичной, более гибкой и более приемлемой для руководителей.

Многочисленные интерактивные процедуры предлагают разные схемы решения многокритериальной задачи и предъявляют неодинаковые требования к ЛПР. При прочих равных условиях (сходимость, затраты ресурсов и др.) предпочтительнее те процедуры, которые генерируют только эффективные решения и вызывают у ЛПР меньше затруднений в выражении своих предпочтений.

10.2.2.1. Метод уступок

Предварительно ЛПР ранжирует критерии по важности. В результате критериям присваиваются номера в порядке убывания важности. После этого начинается основная часть диалога. Решается задача максимизации первого критерия при $X \in D$. Если задача имеет множество оптимальных решений, то на нем ищется решение, наилучшее по второму критерию. Если и оно не единственно, то включается третий критерий, и так до достижения единственного решения. Иначе говоря, ищется лексикографически-оптимальное решение. ЛПР предъявляется полученное решение X^1 со значениями всех критериев. ЛПР анализирует это решение и если оно его не устраивает, диалог продолжается. ЛПР просят указать, на какую величину он согласен снизить значение первого критерия с тем, чтобы улучшить значение второго. В результате формируется новая задача:

$$\begin{aligned} f_2(X) &\Rightarrow \max, \\ f_1(X) &\geq \hat{f}_1 - \Delta_1, \\ X &\in D, \end{aligned} \tag{10.22}$$

где Δ_1 - уступка по первому критерию. Снова ищется лексикографическое решение, начиная с задачи (10.22.).

ЛПР оценивает предъявленное ему новое решение X^2 и прежде всего улучшение второго критерия, которое определяется как разность в двух решениях: $f_2(X^2) - f_2(X^1)$. За такое увеличение f_2 он платит цену, равную Δ_1 . Если значение $f_2(X^2)$ не удовлетворяет ЛПР, он может увеличить уступку Δ_1 и снова решить задачу (10.22.). Возможность улучшения значения одного критерия за счет другого показана на рис 10.14. Решение по первому критерию соответствует точке В. Введение уступки позволяет получить решение с лучшим значением f_2 (точка А). Если решение X^2 не обеспечивает приемлемого значения f_3 , ЛПР должен назначить уступку по второму критерию - Δ_2 . Тогда решается задача

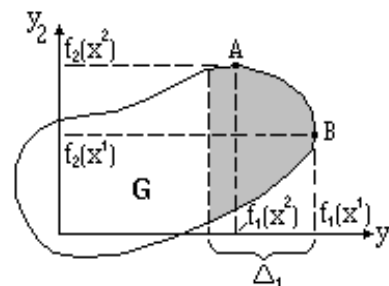


Рис.10.14

$$\begin{aligned} f_3(X) &\Rightarrow \max, \\ f_1(X) &\geq \hat{f}_1 - \Delta_1, \\ f_2(X) &\geq \hat{f}_2(X^2) - \Delta_2, \\ X &\in D. \end{aligned} \quad (10.23)$$

Аналогично формируются задачи по остальным критериям, если их значения не устраивают ЛПР. Очевидно, что в процессе поиска наилучшего решения ЛПР может возвращаться на любое число шагов назад, изменять свои уступки и получать новые решения. Тем самым он выявляет количественные взаимосвязи (замещения) критериев, что облегчает выбор окончательного решения.

Пример 10.6. Решим задачу из примера 10.1. Пусть ЛПР представил ранжирование критериев в виде: f_1, f_3, f_2 . Максимум f_1 достигается в точке А (рис.10.9), где $f_1 = \hat{f}_1 = 12, f_3 = -30, f_2 = 18$. ЛПР не удовлетворен значением критерия f_3 и готов пойти на снижение критерия f_1 на величину $\Delta_1 = 7$. В соответствии с рассмотренной процедурой в условия задачи вводится новое ограничение

$$f_1(X) \geq \hat{f}_1 - \Delta_1$$

или в явном виде

$$-3x_1 + 2x_2 \leq 5.$$

В результате допустимое множество сузится до треугольника AMN (рис.10.15). Найдем решение, максимизирующее f_3 на этом множестве. Оно лежит в вершине N, где $f_1 = 5, f_3 = -12,5$ и $f_2 = 7,5$. Таким образом, за счет снижения первого критерия на 7 единиц увеличилось значение третьего критерия (второго по важности) на 17,5. Однако ЛПР не устраивает значение критерия f_2 .

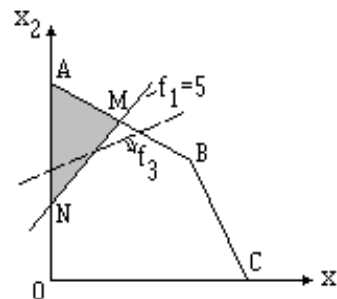


Рис. 10.15

Чтобы повысить его, ЛПР согласен уменьшить f_3 до -18, то есть уступает $\Delta_3=5,5$. Тогда условия задачи дополняются еще одним ограничением

$$f_3(X) \geq -18 \text{ или } -2x_1 + 5x_2 \leq 18,$$

и допустимое множество уменьшается до треугольника NPQ (рис.10.16).

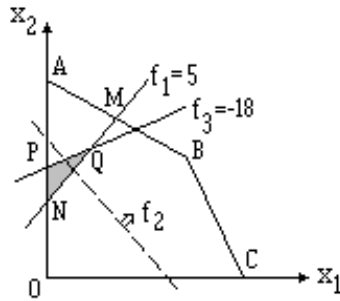


Рис.10.16

Максимизируя f_2 , получим решение в точке Q со значениями критериев: $f_1=5$, $f_3=-18$, $f_2=16$. Как видно, второй критерий увеличился на 8,5 за счет снижения третьего на 5,5. Анализируя полученное решение, ЛПР либо принимает его за окончательное, либо, изменив уступки, продолжает поиск.

Нетрудно убедиться в том, что решения формируемых задач, если они единственны, принадлежат паретовскому множеству исходной многокритериальной задачи.

10.2.2.2. Интерактивное компромиссное программирование

Так называется метод, использующий для поиска удовлетворяющего ЛПР компромиссного решения линейную свертку. При этом от ЛПР не требуется назначать веса, ему лишь нужно на каждой итерации выбрать одно решение из $(m+1)$ -го предлагаемых, после чего веса вычисляются программно.

Сравнивать альтернативные решения легче, если критерии измеряются в одной шкале. С этой целью в данном методе критерии заменяются функциями степени близости, которые определяются по формуле

$$d_i(\mathbf{X}) = \frac{f_i(\mathbf{X}) - \tilde{f}_i}{\hat{f}_i - \tilde{f}_i}, \quad (10.24)$$

где \hat{f}_i, \tilde{f}_i — максимальное и минимальное значения i -го критерия на допустимом множестве D . Как следует из (10.24), $d_i(\mathbf{X})$ может изменяться от 0 до 1. Очевидно, что если $f_i(\mathbf{X})$ — линейная функция, то и $d_i(\mathbf{X})$ тоже линейная.

Теперь паретовские (эффективные) решения можно находить, максимизируя свертку

$$d^{m+1}(\mathbf{X}) = \sum_{i=1}^m \lambda_i d_i(\mathbf{X}) \quad (10.25)$$

при условии $\mathbf{X} \in D$.

Одним из существенных недостатков ряда интерактивных методов является использование параметров, численные значения которых должен указывать ЛПР. В рассматриваемом методе эта проблема исключается, так

как веса в (10.25) определяются не ЛПР, а специальной процедурой, обеспечивающей вычисление на каждой итерации значений весов, оптимальных в максиминном смысле. Процедура основана на игровом подходе, а именно, на формализации и решении игры двух лиц с нулевой суммой. В качестве стратегий первого игрока рассматриваются целевые функции, второго - решения многокритериальной задачи, полученные к данному шагу и не забракованные ЛПР. Платежом на каждой паре стратегий i, j является степень близости i -й целевой функции на j -м решении к своему максимальному значению \hat{f}_i . Тогда вероятности применения стратегий первым игроком и будут иметь смысл весов целевых функций, входящих в свертку (10.25).

В целом процесс поиска компромиссного решения рассматриваемым методом можно представить следующим образом. Процедура начинается с решения $2m$ обычных задач математического программирования для нахождения максимальных и минимальных значений m целевых функций. По ним вычисляют степень близости каждого решения к максимально возможному значению каждой целевой функции. Приняв их за платежи игры 2-х лиц размерности $m \times m$ и решив игровую задачу, получают текущие веса целевых функций. Последние используют для нахождения $(m+1)$ -го решения. Для полученного решения также вычисляются степени близости по всем критериям. Теперь вместе с предыдущими m решениями новое решение предъявляют ЛПР. Его спрашивают, предпочитает ли он одно из этих решений всем другим. Если да, то это решение принимается за окончательное и процесс заканчивается. В противном случае ЛПР просят выделить из предъявленных ему решений наименее предпочитаемое. Это решение исключается, а из оставшихся m решений формируется и решается очередная игровая задача. Далее таким же способом, как описано выше, находят новое альтернативное компромиссное решение и предъявляют его вместе с m уже имеющимися. Итеративный процесс продолжается до тех пор, пока ЛПР не выявит предпочитаемое им решение.

Рассмотренная процедура реализуется следующим алгоритмом.

Шаг 0. Определить \hat{f}_i и \check{f}_i $i = \overline{1, m}$, для чего решить задачи

$$a) \quad \hat{f}_i = \max f_i(\mathbf{X})$$

при условии $\mathbf{X} \in D$. Очевидно, что получаемые решения $(\hat{\mathbf{X}}_i, \hat{f}_i)$ не что иное, как идеальные решения.

$$б) \quad \check{f}_i = \min f_i(\mathbf{X})$$

при условии $\mathbf{X} \in D$. В противоположность предыдущим решения $(\check{\mathbf{X}}_i, \check{f}_i)$ называют антиидеальными.

Шаг 1. Взять решения $\check{\mathbf{X}}_{ij}, i = \overline{1, m}$ в качестве первоначальных решений

$\mathbf{X}^j, j = \overline{1, m}$ и вычислить степени близости по формуле (10.24).

Построить таблицу

	X^1	X^2	...	X^m
f_1	d_1^1	d_1^2	...	d_1^m
f_2	d_2^1	d_2^2	...	d_2^m
...
f_m	d_m^1	d_m^2	...	d_m^m

где d_i^j - степень близости j -го решения к максимальному значению i -й целевой функции.

Шаг 2. Решить следующую игровую задачу одним из методов линейного программирования.

$$d^{m+1} = \sum_{i=1}^m \lambda_i d_i(\mathbf{X}) \Rightarrow \max_{\lambda}$$

при условиях

$$\sum_{i=1}^m \lambda_i d_i^1 \geq d^{m+1},$$

$$\sum_{i=1}^m \lambda_i d_i^2 \geq d^{m+1},$$

$$\dots$$

$$\sum_{i=1}^m \lambda_i d_i^m \geq d^{m+1},$$

$$\sum_{i=1}^m \lambda_i = 1, \forall \lambda_i \geq 0.$$

Шаг 3. Образовать новую функции свертки, используя оптимальные веса, найденные на шаге 2, и решить следующую задачу максимизации этой функции для получения нового альтернативного компромиссного решения \mathbf{X}^{m+1} .

$$d^{m+1}(\mathbf{X}) = \sum_{i=1}^m \lambda_i d_i(\mathbf{X}) \Rightarrow \max_{\mathbf{X}}$$

при условии $\mathbf{X} \in D$.

Шаг 4. Вычислить значения степеней близости нового решения к максимально возможным значениям целевых функций, $d_i^{m+1}, i = \overline{1, m}$.
Добавить колонку с этими значениями к таблице, построенной на шаге 1.

Шаг 5. Представить ЛПР новую таблицу и спросить, предпочитает ли он строго одно решение всем другим m -решениям. Если да, то

идти на шаг 6. Иначе просить ЛПР отметить наименее предпочитаемое решение. Заменить его новым решением, найденным на шаге 4, и вернуться на шаг 2.

Шаг 6. Останов.

Пример 10.7. Применим этот алгоритм к задаче из примера 10.5.

Шаг 0. а) $f_1(\mathbf{X}) = -2x_1 + x_2 \Rightarrow \max$

при условии $\mathbf{X} \in D$,

где D определяется системой

$$x_1 + x_2 \leq 8,$$

$$-x_1 + x_2 \leq 2,$$

$$0 \leq x_1 \leq 6, 0 \leq x_2 \leq 4.$$

Решение: $\hat{\mathbf{X}}_1 = (0, 2)$, $\hat{f}_1 = 2$.

$$f_2(\mathbf{X}) = 4x_1 - x_2 \Rightarrow \max$$

при условии $\mathbf{X} \in D$.

Решение: $\hat{\mathbf{X}}_2 = (6, 0)$, $\hat{f}_2 = 24$.

б) $f_1(\mathbf{X}) = -2x_1 + x_2 \Rightarrow \min$

при условии $\mathbf{X} \in D$.

Решение: $\check{\mathbf{X}}_1 = (6, 0)$, $\check{f}_1 = -12$.

$$f_2(\mathbf{X}) = 4x_1 - x_2 \Rightarrow \min$$

при условии $\mathbf{X} \in D$.

Решение: $\check{\mathbf{X}}_2 = (0, 2)$, $\check{f}_2 = -2$.

Шаг 1. Решения $\check{\mathbf{X}}_1$ и $\check{\mathbf{X}}_2$ принимаем за первоначальные эффективные решения \mathbf{X}^1 и \mathbf{X}^2 . Составляем таблицу степеней близости

	\mathbf{X}^1	\mathbf{X}^2	\hat{f}_i
f_1	1	0	2
f_2	0	1	24

Шаг 2. Решаем задачу линейного программирования:

$$d^3 = \lambda_1 d_1(\mathbf{X}) + \lambda_2 d_2(\mathbf{X}) \Rightarrow \max$$

при условиях

$$\lambda_1 + 0 \geq d^3,$$

$$0 + \lambda_2 \geq d^3,$$

$$\lambda_1 + \lambda_2 = 1,$$

$$\lambda_1, \lambda_2 \geq 0.$$

Решение: $\lambda_1 = 0.5$, $\lambda_2 = 0.5$.

Шаг 3. Составляем новую функцию свертки

$$d^3(X) = 0.5 \frac{f_1(X) - \check{f}_1}{\hat{f}_1 - \check{f}_1} + 0.5 \frac{f_2(X) - \check{f}_2}{\hat{f}_2 - \check{f}_2} =$$

$$= \frac{1}{182}(x_1 + 3x_2 + 85).$$

Решаем следующую задачу линейного программирования для нахождения нового компромиссного решения:

$$d^3(X) = \frac{1}{182}(x_1 + 3x_2 + 85) \Rightarrow \max$$

при условии $X \in D$.

Решение: $X^3 = (4, 4)$, $f_1(X^3) = -4$, $f_2(X^3) = 12$. На рис.10.17 это точка С.

Шаг 4. Вычисляем степени близости полученного решения

$$d_1^3 = \frac{f_1(X^3) - \check{f}_1}{\hat{f}_1 - \check{f}_1} = \frac{-4 + 12}{2 + 12} = 0.571,$$

$$d_2^3 = \frac{f_2(X^3) - \check{f}_2}{\hat{f}_2 - \check{f}_2} = \frac{12 + 2}{24 + 2} = 0.538$$

и показываем ЛПР три эффективных решения:

	X^1	X^2	X^3	\hat{f}_i
f_1	1	0	0.571	2
f_2	0	1	0.538	24

Шаг 5. Предположим, что ЛПР не устраивает ни одно из этих решений, а наименее предпочтительным он считает решение X^1 . Тогда это решение заменяем решением X^3 и возвращаемся на шаг 2.

Вторая итерация

Шаг 2. Решаем следующую задачу для определения новых весов:

$$d^4(X) \Rightarrow \max$$

при условиях

$$0.571 \lambda_1 + 0.538 \lambda_2 \geq d^4,$$

$$\lambda_2 \geq d^4,$$

$$\lambda_1 + \lambda_2 = 1,$$

$$\lambda_1, \lambda_2 \geq 0.$$

Оптимальные значения: $\lambda_1 = 0.447$, $\lambda_2 = 0.553$.

Шаг 3. Решаем задачу с этими весами:

$$d^4(X) = 0.447 d_1(X) + 0.553 d_2(X) =$$

$$= \frac{1}{24}(2x_1 + x_2 + 40) \Rightarrow \max$$

при условии $X \in D$.

Решение: $X^4 = (6.2)$, $f_1(X^4) = -10$, $f_2(X^4) = 22$. На рис.10.17 это вершина Е множества достижимости.

Шаг 4. Вычисляем степени близости нового решения

$$d_1^4 = 0.143, \quad d_2^4 = 0.923$$

и предъявляем его ЛПР вместе с оставшимися:

	X^3	X^2	X^4	\hat{f}_i
f_1	0.571	0	0.143	2
f_2	0.538	1	0.923	24

Шаг 5. Предположим, что ЛПР предпочитает новое решение всем другим. Идем на шаг 6.

Шаг 6. Останов. Наилучшее компромиссное решение нашей задачи: $X = (6.2)$, $f = (-10, 22)$, $d = (0.143, 0.923)$.

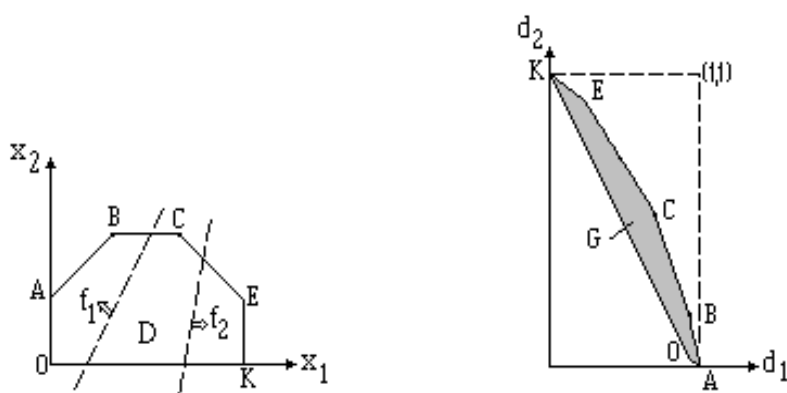


Рис. 10.17

Отметим некоторые особенности рассмотренного метода.

При его применении необходимо иметь в виду, что в соответствии с теоремой 5 максимизация линейной свертки (10.25) гарантирует получение эффективного решения только в случае положительности всех λ_i . Наличие нулевых значений весов может приводить к слабо эффективным решениям.

Из анализа процедуры поиска компромиссного решения следует, что область принятия решений сужается после каждой итерации алгоритма. При этом можно ожидать хорошей сходимости, так как каждое новое решение, предъявляемое ЛПР, является наиболее вероятным среди уже имеющихся, что обусловлено максиминным выбором весов.

Очевидным достоинством метода является использование естественного для ЛПР способа выражения своих предпочтений. Здесь отсутствуют какие-либо эвристические параметры (к ним относят параметры, значения которых должен назначать ЛПР).

Если модель исходной многокритериальной задачи линейна, то вся вычислительная часть интерактивного метода может быть реализована с помощью стандартных пакетов линейного программирования, что

означает возможность решения задач большой размерности. Подзадача определения оптимальных весов (шаг 2) остается линейной при любом виде исходной модели.

Наконец, заметим, что для получения очередного нового решения можно применить вместо линейной максиминную свертку. Такая замена может оказаться существенной для линейных задач, так как линейная свертка позволяет находить решения только в вершинах, в то время как максиминная – и на ребре или грани. И тем самым можно избежать возможных повторений решений.

10.2.2.3. Метод STEM

Если обобщить метод уступок, отказавшись от предварительного ранжирования критериев и допуская возможность уступок сразу по нескольким критериям, и добавить идею вычисления весов, то получим эвристическую процедуру STEM.

В качестве опорной для ЛПР точки используется идеальная точка. Поэтому предварительно решается m задач однокритериальной оптимизации, дающих идеальный вектор критериев $\hat{f}_i, i = \overline{1, m}$. Для определения очередного решения используется минимаксная свертка отклонений от идеального вектора:

$$\max_i [\lambda_i (\hat{f}_i - f_i(X))] \Rightarrow \min_X \quad (10.26)$$

Однако от ЛПР не требуется прямого задания весов. Они вычисляются по формулам:

$$\lambda_i = \begin{cases} 0, & i \in J \\ \frac{\pi_i}{\sum_{p=1}^m \pi_p}, & i \notin J \end{cases} \quad (10.27)$$

$$\pi_i = \begin{cases} \frac{\hat{f}_i - t_i}{\hat{f}_i} \cdot \frac{1}{|\nabla f_i(X^k)|}, & \hat{f}_i > 0 \\ \frac{t_i - \hat{f}_i}{t_i} \cdot \frac{1}{|\nabla f_i(X^k)|}, & \hat{f}_i \leq 0, \end{cases} \quad (10.28)$$

где J – множество индексов критериев, по которым будут вводиться уступки на данном шаге;

$|\nabla f_i(X^k)|$ – длина градиента i -й целевой функции в текущем решении X^k (для линейных функций от X^k не зависит);

t_i – минимальное значение i -го критерия из всех m решений, полученных при максимизации отдельных критериев на предварительном этапе.

Как нетрудно видеть, коэффициенты π_i учитывают относительный размах и скорость изменения критерия.

Первоначально $J=\emptyset$. Если очередное решение X^k не устраивает ЛПР, он должен указать критерии, которые согласен уменьшить (формируется множество J), и дать по ним уступки Δ_i . Тогда вычисляются веса λ_i по формулам (10.27), (10.28) и решается задача (10.26) на новом (суженном) допустимом множестве, которое описывается условиями:

$$\begin{aligned} f_i(X) &\geq f_i(X^k) - \nabla_i, \quad i \in J, \\ f_i(X) &\geq f_i(X^k), \quad i \notin J, \\ X &\in D. \end{aligned}$$

Полученное решение X^{k+1} предъявляется ЛПР и от его заключения зависит продолжать или заканчивать процедуру поиска.

Как и в предыдущем методе, получаемые решения могут быть слабо эффективными.

Заметим, что в случае назначения уступок одновременно более чем одному критерию получаемый результат не дает ясного представления о взаимозамещаемости критериев, как в методе уступок, что может затруднить выбор последующих действий ЛПР.

Метод применим как к линейным, так и нелинейным многокритериальным задачам, но он не гарантирует сходимости к решению, оптимальному в смысле максимизации функции полезности, что обусловлено его эвристичностью.

10.2.2.4. Метод взвешенных метрик Чебышева

Многие интерактивные методы заключаются в том, что строят последовательность уменьшающихся подмножеств множества Парето, что позволяет ЛПР более тонко исследовать область наиболее привлекательных решений. Ряд таких методов использует для этого различные способы сужения интервалов весов критериев. К числу последних относится и алгоритм взвешенных метрик Чебышева. Метрики выступают в качестве обобщенных критериев для нахождения эффективных решений по определенным на предшествующем шаге алгоритма весам частных критериев.

Как известно, метрики характеризуют расстояние между двумя точками в многомерном пространстве. В рассмотренном методе они

измеряют расстояние в критериальном m -мерном пространстве между недостижимым вектором критериев $\hat{\mathbf{Y}}^*$ и множеством достижимости G . Чтобы исключить случаи нулевых расстояний (неединственность критериального вектора с максимальным значением i -го критерия, принадлежность идеального вектора \hat{f} множеству G), в методе используется не идеальный вектор, а такой вектор $\hat{\mathbf{Y}}^*$, что

$$\hat{\mathbf{Y}}^* = \hat{f} + \varepsilon_i,$$

где $\varepsilon_i \geq 0$, а \hat{f}_i , как и раньше, - максимальное значение i -й целевой функции на $X \in D$. Таким образом, всегда $\hat{\mathbf{Y}}^* \notin G$ и геометрически соответствующая точка в многокритериальном пространстве расположена правее и выше множества G .

Метрика Чебышева – это одна из семейства L_p -метрик, в которых расстояние в R^m задается в виде

$$\|\hat{\mathbf{Y}}^* - \mathbf{Y}\|_p = \left[\sum_{i=1}^m |\hat{y}_i^* - y_i|^p \right]^{\frac{1}{p}}, p = \{1, 2, 3, \dots\} \cup \{\infty\}. \quad (10.29)$$

Наиболее применяемые значения p : 1, 2 и ∞ . Метрика L_∞ и называется чебышевской. Для нее выражение (10.29) принимает вид

$$\|\hat{\mathbf{Y}}^* - \mathbf{Y}\|_\infty = \max_i \{|\hat{y}_i^* - y_i|\}. \quad (10.30)$$

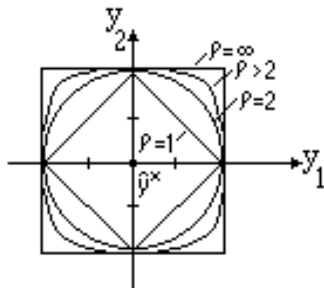


Рис. 10.18

Геометрическое место точек равных расстояний в L_p -метриках (уровень расстояний) представляет собой замкнутую линию (в многомерном случае - поверхность) с центром в точке $\hat{\mathbf{Y}}^*$. Влияние значений p на вид линий уровня показано на рис.10.18 (расстояние равно 2).

Если координатам придать неотрицательные веса λ_i , то получим взвешенные L_p -метрики (L_p^λ):

$$\|\hat{\mathbf{Y}}^* - \mathbf{Y}\|_p^\lambda = \left[\sum_{i=1}^m (\lambda_i |\hat{y}_i^* - y_i|)^p \right]^{\frac{1}{p}}, p = \{1, 2, 3, \dots\} \cup \{\infty\} \quad (10.31)$$

и в частности:

$$\|\hat{\mathbf{Y}}^* - \mathbf{Y}\|_\infty^\lambda = \max_i \{\lambda_i |\hat{y}_i^* - y_i|\}. \quad (10.32)$$

Очевидно, что при равенстве весов форма линий уровня не зависит от абсолютных значений λ_i . При неравных λ_i линии вытягиваются в направлении наименьших весов (рис. 10. 19).

Объединение метрик L_∞^λ и L_1 дает расширенную взвешенную метрику Чебышева $L_\infty^{\lambda, \rho}$:

$$\begin{aligned} \|\hat{\mathbf{Y}}^* - \mathbf{Y}\|_{\infty}^{\lambda, \rho} &= \|\hat{\mathbf{Y}}^* - \mathbf{Y}\|_{\infty}^{\lambda} + \rho \sum_{i=1}^m |\hat{y}_i^* - y_i| = \\ &= \max_i \{\lambda_i (\hat{y}_i^* - y_i)\} + \rho \sum_{i=1}^m (\hat{y}_i^* - y_i), \end{aligned} \quad (10.33)$$

где ρ - малое положительное число. Здесь опущены знаки модуля, т.к. для $\mathbf{Y} \in G$ всегда $\hat{y}_i^* - y_i \geq 0$. Как при этом изменяются линии уровня показано на рис.10.20.

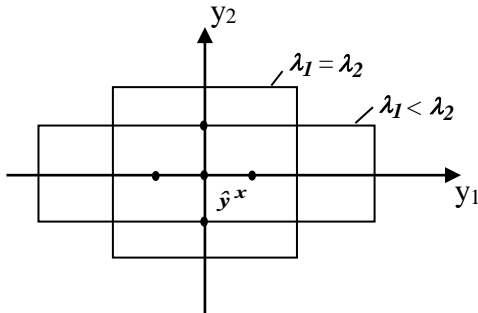


Рис.10.19

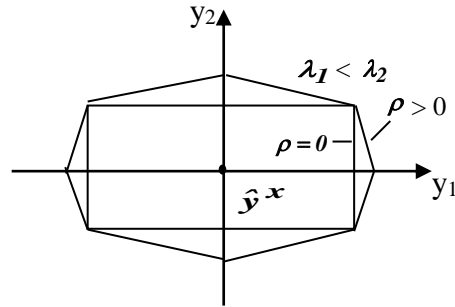


Рис.10.20

Минимизация расстояния между $\hat{\mathbf{Y}}^*$ и множеством G , представленного в одной из метрик, позволяет получить, по крайней мере, слабо эффективное решение (см. теорему 5). В случае взвешенной метрики Чебышева эта задача имеет вид

$$\max_i \{\lambda_i (\hat{y}_i^* - y_i)\} \Rightarrow \min_{y \in G}. \quad (10.34)$$

Введя новую переменную v , преобразуем ее к стандартной задаче минимизации:

$$\begin{aligned} v &\Rightarrow \min, \\ \lambda_i (\hat{y}_i^* - y_i) &\leq v, \quad i = \overline{1, m}; \\ y_i &= f_i(\mathbf{X}), \quad i = \overline{1, m}; \\ \mathbf{X} &\in D. \end{aligned} \quad (10.35)$$

В случае неединственности решения задачи (10.35) возможно получение доминируемых (слабо эффективных) критериальных векторов. Так на рис.10.21 результатом решения задачи (10.35) при некоторых λ_i может быть любая точка на отрезке ВС, но только точка С дает недоминируемый вектор \mathbf{Y} . Для исключения таких ситуаций предложено два способа.

Первый состоит в использовании расширенной взвешенной метрики. Благодаря наклону линий (поверхностей) уровня этой метрики исключается получение слабо эффективных решений. На том же множестве G , что и на рис.10.21, минимизация расстояния в расширенной метрике дает только недоминируемую точку С (рис.10.22). Очевидно, что

чем меньше ρ , тем меньше изменяется наклон линий уровня по сравнению с взвешенной метрикой. Значения ρ рекомендуется брать между 10^{-4} и 10^{-2} .

По аналогии с (10.34) и (10.35) минимизация расстояния в расширенной взвешенной метрике (10.33) сводится к решению

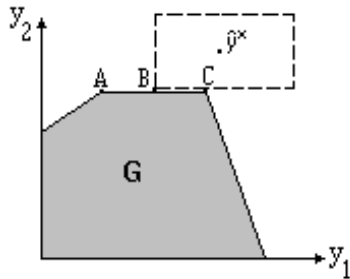


Рис. 10.21

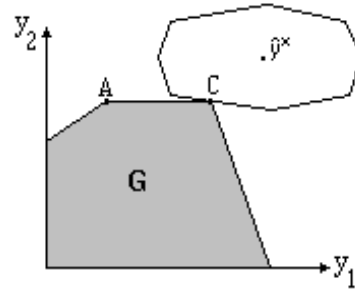


Рис. 10.22

следующей задачи

$$\begin{aligned} v + \rho \sum_{i=1}^m (\hat{y}_i^* - y_i) &\Rightarrow \min, \\ \lambda_i (\hat{y}_i^* - y_i) &\leq v, \quad i = \overline{1, m}; \\ y_i &= f_i(X), \quad i = \overline{1, m}; \\ X &\in D.. \end{aligned} \quad (10.36)$$

Опираясь на теорему 5, можно доказать, что в случаях, когда D – конечное дискретное множество или выпуклый многогранник, задача (10.36) всегда дает недоминируемый критериальный вектор и каждый недоминируемый (паретовский) вектор может быть получен в результате решения (10.36) единственным образом. Однако при других D (нелинейных или дискретных с бесконечным числом точек) отдельные недоминируемые векторы нельзя получить из задачи (10.36). Такие случаи интересны в основном теоретически, так как на практике встречаются редко.

Второй способ, обеспечивающий получение недоминируемого вектора критериев, заключается в переходе к лексикографической оптимизации. Вместо одноэтапной минимизации расстояния в расширенной взвешенной метрике сначала решается задача (10.35) и, если она дает неединственное решение, то на втором этапе минимизируется расстояние в метрике L_1 , то е.сть. ищется

$$\min \sum_{i=1}^m (\hat{y}_i^* - y_i)$$

при тех же условиях, что и в (10.35) или (10.36). При таком подходе отпадает необходимость в выборе значения ρ , а лексикографическая

оптимизация гарантирует получение эффективных решений независимо от свойств множества D и все недоминируемые векторы критериев могут быть вычислены единственным образом.

Теперь остановимся на определении значений λ_i , используемых при отыскании паретовских решений. Первоначально множество весовых векторов λ формируется случайным образом из диапазона $[0,1]$. После выбора ЛПР предпочтительного критериального вектора \bar{Y} пересчитывается соответствующий ему вектор весов. Дело в том, что один и тот же недоминируемый вектор \bar{Y} может быть получен при разных весах, например, как в случае, показанном на рис.10.23. В алгоритме в качестве вектора весов, порождающего \bar{Y} , рассматривается такой, который дает линию уровня, касающуюся G (в точке \bar{Y}) своей вершиной (рис.10.23,б). При этом компоненты вектора весов удовлетворяют следующему соотношению:

$$\lambda_i = \frac{1}{(\hat{y}_i^* - \bar{y}_i)} \cdot \left[\sum_{j=1}^m \frac{1}{(\hat{y}_j^* - \bar{y}_j)} \right]^{-1}, i = \overline{1, m}. \quad (10.37)$$

Эти значения весов становятся центрами более узких интервалов весов $[\underline{\lambda}_i, \bar{\lambda}_i]$, из которых снова случайным образом генерируется множество весовых векторов, порождающих новые задачи минимизации расстояния.

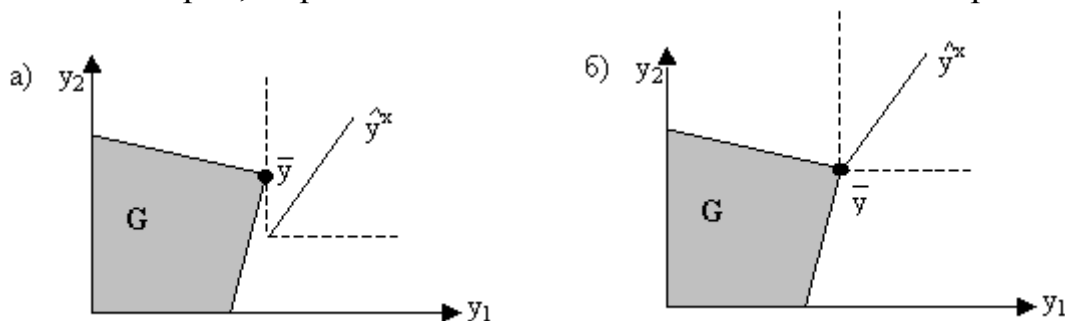


Рис.10.23

Такая процедура продолжается до получения удовлетворяющего ЛПР решения или выполнения заданного числа итераций.

В алгоритме используется целый ряд эвристических параметров. Их значения можно выбирать исходя из следующих рекомендаций. Величину ε_i , входящую в формулу для \hat{Y}^* , можно брать от 1 до 10% от \hat{y}_i^* или диапазона y_i на G . При этом желательно получать целые \hat{y}_i^* . Объем выборки P и число итераций t могут быть того же порядка, что и число критериев m . Коэффициент r сжатия множества Λ выбирается по соотношению

$$\sqrt[m]{\frac{1}{p}} \leq r \leq \sqrt[t]{w},$$

где w - ширина минимального интервала $[\underline{\lambda}_i, \overline{\lambda}_i]$ может быть взята примерно равной $1/m$.

Подготовка к решению может включать также масштабирование целевых функций, что целесообразно делать особенно в случаях, когда критерии измеряются в несопоставимых единицах.

Теперь опишем сам алгоритм, предложенный Штойером и Чу.

Шаг 1. Определить недостижимый критериальный вектор \hat{Y}^* .

Шаг 2. Положить номер итерации $l=0$, а начальный интервал весов $[\underline{\lambda}_i^{(0)}, \overline{\lambda}_i^{(0)}] = [0, 1]$ для всех i .

Шаг 3. Принять $l=l+1$ и построить множество

$$\Lambda^{(l)} = \{ \lambda \in R^m \mid \lambda_i \in [\underline{\lambda}_i^{(l)}, \overline{\lambda}_i^{(l)}], \sum_{i=1}^m \lambda_i = 1 \}.$$

Шаг 4. Случайным образом сгенерировать $50m$ весовых векторов из $\Lambda^{(l)}$.

Шаг 5. Из полученного множества весовых векторов выбрать $2P$ наиболее различающихся.

Шаг 6. Используя выбранные веса, решить $2P$ соответствующих задач минимизации расстояний (в расширенной взвешенной метрике – (10.36) или лексикографически). Результат – множество недоминируемых критериальных векторов.

Шаг 7. Из полученного множества выбрать P наиболее различающихся критериальных векторов.

Шаг 8. Выбранные векторы предъявить ЛПР, который должен выделить из них наиболее предпочтительный – $Y^{(l)}$.

Шаг 9. Пересчитать $\lambda^{(l)}$:

$$\lambda_i = \frac{1}{(\hat{y}_i^* - y_i^{(l)})} \cdot \left[\sum_{i=1}^m \frac{1}{(\hat{y}_i^* - y_i^{(l)})} \right]^{-1}, \quad i = \overline{1, m}.$$

Шаг 10. Определить новые (более узкие) интервалы весов:

$$[\underline{\lambda}_i^{(l+1)}, \overline{\lambda}_i^{(l+1)}] = [0, r^l], \quad \text{если} \quad \lambda_i^{(l)} - \frac{r^l}{2} \leq 0;$$

$$[\underline{\lambda}_i^{(l+1)}, \overline{\lambda}_i^{(l+1)}] = [1 - r^l, 1], \quad \text{если} \quad \lambda_i^{(l)} + \frac{r^l}{2} \geq 1;$$

$$[\underline{\lambda}_i^{(l+1)}, \overline{\lambda}_i^{(l+1)}] = [\lambda_i^{(l)} - \frac{r^l}{2}, \lambda_i^{(l)} + \frac{r^l}{2}] \quad \text{в остальных случаях.}$$

Здесь r^l означает 1-ю степень r .

Шаг 11. Если $l < t$ и ЛПР желает продолжить поиск, перейти к шагу 3, иначе - к шагу 12.

Шаг 12. Остановиться, если ЛПР согласен принять за окончательное решение векторы ($Y^{(l)}, X^{(l)}$). Иначе вернуться на шаг 3.

В приведенном алгоритме не уточняется способ фильтрации, используемый для отбора наиболее различающихся векторов. Один из возможных приемов – кластерный анализ с заданным числом кластеров, равным количеству требуемых объектов, с последующим оставлением одного представителя от каждого кластера. Следует заметить, что выбор способа фильтрации не имеет существенного значения. Важно лишь в качестве начальной точки фильтрации на шаге принимать вектор, выбранный ЛПР на предыдущей итерации. Тогда среди новых P векторов не будет близких к $Y^{(i)}$ и, следовательно, выборка будет наиболее информативной.

При реализации алгоритма могут быть полезны следующие практические советы.

ЛПР будет легче ориентироваться в представляемых ему результатах, если на экране дисплея показать графически диапазоны изменения значений критериев на паретовском множестве. Предъявляемые ЛПР для оценки критериальные вектора можно наносить на эти диапазоны, отображая каждый вектор своим цветом (в виде точек или ломанных, соединяющих отображаемые значения критериев данного вектора).

Чем больше число решений предъявляется ЛПР на одной итерации, тем быстрее сходится алгоритм. Но не следует забывать об информационных возможностях ЛПР, которые во многих работах оцениваются формулой «семь плюс-минус два». Конкретное число определяется с учетом опыта и характера ЛПР.

Сходимость алгоритма может быть повышена, если ЛПР введет в задачу минимизации расстояний ограничения на критерии снизу (они должны быть больше минимальных на эффективном множестве, иначе множества выбора не будет сокращено). Это становится необходимым, если ЛПР не приемлет значения некоторых критериев ниже определенных уровней.

10.2.2.5. Прогрессивный алгоритм принятия многокритериальных решений

Метод, излагаемый в этом параграфе, стоит особняком от рассмотренных выше, так как разработан для задач с дискретными альтернативами. И по этой же причине он включен в конспект, чтобы дать некоторое представление о возможных подходах к решению таких задач.

На практике нередко возникают ситуации, когда при решении многокритериальной задачи с дискретными альтернативами трудно или дорого, или вообще невозможно (недоступно) рассматривать одновременно все альтернативы. Например, при подборе кандидатов на должность невозможно быстро охватить всех потенциально желающих

занять эту должность, а длительное ожидание может быть неприемлемо. Тогда выбор производится поэтапно до определения подходящей кандидатуры (и, как правило, не до исчерпания всех альтернатив). Аналогичная проблема имеет место при конкурсном отборе проектов: если не находится проект, устраивающий комиссию, то может объявляться дополнительный конкурс. На этой концепции и основан прогрессивный алгоритм.

Предлагается сначала исследовать начальную (доступную) выборку альтернатив и при невозможности получить на ней приемлемое для ЛПР решение производится добавление альтернатив с последующим продолжением поиска. Отличительной особенностью метода является и то, что хотя он базируется на функции ценности ЛПР, определение или построение точной функции ценности не требуется.

Кратко алгоритм можно описать следующим образом. ЛПР предъявляется начальная выборка альтернативных решений. Сравнивая попарно все или часть альтернатив, ЛПР должен выразить свои предпочтения по каждой рассмотренной паре. Эти данные используются в формальных тестах, представляющих собой задачи линейного программирования, с помощью которых определяется класс функций, к которому можно отнести функцию ценности ЛПР. Из имеющегося набора альтернатив находится наилучшая. Затем определяются области, содержащие возможно или наверняка лучшие альтернативы, чем те что уже рассмотрены. С помощью статистического моделирования вычисляются вероятностные оценки для таких альтернатив. При этом используется предлагаемое или известное распределение генеральной совокупности, из которой берутся выборки. Эти оценки, являющиеся границами истинной вероятности отыскания в последующем лучших альтернатив, предъявляются ЛПР, который по ним решает продолжать или заканчивать поиск. В случае продолжения поиска добавляются новые альтернативы и процедура повторяется.

Теперь остановимся на некоторых деталях алгоритма. По информации ЛПР о его предпочтениях на парах альтернатив из начальной выборки формируется множество

$$P = \{(y, z) | y \in Y, z \in Y \text{ \& } y \succ z\},$$

где y, z – пара альтернатив в критериальном пространстве из имеющейся выборки $Y \subset G$. Предпочтения в P должны быть транзитивны, что следует из свойства функции ценности возрастать с увеличением предпочтения. Как правило, информация от ЛПР оказывается неполной и тогда применяют способы, позволяющие дополнить множество P , например, использующие свойство транзитивности.

Рассматривается три класса функций, одному из которых может принадлежать истинная функция ценности $u(y)$: линейные (U_L), квазивогнутые (U_Q) и общего вида (U_G). Очевидно, что $U_L \subset U_Q \subset U_G$. Как

отмечалось выше, для идентификации $u(y)$ применяются специальные тесты. Первый тест позволяет принять или отвергнуть гипотезу о принадлежности $u(y)$ классу U_L . Для этого решается следующая задача ЛП:

$$\begin{aligned} \varepsilon &\Rightarrow \max, \\ \lambda(y-z) &\geq \varepsilon, \quad \forall (y, z) \in P, \\ \lambda &\geq 0. \end{aligned} \quad (10.38)$$

Пусть $y \succ z$ для ЛПР, если $u(y) \geq u(z) + \delta$, где $\delta > 0$, характеризует ощутимое различие альтернатив. Тогда линейная $u(y)$ согласуется с P , то есть $u(y) \in U_L$, если $\varepsilon^* = \max \varepsilon \geq \delta$. При $\varepsilon^* < \delta$ гипотеза линейности функции ценности отвергается и проверяется гипотеза, что $u(y) \in U_Q$ согласуется с P , путем решения линейной задачи, подобной (10.38), но более детализированной по областям предпочтений. Если второй тест дает $\varepsilon^* < 0$, то и эта гипотеза будет отвергнута, и остается признать, что $u(y) \in U_G$.

Целесообразность продолжения поиска с дополнением выборки определяется ЛПР после получения оценки возможности нахождения альтернатив, лучших чем в текущей выборке. Очевидно, что при известной функции ценности $u(y)$ множество возможных альтернатив, лучших чем в Y , определяется как

$$R(Y) = \{z | z \in G \text{ \& } u(z) > u(y) \text{ для } \forall y \in Y\}.$$

Однако оно не может быть однозначно определено, когда известен лишь класс U , которому принадлежит функция ценности. Поэтому в таких случаях рассматриваются следующие множества:

-множество действительно лучших альтернатив

$$R_M(Y) = \{z | z \in G \text{ \& } u(z) > u(y) \text{ для } \forall y \in Y \text{ и } \forall u \in U\};$$

-множество действительно худших альтернатив

$$R_L = \{z | z \in G \text{ \& } \exists y \in Y \text{ такой, что } u(z) \leq u(y) \text{ для } \forall u \in U\};$$

-множество возможно лучших альтернатив

$$R_L^C = G - R_L.$$

Очевидно, что $R_M \subseteq R_L^C(Y)$, но при известной $u(y)$ эти множества совпадают.

Для классов U_L и U_Q имеются тесты в виде задач ЛП, позволяющие отнести рассматриваемую альтернативу $z \in G$ ($z \notin Y$) к множеству R_L или R_M . В случае $u \in U_Q$ идентификация альтернатив из-за отсутствия тестов сильно ограничена: только при установлении соотношения доминирования (условия достаточного, но не необходимого) можно отнести рассматриваемую альтернативу к одному из этих множеств. Заметим, что и при квазивогнутых функциях тест, дифференцирующий альтернативы, дает также только достаточные условия. Поэтому в классах U_Q и U_G нельзя точно идентифицировать множества R_M и R_L .

Для определения оценки вероятности нахождения лучших альтернатив (вне исследованной выборки) новое множество S альтернатив

генерируется (имитируется) случайным образом по данной функции распределения вероятностей. В соответствии с ранее определенным классом функции ценности идентифицируются альтернативы из S , относящиеся к множествам R_m и R_L^C . Отношение числа элементов множеств R и S дает приближенные значения вероятностей нахождения действительно и возможно лучших альтернатив. При уровне риска 5% доверительный интервал для неизвестной точной вероятности p определяется неравенством

$$\hat{p} - 1.96\sqrt{\hat{p}(1 - \hat{p})/n} \leq p \leq \hat{p} + 1.96\sqrt{\hat{p}(1 - \hat{p})/n},$$

где \hat{p} – оценка, n – число элементов в S . Это соотношение позволяет находить доверительный интервал по n или, наоборот, n по желаемой точной оценки (величине доверительного интервала).

На практике ЛПР интересует не столько значение p , сколько вероятность нахождения по крайней мере одной лучшей альтернативы и размер дополнительной выборки m , позволяющей уверенно (в статическом смысле) найти хотя бы одну лучшую альтернативу. Если задать нижнюю границу ν для вероятности нахождения хотя бы одной лучшей альтернативы, то размер дополнительной выборки получим из неравенства

$$m \geq \ln(1 - \nu) / \ln(1 - p).$$

Так как мы знаем не p , а доверительный интервал, m определяется по двум значениям: нижней и верхней границам p .

Апробирование тестов, используемых в прогрессивном алгоритме, показало, что тест на квазивоугнутость функции ценности можно применять при числе критериев, не превышающем трех. Более того, установление квазивоугнутости не обязательно приведет к сужению границ вероятности нахождения лучшей альтернативы. Эти границы могут быть сужены при большем числе критериев, если оказывается справедливой гипотеза линейности. Но сила тестов ослабевает при 4-х и более критериях. Поэтому рекомендуется использовать в основном тест на линейность, так как в большинстве случаев знание только верхней границы вероятности достаточно для принятия решения об окончании или продолжении поиска. Минимальный размер начальной выборки, приемлемый для получения минимально необходимой информации о предпочтениях, составляет 8-10 альтернатив.

В заключение отметим, что прогрессивный алгоритм получает свое дальнейшее развитие на основе теории перспективности [40].

10.2.3. Построение эффективного множества

Очевидно, что если ЛПР показать эффективное множество в целом, то он сам сможет выбрать тот эффективный вектор, который предпочитает

всем остальным. В таком случае для решения задачи не потребуется извлекать знания о предпочтениях ЛПР. Этот подход особенно удобен, если имеет место групповое принятие решений. Однако при числе критериев свыше двух построение эффективного множества оказывается непростой задачей. Дело не только в невозможности многомерного представления всего множества, но и в том, что даже для линейных задач эффективное множество является невыпуклым (часть границы выпуклого многогранного множества).

Рассмотрим кратко некоторые способы построения эффективных множеств. Для линейных многокритериальных задач эффективное множество векторов критериев полностью определяется входящими в него вершинами множества достижимости G (для множества эффективных решений – эффективными вершинами D). Применяя один из методов свертки с весами и решая задачу с разными ненулевыми значениями весов (построив сетку весов или параметрически), можно найти многие (реже – все) эффективные вершины. Заметим, что линейная свертка дает только вершины, другие способы свертки могут давать эффективные точки на ребре или грани. Чтобы получить представление о всем эффективном множестве, следует после нахождения очередного оптимального (эффективного) решения найти все альтернативные оптимальные решения, которые существуют, если среди оценок небазисных переменных есть нулевые (вводя поочередно эти переменные в базисное решение, можно получит все вершины эффективной грани). Подробно и наглядно эти вопросы рассмотрены в книге Р. Штойера [35]. При воссоздании эффективного множества по вершинам нужно помнить, что из-за его невыпуклости не всякая выпуклая комбинация эффективных вершин дает эффективную точку. Так как на рис.10.24 выпуклая комбинация эффективных вершин порождает треугольник ABC , хотя эффективными являются только его стороны AB и BC . Для наглядности представления эффективное множество линейной задачи можно проектировать на плоскость – оно будет иметь вид одного или нескольких многоугольников с общими ребрами и вершинами (как развертка трехмерной фигуры). У каждой вершины может быть выведен соответствующий критериальный вектор. Выбор наилучшего решения облегчится, если ЛПР, перемещая указатель по представленному эффективному множеству, будет получать значения критериального вектора в точке указателя или иное представление, например, в виде графиков, о поведении вектора.

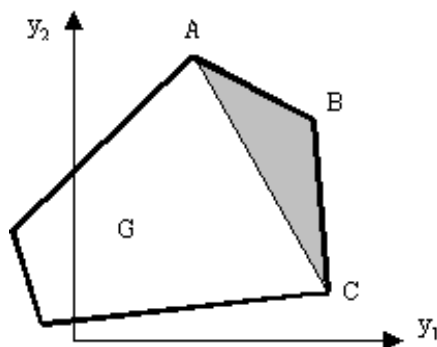


Рис. 10.24

В отличие от описанного способа метод ограничений может применяться как для линейных, так и нелинейных многокритериальных задач. В критериальном пространстве строится сетка, значения критериев в узлах которой выступают в качестве ограничений на соответствующие критерии. Для этого сначала решается m однокритериальных задач (на максимум каждого частного критерия), которые дают диапазоны изменения каждого критерия $[\hat{f}_i, \check{f}_i]$ на эффективном множестве. Сетка, накладываемая на этот интервал, включает $S+1$ узел:

$$f_i^s = \check{f}_i + \frac{s}{S}(\hat{f}_i - \check{f}_i), \quad s = 0, 1, \dots, S. \quad (10.39)$$

Затем для каждого критерия решаются задачи:

$$\begin{aligned} f_i(X) &\Rightarrow \max, \\ f_k(X) &\geq f_k^s, \quad k = \overline{1, m}, k \neq i; \\ X &\in D \end{aligned} \quad (10.40)$$

для всех возможных сочетаний f_k^s из (10.39). Чем мельче сетка, тем точнее можно представить эффективное множество (рис.10.25). Основным недостатком данного метода является большое число решаемых задач (10.40). При этом часть из них может оказаться неразрешимой, а часть может дать лишь слабо эффективные точки.

Так как не все процедуры гарантируют получение эффективных решений, для проверки эффективности следует использовать условия оптимальности, рассмотренные в разделе 10.1.4. Если решение

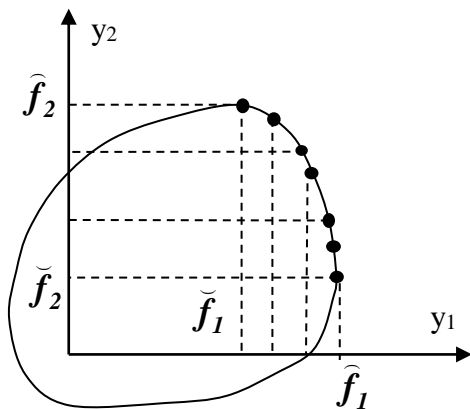


Рис.10.25

удовлетворяет необходимым условиям, то гарантировать его эффективность нельзя. Но если оно не выполняется, то решение однозначно не эффективно. Удовлетворение достаточных условий гарантирует эффективность решения. В противном случае ничего определенного об эффективности решения сказать нельзя. Наконец решение всегда эффективно, если оно удовлетворяет как необходимым, так и достаточным условиям.

В задачах с выпуклым множеством D допустимых решений и линейными критериями множество достижимости G выпукло. В таких случаях легче построить все множество G , чем невыпуклое эффективное множество. В то же время G дает полное представление о структуре последнего. Это преимущество особенно проявляется в линейных многокритериальных задачах, т.к. многогранное выпуклое множество G можно описать в виде

$$G = \{f \in E^m \mid Hf \leq h\}, \quad (10.41)$$

удобном для его построения. Имея G , легко получить любое двумерное сечение, необходимое ЛПР, или его проекции, которые выводятся на экран, обеспечивая более полное представление обо всем множестве достижимости и о его эффективном подмножестве. Можно также использовать аппроксимацию G более простой геометрической фигурой, имеющей ту же эффективную границу, что и G .

Для построения множества G в виде (10.41) применяют теорию линейных неравенств. Рассмотрим неравенство

$$Av + Bw \leq C, \quad (10.42)$$

в котором v и w – k и l -мерные векторы переменных. Очевидно, что система (10.42) порождает в $(k+l)$ -мерном пространстве некоторое многогранное множество M . Построим на нем множество M_w таких точек $w' \in E^l$, что для каждой найдется хотя бы одна точка $v' \in E^k$, образующая пару $\{v', w'\}$, удовлетворяющую (10.42). Поясним построение M_w на примере простой системы с $k=l=1$:

$$\begin{aligned} 1) v + w \leq 3, \quad 2) -v + w \leq 5, \quad 3) -w \leq -0.5, \\ 4) v - 2w \leq 0, \quad 5) -2v - w \leq -1, \quad 6) v \leq 2. \end{aligned}$$

Множество M_w получим, если исключим переменную v , для чего следует сложить пары неравенств с противоположными знаками при v (предварительно неравенство делится на коэффициент при v). В результате этих преобразований имеем:

$$\begin{aligned} 1)+2): 2w \leq 8, \quad 1)+5): 0.5w \leq 2.5, \quad 2)+4): -w \leq 5, \quad 2)+6): w \leq 7, \\ 3)-w \leq -0.5, \quad 4)+5): -2.5w \leq 0.5, \quad 5)+6): -0.5w \leq 1.5. \end{aligned}$$

Совместное решение неравенств и дает множество M_w :

$$0.5 \leq w \leq 4,$$

которое представляет собой отрезок оси w (рис.10.26), то есть ортогональную проекцию M на пространство E^l (здесь E^1). Этот метод может быть улучшен путем использования процедуры фильтрации несущественных ограничений в описании M_w .

Применительно к линейной задаче с n переменными и m критериями многогранное множество в пространстве E^{n+m} представляется в виде

$$Z = \{X, f \in E^{n+m} \mid f = C \cdot X, A \cdot X \leq B\}.$$

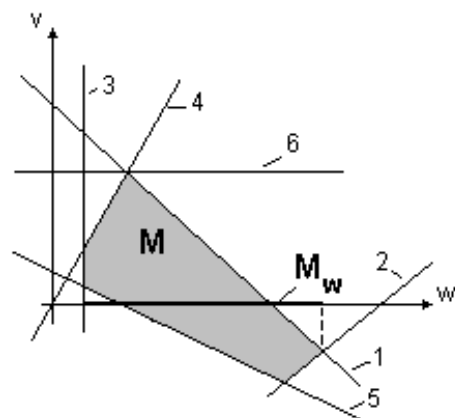


Рис. 10.26

Очевидно, что интересующее нас множество G (аналог M_w) является ортогональной проекцией Z на критериальное пространство E^m и может быть построено рассмотренным методом. Как отмечалось выше, в сложных случаях вместо точного множества G легче представить его аппроксимацию, но при этом упрощенное множество должно иметь то же эффективное множество, что и G . Более подробное рассмотрение этих вопросов выходит за рамки конспекта. В качестве иллюстрации на рис.10.27 показаны двумерные сечения (проекции) множества G для задачи с 4 критериями.

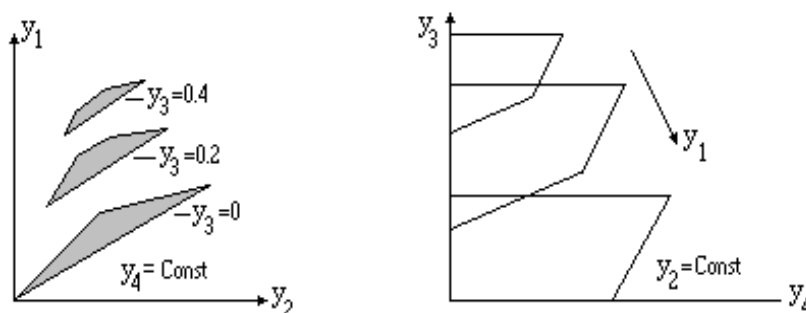


Рис. 10.27

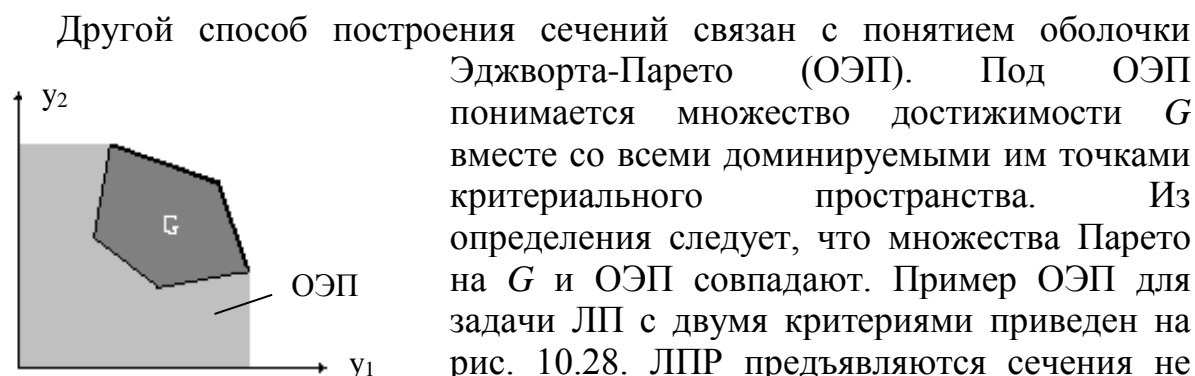


Рис. 10.28

Другой способ построения сечений связан с понятием оболочки Эджворта-Парето (ОЭП). Под ОЭП понимается множество достижимости G вместе со всеми доминируемыми им точками критериального пространства. Из определения следует, что множества Парето на G и ОЭП совпадают. Пример ОЭП для задачи ЛП с двумя критериями приведен на рис. 10.28. ЛПР предьявляются сечения множества G , а оболочки Эджворта-Парето. В [21] показано, что при этом упрощается построение сечений и, что особенно важно, на таких сечениях легче анализировать подмножества Парето.

В практических приложениях методов решения многокритериальных задач редко используется какой-либо один подход в чистом виде, а комбинация подходов позволяет лучше приспособиться к особенностям конкретной проблемы и ЛПР. При этом может оказаться особенно плодотворным сочетание интерактивных процедур и методов представления достижимого (эффективного) множества с использованием средств мультимедиа.

СПИСОК ЛИТЕРАТУРЫ

1. Акоф Р., Сасиени М. Основы исследования операций. -М.:Мир, 1971.
2. Акулич И.Л. Математическое программирование в примерах и задачах: Учебное пособие.-М.:Высш.шк.,1993.-336 с.
3. Аоки М. Введение в методы оптимизации. -М.:Наука, 1977.
4. Аронович А.Б. Сборник задач по исследованию операций: Учеб. пособие для студентов экон. спец. вузов/ Аронович А.Б., Афанасьев М.Ю., Суворов Б.П.. -М.: Изд-во Моск.ун-та, 1997. -253 с
5. Базара М., Шетти К. Нелинейное программирование. Теория и алгоритмы.-М.:Мир,1982.-583 с.
6. Батищев Д.И. Генетические алгоритмы решения экстремальных задач: Учеб.пособие. -Воронеж, 1995. -69 с.
7. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. -М.:Наука,1965.
8. Вагнер Г. Основы исследования операций (В 3-х томах)-М.:Мир,-т.1. 1972-335с.-т.2. 1973.-488с.-т.3. 1973.-501 с.
9. Вентцель Е.С., Исследование операций.-М.:Сов.радио, 1972.-552 с.
10. Вентцель Е.С. Исследование операций. Задачи, принципы, методология.-2-е изд., стер -М.: Высш. шк., 2001 .-208 с.
11. Вилкас Э.И. .Майминас Е.З. Решения: теория, информация, моделирование.- М.: Радио и связь, 1981.- 328 с.
12. Волков И. К., Загоруйко Е. А. Исследование операций: Учеб. для вузов/ ред. В.С.Зарубин.-2-е изд -М.: Изд-во МГТУ, 2002.-435 с.
13. Вошинин А.П., Сотиров Г.Р. Оптимизация в условиях неопределенности.- Изд-во МЭИ(СССР), Техника(НРБ), 1989.- 224 с.
14. Гермейер Ю.Б. Введение в теорию исследования операций. М: Наука,1971.- 383 с.
15. Гольдштейн А.Л. Исследование операций: Многокритериальные задачи: Конспект лекций.-Пермь: ПГТУ, 1995.-72 с.
16. Гольдштейн А.Л. Задачи и методы исследования операций: Учеб. пособие/Перм. гос. тех. ун-т, Пермь, 2000. Ч.1.-114 с.
17. Дегтярев Ю.И. Системный анализ и исследование операций: Учеб. для студентов вузов, обучающихся по спец. "Автоматизир. системы обработки информ. и упр.". -М.: Высш.шк., 1996. -335 с
18. Емельянов С.В., Ларичев О.И. Многокритериальные методы принятия решений.- М.: Знание, 1985.- 32 с.
19. Зайченко Ю.П., Шумилова С.А. Исследование операций: Сборник задач. -Киев:Вища школа,1990.
20. Кини Р.Л.,Райфа Х. Принятие решений при многих критериях: предпочтения и замещения М.: Радио и связь, 1981.- 560 с.
21. Компьютер и поиск компромисса. Метод достижимых целей / Лотов

- А.В. и др. –М.: Наука, 1997. –239 с.
22. Кузнецов А.В. Руководство к решению задач по математическому программированию: Учеб. пособие для вузов/ Кузнецов А.В., Холод Н.И., Костевич Л.С., ред. А.В.Кузнецов.-2-е изд., перераб. и доп - Минск: Вышэйш. шк., 2001 .-448 с.
 23. Ларичев О.И. Объективные модели и субъективные решения. М.: Наука,1987.- 143 с.
 24. Ларичев О.И. Теория и методы принятия решений, а также Хроника событий в Волшебных Странах: Учебник. –М.: Логос, 2000. –296 с.
 25. Моисеев Н.Н. Математические задачи системного анализа. М.: Наука, 1981.- 488 с.
 26. Муртаф Б. Современное линейное программирование,-М.:Мир,1984.- 224 с.
 27. Орловский С.А, Проблемы принятия решений при нечеткой исходной информации.-М.:Наука,1981,-208 с.
 28. Пантелеев А.В. Методы оптимизации в примерах и задачах: Учеб. пособие/А.В.Пантелеев, Т.А.Летова.–М.: Высш. шк., 2002.–544 с.
 29. Перегудов Ф.И.,Тарасенко Ф.П. Основы системного анализа. -Томск, 1997. -389 с
 30. Подиновский В.В., Гаврилов В.М. Оптимизация по последовательно применяемым критериям.- М.: Сов. радио, 1975.- 192 с.
 31. Подиновский В.В., Ногин В.Д. Парето-оптимальные решения многокритериальных задач.- М.: Наука, 1982.- 256 с.
 32. Современное состояние теории исследования операций. Под ред. Н.Н.Моисеева.- М.: Наука,1979.- 464 с.-(Оптимизация и исследование операций).
 33. Таха Х.А. Введение в исследование операций. - М.: Вильямс, 2001.- 911 с.
 34. Теория выбора и принятия решений: Учебное пособие.- М.: Наука, 1982.- 328 с.
 35. Фишберн П. Теория полезности для принятия решений.- М.: Наука. 1978.- 352 с.
 36. Штойер Р. Многокритериальная оптимизация. Теория, вычисления, и приложения.- М.: Радио и связь,1992.- 504 с.
 37. Юдин Д.Б., Гольштейн Е.Г. -Задачи и методы линейного программирования.-М:Сов.радио,1964.-736 с.
 38. Gibson K., Badinelli R. A comparison of Interactive multipleobjective decision making procedures// Comput. Opns. Res.,1987, vol.14,No.2,pp.97-105.
 39. Evren R. Interactive compromise programming//J.Opl.Res.Soc., 1987, vol.38,No.2,pp.163-172.
 40. Korhonen P. Using Harmonious Houses for Visual pairwise comparison of multiple criteria alternatives// Dec. Support system, 1991.vol.7,pp.47-54.

41. Korhonen P., Moskowitz H., Salminen P., Wallenius J. Further developments and tests of a progressive algorithm for multiple criteria decision making/Y Operations research, 1993, vol.41, No.6, pp.1033-1045.