Б.А. ЕСИПОВ

МЕТОДЫ ОПТИМИЗАЦИИ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

2007

СГАУ

САМАРА

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»

Б.А. ЕСИПОВ

МЕТОДЫ ОПТИМИЗАЦИИ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Утверждено Редакционно-издательским советом университета в качестве учебного пособия

САМАРА
Издательство СГАУ
2007

УДК 519.8
ББК 22.1
Е 833

Инновационная образовательная программа «Развитие центра компетенции и подготовка специалистов мирового уровня в области аэрокосмических и геоинформационных технологий»

Рецензенты: д-р техн. наук, проф. А. Н. Коварцев,
д-р физ.-мат. наук, проф. С. Я. Шатских

Есипов Б.А.

Е 833 **Методы оптимизации и исследование операций**: учеб. пособие / *Б.А.Есипов.* – Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2007. – 180 с. : ил.

ISBN 978-5-7883-0640-7

В учебном пособии дано краткое изложение основных методов оптимизации и исследования операций. Материал пособия основан на лекциях, читаемых автором для студентов факультета «Информатика», и соответствует программам курсов «Методы оптимизации», «Теория игр и исследование операций», «Теория принятия решений». Настоящее пособие является вспомогательным материалом к прослушиваемому курсу лекций, поэтому применяется конспективный стиль изложения. Приведены алгоритмы и примеры их работы, а также рисунки, поясняющие суть методов, что позволяет студентам разрабатывать компьютерные программы для выполнения индивидуальных заданий курсовой работы.

Предназначено для студентов специальностей «Прикладная математика и информатика», «Автоматизированные системы обработки информации и управления», а также аспирантов и специалистов, интересующихся методами оптимизации и исследования операций. Подготовлено на кафедре «Информационные системы и технологии» Самарского государственного аэрокосмического университета.

УДК 519.8
ББК 22.1

ISBN 978-5-7883-0640-7

© Есипов Б.А., 2007
© Самарский государственный аэрокосмический университет, 2007

Оглавление

Введение 6
1. Методология системного анализа и исследование операций 7
1.1. Системный анализ, система, оптимизация 7
1.2. Схема операционного проекта 8
1.3. Особенности математического моделирования операций 11
1.4. Постановка задачи исследования операций в детерминированном случае и в условиях неопределенности 12
1.5. Пример математического моделирования операции (задача о краске) 13
2. Линейное программирование (ЛП) 16
2.1. Общая и основная задачи ЛП 17
2.2. Геометрическая интерпретация задачи ЛП 19
2.3. Идея симплекс-метода решения задачи ЛП 21
2.4. Симплекс-таблица, стандартный алгоритм симплекс-преобразования 23
2.5. Алгоритм отыскания опорного решения задачи ЛП 25
2.6. Алгоритм отыскания оптимального решения задачи ЛП 26
2.7. Алгоритм получения первого базисного решения с использованием симплекс – процедуры (метод искусственного базиса) 29
2.8. Вырожденная задача ЛП 31
2.9. Двойственная задача ЛП 32
3. Транспортные задачи (ТЗ) 35
3.1. Математическая модель ТЗ по критерию стоимости 35
3.2. Нахождение опорного плана транспортной задачи 36
3.3. Оптимизация плана ТЗ, распределительный метод 38
3.4. Метод потенциалов решения ТЗ 40
3.5. Решение ТЗ с неправильным балансом 43
3.6. ТЗ по критерию времени, типы критериев 45
4. Дискретное программирование 48
4.1. Особенности задач дискретного программирования 49
4.2. Примеры моделей задач дискретного программирования 50
4.2.1. Задача о покрытии 54
4.2.2. Задача о коммивояжёре 54
4.2.3. Задача о раскрое материала 55
4.2.4. Задача о ранце 58
4.3. Алгоритм решения задачи о ранце 58
4.4. Решение задач ЛЦП методом отсечений Гомори 61
4.5. Метод ветвей и границ (МВГ) 66

4.6. Алгоритм МВГ для задачи ЛЦП 68
4.7. Алгоритмы решения задач булевского программирования 72
5. Динамическое программирование (ДП) 78
5.1 Принцип оптимальности Р.Беллмана 79
5.2. Решение графовых задач на основе принципа Беллмана 81
5.3. Функциональное уравнение Беллмана 82
5.4. Задачи распределения ресурсов 84
5.4.1. Классическая задача распределения ресурсов 84
5.4.2. Неоднородные этапы и распределение ресурсов по n отраслям 85
5.4.3. Распределение ресурсов с резервированием 85
5.4.4. Распределение ресурсов с «вложением доходов» 87
5.5. Расширение модели задач динамического программирования 89
5.6. Пример решения задачи распределения ресурсов 91
5.7. Эффективность динамического программирования 93
6. Нелинейное программирование 95
6.1. Особенности задач нелинейного программирования 95
6.2. Прямые методы одномерной оптимизации нелинейных функций без ограничений 97
6.3. Градиентные методы многомерной оптимизации 99
6.3.1. Классический градиентный метод 100
6.3.2. Покоординатный метод 101
6.3.3. Метод наискорейшего спуска и его модификации 101
6.4. Метод деформируемого многогранника Нелдера-Мида 101
6.5. Задача НЛП с ограничениями-равенствами 103
6.6. Выпуклое НЛП 106
6.7. Теорема Куна-Таккера для выпуклого НЛП 108
6.8. Квадратичное программирование 109
6.9. Методы возможных направлений 114
6.10. Метод проекции градиента 118
6.11. Методы штрафных и барьерных функций 121
6.12. Метод скользящего допуска 127
7. Особенности современной теории принятия оптимальных решений 130
7.1. Общая постановка задачи принятия решения 131
7.2.Классификация задач принятия решений 133
7.3. Многокритериальная оптимизация 135
7.4. Определение множества Парето 138
7.5. Методы условной многокритериальной оптимизации 140
8. Игровые модели принятия решений 144
8.1. Основные понятия теории игр 144
8.2. Платежная матрица антагонистической игры, принцип минимакса 145
8.3. Решение игр в смешанных стратегиях 147

8.4. Упрощение игр и аналитическое решение игр 2×2 148
8.5. Геометрическое решение игр 150
8.6. Решение игр со многими стратегиями на основе метода линейного программирования 153
8.7. Биматричные игры 155
8.8.Кооперативные игры 156
Элементы теории статистических оптимальных решений 159
9.1. Принятие решений при известных априорных вероятностях 160
9.2. Методы принятия решений в условиях априорной неопределенности 162
9.3. Планирование эксперимента при принятии решений 163
9.4. Многоэтапное принятие решений 165
10. Экспертные процедуры для принятия решений 169
10.1. Общая схема экспертизы 169
10.2. Задача оценивания 170
10.3. Подготовка экспертизы 170
10.4. Методы обработки экспертной информации 171
10.5. Метод Делфи для численной оценки 173
10.6. Строгое ранжирование 174
10.7. Нестрогое ранжирование 175
10.8. Метод попарных сравнений 176
Список литературы 178

ВВЕДЕНИЕ

Одним из главных направлений деятельности специалиста в любой сфере является совершенствование существующих и создание новых изделий, систем или технологий: создание лучших машин, экономное расходование ресурсов, сокращение сроков строительства и т.д.

Обычно та или иная цель может быть достигнута разными путями, но всегда важно знать наилучший из них, так как в реальных условиях приходится считаться с ограниченностью материальных ресурсов и времени, расходуемых на достижение цели. Понятие «лучший» начинает что-либо означать тогда, когда назван количественный показатель или критерий качества принимаемого решения. Например, изделие А лучше изделия В с точки зрения затрат материала; система С лучше системы D по показателю надежности и т.п. Вот почему получение наилучших вариантов возможно только при количественном описании предметной области, т.е. на основе математической модели.

Методология анализа сложных систем, их математическое моделирование и нахождение на этой основе наилучших (оптимальных) решений в общем виде изучается в науке «исследование операций». В рамках этого общего направления изучаются и математические «методы оптимизации». Последнее название применяют для методов нахождения экстремумов функций и функционалов, когда математическая модель задачи уже сформулирована.

Большой вклад в развитие методологии исследования операций и методов оптимизации внесли российские и зарубежные ученые: Л.В.Канторович, Л.С.Понтрягин, Н.Н.Моисеев, Дж. Данциг, Г.Кун и А.Таккер, Р.Беллман, Р.Гомори и многие другие.

При подготовке рукописи этой книги использован опыт преподавания на факультете информатики Самарского государственного аэрокосмического университета. Это предопределило перечень вопросов, которые в первую очередь должны были быть освещены. При изложении автор предельно кратко старался излагать сугубо теоретические вопросы, оставляя место основным идеям и методам, имеющим универсальное значение. Особый акцент сделан на практические алгоритмы решения разнообразных задач, которые можно положить в основу разрабатываемых компьютерных программ.

1. МЕТОДОЛОГИЯ СИСТЕМНОГО АНАЛИЗА И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

1.1. Системный анализ, система, оптимизация

В середине двадцатого века во многих областях деятельности человека сформировалась необходимость изучения и совершенствования сложных систем, в том числе систем организационного типа: систем обороны; производственных систем, отраслей, предприятий и др.

Оказалось:

1) в сложных системах взаимосвязь между элементами играет бóльшую роль, чем свойства самих элементов;

2) в таких системах элементы могут быть разнородны (оборудование, персонал, материалы, транспорт, условия поставок и сбыта и т.д.).

Термины системный анализ и исследование операций возникли, когда были образованы группы по исследованию военных операций в армии США и Англии во время второй мировой войны.

К этому времени, во-первых, был накоплен опыт применения математических методов для моделирования и решения некоторых задач экономики (В.Леонтьев, Л.В.Канторович), была теоретически обоснована возможность решения задач большой размерности на ЭВМ. Позже были созданы первые образцы таких машин. Совпали необходимость и возможность. Возникли новые идеи совершенствования организационных систем на основе математической теории игр (Нейман – Моргенштерн). Методология, сформулированная тогда и вобравшая в себя все научные достижения в области изучения сложных систем, оказалась применимой не только к боевым операциям, но и к другим сферам. Но термин «исследование операций» остался. Системный (операционный) подход – основа для изучения и совершенствования сложных систем. Применительно к специалистам - инженерам в области применения математики и информационных технологий в промышленности исследование операций формирует определенную технологию совершенствования существующих и создания новых систем как технических, так и организационных.

Система – множество элементов с определенными способами взаимодействия между ними, которые все вместе выполняют цель системы.

Процесс – все, что происходит в системе. Система работает, значит в ней происходит процесс.

Операция – часть процесса, которая наделена свойствами всей системы. Операция – это управляемое мероприятие, выполняющее определенную цель, сопоставимую с целью всей системы. Например, операция составле-

ния расписания учебных занятий для учебного процесса в системе «университет».

При исследовании сложных организационных систем:

1) невозможен экспериментальный метод исследования;
2) невозможно описание поведения систем только на основе какой-либо естественнонаучной теории;
3) при описании таких систем количество факторов, которые необходимо учитывать, велико.

Поэтому очевидно, что такие системы невозможно моделировать, изучать и совершенствовать без использования компьютерных средств и технологий.

1.2. Схема операционного проекта

Весь комплекс работ по изучению и совершенствованию системы (операции) проводит операционная группа системных аналитиков. Этот проект проводится в интересах лица, принимающего решения (ЛПР). ЛПР может отвергнуть проект, а может принять. На рис.1 изображена примерная схема этапов операционного проекта.

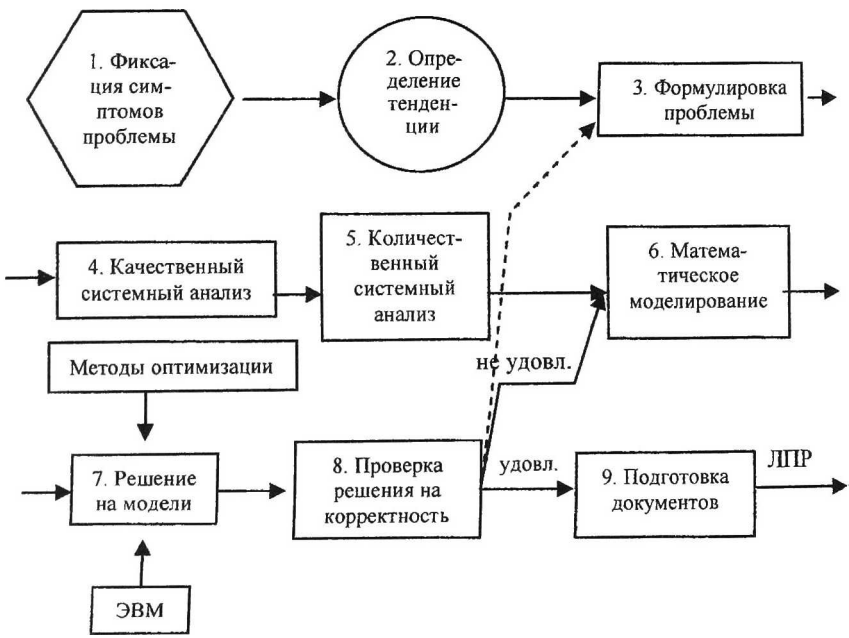


Рис. 1

Дадим краткую характеристику этапов операционного исследования.

1. В самом общем случае поводом для изучения и совершенствования системы служат зафиксированные симптомы, обнаруживающие проблемные вопросы в работе системы.

2.Установленные симптомы проблемы могут образовывать связанную цепочку фактов (тенденцию), которая помогает сформулировать проблему.

3.Важнейшим этапом исследования системы является четкая формулировка проблемы, которая присутствует на данном уровне жизнедеятельности системы.

4.Качественный системный анализ – это расщепление целостной системы (операции) на отдельные элементы (сущности). Для этого нужно:

- выделить изучаемую систему (операцию) из вышестоящей системы (операции);
- сформулировать цель, выполняемую системой (операцией);
- перечислить факторы, которые влияют на достижение цели;
- определить возможные ограничения, в рамках которых можно совершенствовать систему (операцию).

5.Количественный системный анализ предполагает описать все перечисленные факторы, которые участвуют в операции на количественном уровне, т.е. на основе измеримых параметров. Для этого:

- устанавливается критерий K – величина, количественно измеряющая степень достижения цели системы (операции);
- вводятся количественные внутренние параметры системы, которые измеряют факторы, участвующие в описании системы (операции);
- все множество этих параметров необходимо разбить на две части:
  а) неуправляемые параметры (константы), которые мы в данной конкретной системе (операции) менять не можем (производительность, нормы расхода материалов и т.п.), их обозначают как коэффициенты $A = (a_1, a_2, \ldots, a_k)$;
  б) управляемые параметры (переменные) – величины, которые мы можем менять $X = (x_1, x_2, \ldots, x_n)$.

6. Суть математического моделирования – установление количественных связей между введенными величинами K, A, и X в виде так называемой операционной модели.

Первая часть операционной модели - это модель целевой функции, она устанавливает функциональную зависимость критерия K от неуправляемых параметров A и управляемых величин X в виде $K = f(X, A)$. f – может быть функция, заданная аналитически, таблично или алгоритмом. В ряде практических задач в качестве элементов множества X выступают функции. В этом случае f - функционал. Задачи такого рода называются вариационными и в данном пособии не рассматриваются. Для целевой функции указывается направление улучшения критерия

$$K = f(X, A) \rightarrow \min(\max). \tag{1.1}$$

Этим выражением и определяется смысл оптимизации системы (операции).

Вторая часть операционной модели – математическое описание ограничений на выбор переменных X. Все ограничения в общем виде можно записать в виде неравенств (равенств):

$$\varphi_i(X, A) \leq 0, \; i = \overline{1,m}. \tag{1.2}$$

Каждая функция $\varphi_i(X, A)$ называется функцией ограничения. В некоторых задачах имеются требования на сам вид переменных X или K.

$$\left.\begin{array}{l} X \in D \\ K \in M \end{array}\right\}. \tag{1.3}$$

Например, часто возникает требование, чтобы X или K были целыми. В некоторых случаях они должны принадлежать некоторому стандартному множеству значений.

Модель в виде 1.1, 1.2, 1.3 – модель операционного вида или оптимизационная модель (неоптимизационная – без целевой функции).

Модель 1.1, 1.2, 1.3 позволяет поставить задачу оптимизации системы (операции) как математическую задачу: найти такие управляемые переменные X, которые удовлетворяли бы системе ограничений 1.2, 1.3 и обеспечивали бы наилучшее значение критерия K.

7.Решение поставленной математической задачи требует привлечения методов оптимизации, включающих кроме классических математических методов также и специальные методы исследования операций. Реальные задачи приводят к большой размерности (до десятков тысяч). Поэтому современные методы нахождения оптимальных решений ориентированы на использование компьютерных средств.

8.Сопоставляя полученное решение с содержательной постановкой задачи можно обнаружить противоречия или какие-нибудь некорректные элементы решения. Причиной некорректности могут быть ошибки в математической модели или неучет некоторых существенных ограничений. На этом этапе может принимать участие лицо, принимающее решение (ЛПР). Если полученное решение приемлемо – оно принимается, если нет - необходимо вернуться на этап математического моделирования или даже на более ранние этапы исследования.

9.Найденное оптимальное решение X^* позволяет подготовить управляющее решение в форме документа для ЛПР.

Таким образом, операционное исследование – итерационный процесс, который сходится к определенному оптимальному решению. Рассмотренная схема является примерной. Она позволяет лучше понять смысл системного

анализа и исследования операций как науки. Исследование операций (ИСО) - это наука о количественном обосновании оптимальных решений на основе построения и использования математической модели. К сожалению, процесс перевода на количественное описание сложных систем и операций не является технологией. Математическая модель может получиться удачной или неудачной для получения практических решений. Вот почему знаменитый операционист Томас Саати иронично определил науку «Исследование операций» как « искусство давать плохие советы в тех практических случаях, в которых другие науки ничего не могут посоветовать».

1.3. Особенности математического моделирования операций

Математическое моделирование – самый сложный этап ИСО. Математическая модель – это такое описание системы, которое позволяет специалисту выполнить цель исследования и оптимизации системы. Для одной и той же системы можно построить различные модели, чтобы выявить различные свойства: например аэродинамическая модель, прочностная модель и т.д. Модель должна быть адекватной реальной задаче, т.е. решения, полученные в рамках построенной модели, должны обладать такими же свойствами, что и реальная система.

К сожалению, не существует какого-либо алгоритма, по которому необходимо создать модель. Можно руководствоваться лишь принципами моделирования:

1. Необходимо решить вопрос размерности модели. Если число параметров увеличить, то мы более точно отобразим реальное событие, но в модели будет трудно выявить основные свойства (наиболее значимые). Задача становится необозримой и может не иметь решения. Поэтому число переменных по возможности стараются уменьшать, оставляя главные, существенные. Но уменьшая число переменных, мы можем опустить существенные, и модель становится неадекватной.

2. Модель зависит от точности, с которой нужно получить решение.

3. Модель зависит от того, насколько мы знаем исходные данные.

4. Подход к построению модели может быть двояким:

4.1.Создание оригинальной модели (не учитывая предыдущей модели в этой области), т.е. разработка «с чистого листа». Это может дать, а может не дать хороший результат. Преимущество: «свежий взгляд на вещи», нет чужих ошибок. Но это может быть дороже и потребует большого времени.

4.2.Использование типовых моделей для моделирования конкретных операций. В настоящее время существует большое число типовых моделей, описывающих наиболее распространенные виды операций и систем:

- модель линейного программирования (ЛП);
- модель динамического программирования;
- игровые модели;

- модель массового обслуживания;
- модель систем управления запасами;

и многие другие.

Такой подход наиболее подходит для обучения специалистов, так как дает им инструмент для математического моделирования реальных систем в их предметной области.

1.4. Постановка задачи исследования операций в детерминированном случае и в условиях неопределенности

Если в операционной модели

$$K = f(X, A) \to \min(\max) \tag{1.4}$$

$$\varphi_i(X, A) \le 0,\ i = \overline{1,m}, \tag{1.5}$$

все неуправляемые переменные A – заранее точно известные величины (const), а f – детерминированная функция, то такая модель называется детерминированной.

В некоторых задачах это не выполняется. Среди неуправляемых переменных есть такие $Z=(z_1, z_2, ..., z_l)$, которые меняются случайно. В этом случае говорят, что операция проходит в условиях неопределенности. $K=f(X, A, Z)$. K меняется не только с изменением X, и задача на максимум K становится некорректной.

В этом случае ставится задача выбора такой критериальной величины, которая менялась бы только от X.

В простейшем случае, когда f – линейная функция, можно искусственно свести задачу к детерминированной, заменив Z на $M[Z]$, т.к. в этом случае

$$M[K] = f(X, A, M[Z]).$$

Это же можно сделать, если f слабо меняется от Z. Но в общем случае это не так. В общем случае усреднение проводят по всем правилам: $M[K] = \underbrace{\int\int..}_{l} \int_{штук} \theta(z_1..z_l) \cdot f(X, A, Z) dz_1..dz_l \to max$, где функция θ - плотность распределения вероятностей величин Z.

Вид целевой функции после такого преобразования существенно усложняется. Поэтому усложняется задача отыскания решения.

1.5. Пример математического моделирования операции (задача о краске)

Для демонстрации существа операционного подхода рассмотрим простой пример.

Содержательное описание задачи

Для окраски помещения необходимо купить 15 кг краски. Эту краску можно купить в банках двух типов: по 1,5 кг, стоимостью 10 рублей каждая, или банках весом 0,9 кг, стоимостью 8,5 рублей. Для перевозки используется ящик, в который может уместиться 8 банок первого типа или 25 банок второго типа.

Необходимо дать математическую формулировку задачи минимизации стоимости покупки. Найти, сколько целых банок каждого типа надо купить.

Необходимо дать графическую интерпретацию решения. Сравнить с решением, которое получится, если банка краски первого типа будет стоить 17 рублей.

Решение

Обозначим x_1 - количество банок первого типа, x_2 - количество банок второго типа. Так как количество купленной краски не должно быть меньше 15 кг, то $1,5x_1 + 0,9x_2 \geq 15$.

Каждая банка первого типа занимает 1/8 объема ящика, а второго 1/25 часть, поэтому ограничение на объем ящика можно выразить следующим образом: $\frac{1}{8}x_1 + \frac{1}{25}x_2 \leq 1$.

Стоимость покупки обозначим L, она равна $L = 10x_1 + 8,5x_2 \to \min$.

Получаем задачу, принадлежащую к классу задач линейного программирования (ЛП):

$$\begin{cases} L = 10x_1 + 8,5x_2 \to \min; & (1.6) \\ 1,5x_1 + 0,9x_2 \geq 15; & (1.7) \\ 1/8\, x_1 + 1/25\, x_2 \leq 1. & (1.8) \end{cases}$$

Если банки вскрывать нельзя, то необходимо добавить ограничения:

$$x_1 \text{ и } x_2 \text{ - целые.} \qquad (1.9)$$

Задача 1.6, 1.7, 1.8, 1.9 – это задача линейного целочисленного программирования (ЛЦП).

Если количество переменных в задачах ЛП и ЛЦП большое, то их необходимо решать с использованием специальных алгоритмов (которые мы изу-

чим) и компьютерных программ. В нашем случае – две переменные, следовательно, можно использовать простое геометрическое решение.

Рассмотрим равенства (1.7) и (1.8).

$1{,}5x_1 + 0{,}9x_2 = 15$ - уравнение прямой линии.

Делим на 15 правую и левую части уравнения: $x_1/10 + x_2/16{,}6 = 1$.

Это каноническое уравнение прямой. Построим его (рис.2).

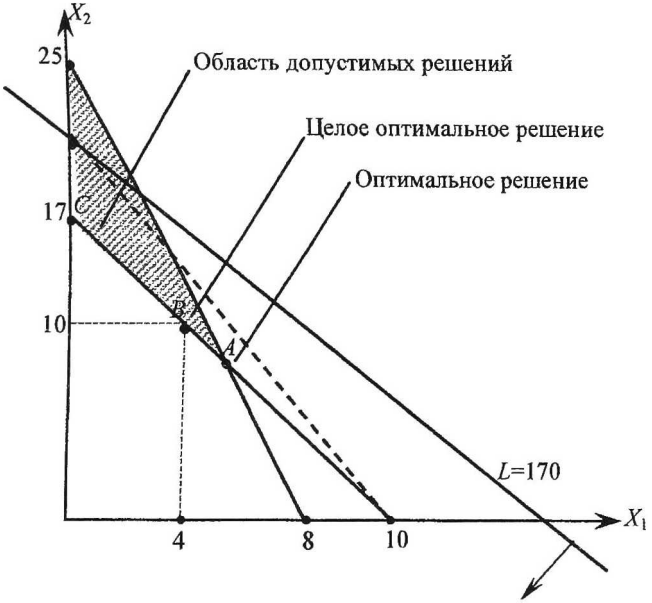


Рис. 2

Уравнение (1.8) имеет вид $\frac{1}{8}x_1 + \frac{1}{25}x_2 = 1$.

Проверяя допустимость какой-нибудь точки для неравенств (1.7) и (1.8), например, точки (0; 0), определяем область допустимых решений, удовлетворяющих (1.7) и (1.8).

Изобразим линию постоянного уровня целевой функции L=const. возьмем, например, L=170, получим уравнение прямой $170 = 10x_1 + 8{,}5x_2$.

Приведем его к канонической форме и построим $x_1/17 + x_2/20 = 1$.

Убеждаемся, что если $L \rightarrow \min$, то эта прямая движется параллельно самой себе к началу координат. Оптимальной точкой будет точка выхода этой прямой из области допустимых решений.

Очевидно, что точка A является оптимальной для случая, когда банки можно вскрывать (x_1 и x_2 могут быть нецелыми), а точка B – оптимальная точка для целого решения.

Найдем координаты точки A, решив совместно уравнения (1.7) и (1.8):

$$\begin{cases} 1,5x_1 + 0,9x_2 = 15; \\ 1/8x_1 + 1/25x_2 = 1. \end{cases}$$

Получим $x_1^* = 5,8$; $x_2^* = 7,1$; $L^* = 118,35$.

Заметим, что «округленное» решение x_1=6; x_2=7 не является допустимым.

Оптимальное решение, которое является последним при выходе прямой L, находим из целых допустимых. Это точка B с координатами x_1=4; x_2=10; L=125.

Если стоимость банки краски первого типа повысится до 17 рублей, то целевая функция будет $L = 17x_1 + 8,5x_2 \to \min$.

Построим прямую для L=170. $x_1/10 + x_2/20 = 1$.

Очевидно, что оптимальная точка будет в точке С: x_1=0; x_2=16,6.

Целое оптимальное решение будет x_1=0; x_2=17; L'=144,5.

Ответ:

1) Если банки можно вскрывать, то необходимо взять
   - 5,8 банок первого типа;
   - 7,1 банок второго типа, стоимость L^*=118,35.
2) Если банки вскрывать нельзя, то необходимо взять
   - 4 банки первого типа;
   - 10 банок второго типа, стоимость L=125.
3) Если стоимость банки первого типа станет 17 рублей, то оптимальное решение изменится: x_1=0; x_2=17; L'=144,5.

2. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ (ЛП)

ЛП - наиболее распространенный и хорошо изученный раздел исследования операций. Изучение ЛП можно проводить в двух направлениях:

1) как развитие линейной алгебры (чисто математическое);

2) изучение конкретных моделей, приводящих к ЛП.

Впервые задача ЛП была сформулирована Л.В.Канторовичем, который применил математическую модель задачи ЛП для экономики (1939 г.). Впоследствии, в 1947 г. американец Джон Данциг разработал алгоритм решения этой задачи. С этого момента ЛП стало основным методом в системном анализе и в первую очередь в задачах экономики. В 1975г. Л.В. Канторович получил Нобелевскую премию как основоположник ЛП.

Задача ЛП – такая задача исследования операций, когда целевая функция и все функции ограничений линейны, а все переменные – действительные числа:

$$L = \sum_{j=1}^{n} c_j x_j \to \min(\max),$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_{ij}, i = \overline{1,m},$$

$$x_j \in R, \forall j = \overline{1,n}.$$

Пример: Задача о диете.

Имеются четыре вида продуктов $П_1$, $П_2$, $П_3$, $П_4$. Известна стоимость каждого продукта c_1, c_2, c_3, c_4. В рационе должно быть белков не менее b_1; жиров не менее b_2; углеводов не менее b_3 . Известно удельное содержание каждого вещества в единице продукта. Необходимо рассчитать рацион, содержащий необходимое количество полезных веществ и при этом, чтобы стоимость всего рациона была минимальной. Построим математическую модель:

j – номер продукта ; i – номер вещества;

a_{ij} – удельное количество в j – м продукте i – го вещества

(например, a_{23} =30г/кг).

Введем величину x_j – количество j-го продукта в рационе (управляемая переменная), величины c_j, a_{ij}, b_i неуправляемые параметры – константы. Выберем критерий L – общая стоимость рациона.

Тогда целевая функция и ограничения имеют вид

$$L=c_1x_1+c_2x_2+c_3x_3+c_4x_4 \to \min, \qquad (2.1)$$

$$a_{11}x_1+a_{12}x_2+a_{13}x_3+a_{14}x_4 \geq b_1,$$

$$a_{21}x_1+a_{22}x_2+a_{23}x_3+a_{24}x_4 \geq b_2, \qquad (2.2)$$

$$a_{31}x_1+a_{32}x_2+a_{33}x_3+a_{34}x_4 \geq b_3.$$

Математически эту задачу можно сформулировать так: найти такие значения x_1, x_2, x_3, x_4, чтобы они удовлетворяли системе линейных неравенств и при этом линейная функция L была бы минимальной. В компактном виде это можно записать:

$$L = \sum_{j=1}^{4} c_j x_j \to \min, \qquad (2.3)$$

$$\sum_{J} a_{ij} x_j \geq b_j, i = \overline{1,m} (i = \overline{1,3}), \qquad (2.4)$$

$$x_j \geq 0. \qquad (2.5)$$

Термин ЛП буквально означает линейное планирование.

2.1. Общая и основная задачи ЛП

Моделируя реальную задачу, мы можем получить задачу ЛП с некоторыми особенностями:

1) в одном случае $L \to \min$, в другом- $L \to \max$;

2) в ограничениях могут быть знаки: $\geq$, $\leq$, $=$, $>$, $<$;

3) переменные могут быть любыми (как положительными, так и отрицательными).

Рассмотрим стандартную (основную) форму задачи ЛП, будем сокращенно называть ее ОЗЛП. В ней:

1) целевая функция минимизируется: $L \to \min$;

2) ограничения только равенства;

3) все переменные неотрицательны: $x_j \geq 0, \forall_j$.

Любую задачу ЛП в общем виде можно привести к ОЗЛП, для чего необходимо выполнить следующие действия:

1. Если $L \to \max$, необходимо ввести $L^1 = -L$, тогда если $L \to \max$, то $\to \min$, и при этом $\arg\min L^1 = \arg\max L$.

2. Чтобы привести неравенство к равенству, необходимо
   - перенести все части неравенства в одну сторону (ту, которая ≥ 0);
   - полученную функцию обозначить новой дополнительной переменной, которая ≥ 0;

Это обозначение является равенством, эквивалентным неравенству.

Пример:

$2x_1 - x_2 + x_3 \geq 4,$

$x_1 \leq x_2 + 3x_3.$

Введем дополнительные переменные $y_1 \geq 0, y_2 \geq 0$:

$y_1 = 2x_1 - x_2 + x_3 - 4, \quad y_1 \geq 0,$

$y_2 = x_2 + 3x_3 - x_1, \qquad y_2 \geq 0.$

Ясно, что за переход к равенству приходится платить увеличением числа переменных.

3. Пусть x_k – переменная с любым знаком. Представим ее разностью двух положительных переменных. Тогда, если вместо x_k подставить разность $x_k = x_к^{/} - x_к^{//}$, то остаются только неотрицательные переменные:

Пример:

$L = x_1 - 3x_2 + x_3 \to \max,$

$x_1 + 2x_2 - x_3 \leq 3,$

$x_1 \geq x_2,$

$x_1 \geq 0, \; x_2 \geq 0, x_3 - \textit{любое}.$

1. $L' = -L = -x_1 + 3x_2 - x_3 \to \min,$
2. $y_1 = 3 - x_1 - 2x_2 + x_3,$ $y_2 = x_1 - x_2,$
3. $x_3 = x_4 - x_5.$

Получили ОЗЛП: $L' = -x_1 + 3x_2 - x_4 + x_5 \to \min,$

$y_1 = 3 - x_1 - 2x_2 + x_4 - x_5,$

$y_2 = x_1 - x_2,$

$(x_1, x_2, x_4, x_5, y_1, y_2) \geq 0.$

Таким образом, ОЗЛП в общем виде можно записать так:

$$L = \sum c_j x_j \to \min, \tag{2.6}$$

$$\sum a_{ij} x_j = b_i, i = \overline{1,m}, \tag{2.7}$$

$$x_j \geq 0, \forall j = \overline{1,n}. \tag{2.8}$$

В матричном виде: CX →min , AX=B, X>=0.

Известно, что система линейных уравнений (СЛУ) (2.7) совместна, когда ранг матрицы A равен рангу AB; A = $\{a_{ij}\}$, B = $\{b_i\}$. Но допустимыми решениями задачи ЛП являются не все, а только ≥ 0 решения, т.к. должно выполняться (2.8). Если $m = n$, то может быть только одно решение; если $m < n$, то может быть бесчисленное множество решений, но таких, которые удовлетворяют системе (2.7). Эти точки образуют так называемую область допустимых решений (ОДР).

2.2. Геометрическая интерпретация задачи ЛП

Рассмотрим частный случай n-m=2. Тогда m переменных выбираются базисными, а $(n - m)$ – свободными. Рассмотрим такую систему, где

x_1, x_2 – *свободные*,

$x_3, x_4, ..., x_n$ – *базисные*,

$$\left.\begin{aligned} x_3 &= \alpha_{31}x_1 + \alpha_{32}x_2 + \beta_3 \\ x_4 &= \alpha_{41}x_1 + \alpha_{42}x_2 + \beta_4 \\ &\dots \\ x_n &= \alpha_{n1}x_1 + \alpha_{n2}x_2 + \beta_n \end{aligned}\right\}, \qquad (2.9)$$

$$L = \gamma_1 x_1 + \gamma_2 x_2 + \gamma_0 \to \min.$$

Рассмотрим (2.9).

Пусть x_3=0,

$\alpha_{31}x_1 + \alpha_{32}x_2 + \beta_3 = 0$ – уравнение прямой. Значит для x_3>0 точки лежат по одну сторону от этой прямой. Покажем это штриховкой. Аналогично для других равенств (2.9). Очевидно, точки, где все штриховки пересекаются, и есть ОДР (рис. 3).

Область допустимых решений (ОДР) содержит все точки, удовлетворяющие задаче.

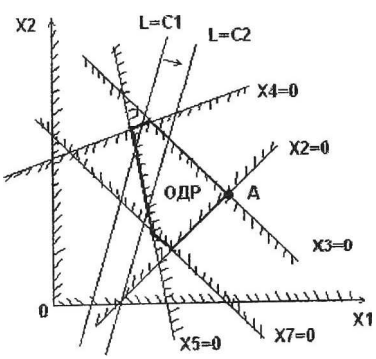


Рис 3

Рассмотрим теперь целевую функцию L:

Пусть $L = C_1 : \gamma_1 x_1 + \gamma_2 x_2 = C_1 - \gamma_0$ - уравнение прямой.

Точки на прямой $L = C_2$ лучше, чем $L = C_1$ (в области), т.к. они дают меньшее значение целевой функции.

Оптимальное решение получается в точке выхода линии уровня целевой функции L (точка A) из ОДР. Отсюда ясно, что в задаче ЛП внутри ОДР не может быть оптимальной точки.

Выводы из геометрической интерпретации ОЗЛП.

1) ОЗЛП не обязательно имеет решение (рис. 4).

Нет допустимого решения.

Рис. 4

2) ОДР всегда выпуклый многоугольник (рис. 5):

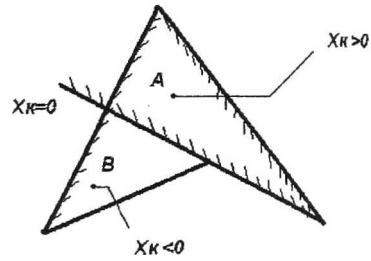


Рис. 5

3) Если есть допустимое решение, то не обязательно есть оптимальное решение. ОДР может быть открытой, в этом случае оптимальной точки может не быть, если целевая функция изменяется, как показано в I (движемся в открытую область). Но если область открыта, то это не значит, что нет оптимального решения. Для целевой функции II оптимальное решение находится в точке A (рис. 6).

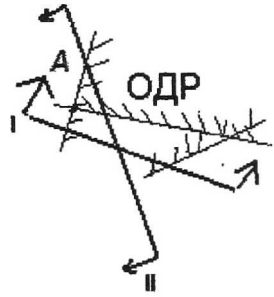


Рис. 6

4) В задаче может быть бесчисленное множество оптимальных решений. Оптимальное решение будет в любой точке AB (рис. 7).

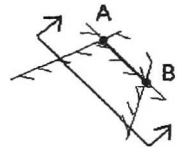


Рис. 7

5) Если оптимальное решение есть, то его необходимо искать в вершине многоугольника. Признаком вершины является пересечение двух прямых, т.е. равенство нулю двух переменных.

6) Если n-m=2, т.е. в ОЗЛП две свободные переменные. Тогда, чтобы попасть в вершину, можно приравнять к нулю две свободные переменные.

7) Если взять две переменные равными нулю (рис. 8), то мы можем попасть в недопустимую вершину (точка 1).

Точка 1: ((x3=0) и (x5=0)). Чтобы перейти в соседнюю вершину (точка 2), нужно вместо x5 =0 взять x4=0, а x5 не равно нулю. Получаем точку 2 ((x3=0) и x4=0)). Допустимую вершину назовем опорной.

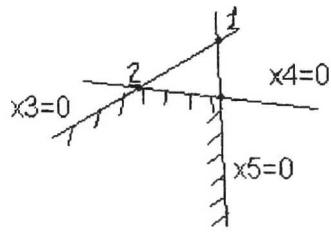


Рис. 8

Рассмотрим случай, когда $n - m = 3$.

$x_4 = \alpha_{41}x_1 + \alpha_{42}x_2 + \alpha_{43}x_3 + \beta_4$,

...

$x_4 = 0$ – *плоскость*,

$x_4 \geq 0$ – *полупространство.*

Тогда ОДР – выпуклый многогранник.

$L = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 x_3$ –> min - целевая функция (при L= const – плоскость).

Допустимые решения находятся во внутренних точках многогранника, а оптимальное, очевидно, может находиться только в вершине многогранника в точке «последнего» касания плоскости – целевой функции и многогранника, где пересекаются 3 плоскости, т.е. 3 переменных равны нулю.

Если в общем случае $n - m = k$, то у задачи k свободных переменных. Следовательно, нужно искать оптимальное решение там, где k переменных равны нулю. Т.к. в вершине в особом случае может пересекаться и большее число плоскостей, то в оптимальной точке $\geq k$ переменных должны быть равны нулю. Заметим, что это является необходимым условием оптимального решения для задачи ЛП.

2.3. Идея симплекс-метода решения задачи ЛП

Рассмотрим ОЗЛП, где базисные переменные $x_{k+1}, \ldots, x_n$ разрешены относительно свободных $x_1, x_2, \ldots, x_k$. Например:

$$L = 2x_1 + 3x_2 + ... + 5x_k + 6 \to \min,$$
$$x_{k+1} = 2x_1 + 3x_2 - x_3 + ... + 6x_k + 4,$$
$$x_{k+2} = x_1 - x_2 + x_3 + ... + 7x_k - 7,$$
.....
$$x_n = 2x_1 + 2x_2 - 3x_3 + ... + x_k + 2.$$

(x_1=x_2=...x_k=0) – вершина, при этом базисные переменные x_{k+1}=4, x_{k+2}= - 7, x_n= 2.

В ОЗЛП все переменные должны быть ≥0, следовательно, эта вершина недопустима. Признаком недопустимости вершины является наличие отрицательных свободных членов в ограничениях равенствах. Если отрицательный свободный член есть, то надо перейти в другую вершину, для чего одну базисную переменную сделать свободной и приравнять к нулю, а какую-то свободную превратить в базисную. Следовательно, переход из одной вершины в другую осуществляется путем замены базисной переменной на свободную. Это называется симплекс-преобразованием. Если мы будем переходить из вершины в вершину «в сторону ОДР», то за конечное число таких переходов попадем в допустимую вершину. Назовем ее опорной вершиной, а соответствующее решение опорным решением.

Пусть получено опорное решение, например, для следующей задачи. Проверим, оптимально оно или нет.

$$x_3 = 2 + x_1 - x_2,$$
$$x_4 = 4 - x_1 + x_2,$$
$$L = 3 + 2x_1 - x_2 \to \min,$$
$$x_1 = x_2 = 0; x_3 = 2; x_4 = 4; L = 3.$$

Видно, что если все коэффициенты при свободных переменных в целевой функции положительны, то никакие увеличения свободных переменных от 0 вверх не уменьшают L, следовательно, это L является оптимальным значением. Если же есть отрицательные коэффициенты при свободных переменных (в нашем примере при x_2), то ее надо сделать базисной, т.к. если она будет положительной, то L будет меньше. Но x_2 может увеличиваться не бесконечно, так как в ограничениях есть отрицательный коэффициент при x_2. Видно, что увеличивать x_2 можно только до x_2=2, так как в противном случае x_3 будет отрицательным, а это невозможно. Так мы нашли базисную переменную x_3 , которую надо поменять со свободной x_2. Поиск этой пары для симплекс-преобразования и является основным элементом симплекс-метода.

Таким образом, симплекс-метод – это метод решения задач ЛП, который основывается на процедуре перехода от одной вершины к другой до тех пор, пока не придем в оптимальную вершину. Он состоит из трех алгоритмов:

1) Алгоритм перехода из одной вершины в другую, когда известна пара преобразования. Он пересчитывает коэффициенты системы уравнений для нового базиса. Назовем его *алгоритмом симплекс-преобразования*.
2) Алгоритм нахождения такой пары преобразования, чтобы при переходе в новую вершину мы приближались к ОДР. Это *алгоритм отыскания опорного решения*.
3) Алгоритм нахождения такой пары преобразования, чтобы новая опорная вершина давала лучшее значение целевой функции. Это *алгоритм отыскания оптимального решения*.

В симплекс-методе начинают со стандартной вершины (все свободные переменные равны нулю), т.е. с точки 1 (это недопустимая вершина, смотри рисунок 9). Дальше последовательно переходим из одной вершины в другую (2,3) и попадаем в опорную вершину 4.

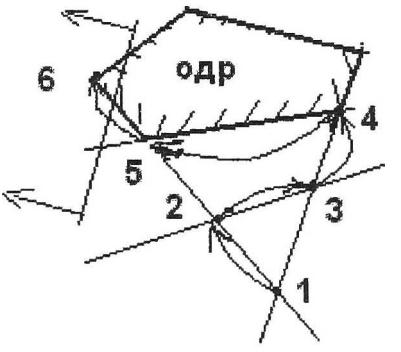


Рис. 9

Найдя опорное решение, мы движемся по вершинам ОДР так, чтобы целевая функция улучшалась, и при этом эти вершины были опорными (5 и 6 – оптимальная).

Эти три алгоритма гарантируют либо отыскание оптимального решения, либо доказательство отсутствия допустимого решения, либо доказательство отсутствия оптимального решения.

2.4. Симплекс-таблица, стандартный алгоритм симплекс-преобразования

Существует много форм симплекс–таблиц (СТ). Мы рассмотрим наиболее компактную форму, в которой строки – базисные переменные, а столбцы свободные переменные. Каждая СТ содержит коэффициенты модели задачи ЛП . Для записи СТ необходимо представить ОЗЛП в стандартной форме:

$$L = \gamma_0 - (\gamma_1 x_1 + \gamma_2 x_2 + \ldots + \gamma_k x_k),$$
$$y_1 = b_1 - (\alpha_{11} x_1 + \alpha_{12} x_2 + \ldots + a_{1k} x_k),$$
$$y_2 = b_2 - (\alpha_{21} x_1 + \alpha_{22} x_2 + \ldots + a_{2k} x_k),$$
$$\ldots$$
$$y_n = b_n - (\alpha_{n1} x_1 + \alpha_{n2} x_2 + \ldots + a_{nk} x_k),$$

т.е. на первом месте свободный член, а коэффициенты при свободных переменных меняют знак.

Пример:

$$L = 2x_1 - x_3 + 4,$$
$$y_1 = 2x_1 + x_2 + x_3,$$
$$y_2 = 3x_1 + x_2 - x_3 + 1,$$
$$y_3 = 2x_1 - x_2 - 2.$$

Стандартная форма:

$$L = 4 - (-2x_1 + 0x_2 + 1x_3),$$
$$y_1 = 0 - (-2x_1 - 1x_2 - 1x_3),$$
$$y_2 = 1 - (-3x_1 - 1x_2 + 1x_3),$$
$$y_3 = -2 - (-2x_1 + 1x_2 - 0x_3).$$

Записываем коэффициенты этой формы в левый верхний угол соответствующей клетки СТ. В нижней части будем писать промежуточные результаты вычислений (табл. 1).

Каждое решение, соответствующее СТ выглядит так: свободные = 0, базисные = свободным членам.

Таблица 1

	1	x_1	x_2	x_3
L	4 0	-2 0	0 0	1 0
y_1	0 -1	-2 3	-1 -1	-1 -1
y_2	1 -1	-3 3	-1 -1	1 -1
y_3	-2 1	-2 -3	1 1	0 1

$$x_1 = x_2 = x_3 = 0;$$
$$y_1 = 0; y_2 = 1; y_3 = -2;$$

L=4.

Рассмотрим порядок вычислений для перехода в новую вершину.

Пусть нам необходимо выполнить симплекс-преобразование $x_i \leftrightarrow x_j$. Тогда нужно получить новую симплекс-таблицу. Для этого выполним алгоритм.

Алгоритм 1:

1. Отыскиваем в СТ элемент α_{ij} и объявляем его генеральным. Тогда i-я строка и j-й столбец – генеральные.
2. Вычисляем величину, обратную генеральному элементу, и записываем ее в нижней части генеральной клетки:

$$\lambda = \frac{1}{a_{ij}}.$$

3. Все элементы генеральной строки умножаем на λ. Результат записываем в нижней части соответствующих клеток.
4. Все элементы генерального столбца умножаем на -λ. Результат записываем в нижней части клеток.
5. Отмечаем верхние элементы генеральной строки и нижние элементы генерального столбца.
6. В каждой клетке, не принадлежащей ни генеральному столбцу, ни генеральной строке, в нижнюю часть записываем произведение отмеченных элементов, стоящих в том же столбце и той же строке, что и данная клетка.
7. Переписываем СТ, заменяя в соответствии с табл. 2:

Таблица 2

	1	x_1	x_2	x_3
L	4	-2	0	1
y_1	-1	1	-1	-2
y_2	-1	3	-1	-1
y_3	-1	-5	1	1

- обозначение базисной переменной x_i обозначением свободной переменной x_j и наоборот;
- элементы генеральной строки и генерального столбца нижними элементами;
- все оставшиеся элементы суммой нижнего и верхнего числа.

В приведенных СТ показано преобразование $y_2 \leftrightarrow x_2$.

2.5. Алгоритм отыскания опорного решения задачи ЛП

Этот алгоритм выполняется сразу после записи симплекс-таблицы.

Алгоритм 2:

1. Просматриваем столбец свободных членов (не смотря на свободный член строки *L*). Если все свободные члены ≥ 0, то данная таблица уже соответствует опорному решению, иначе переходим к п. 2.
2. В столбце свободных членов выбираем первый по порядку отрицательный элемент и в этой строке находим еще один отрицательный элемент. Если такого элемента нет, то задача ЛП не имеет решения, если есть, то переходим к п. 3.
3. Столбец с найденным отрицательным элементом объявляем генеральным.
4. В генеральном столбце фиксируем элементы, имеющие такой же знак, что и знак соответствующих свободных членов, и среди них в качестве генерального элемента выбираем тот, отношение к которому соответствующего свободного члена минимально:

$$\min_i\left(\frac{b_i}{a_{ij}}\right),\ \text{знак } b_i = \text{знак } \alpha_{ij}.$$

5. По выбранному генеральному элементу выполняем стандартное симплекс- преобразование (алгоритм 1) и переходим к п. 1.

Если при вычислениях нет ошибок, то количество отрицательных свободных членов должно уменьшаться или должна уменьшаться их абсолютная величина.

Примечание к пункту 2:

Задача ЛП не имеет решения, когда соответствующая строка СТ имеет, например, вид: $y_4 = -1-(2x_1 + x_2 + 4x_3)$, которая не удовлетворяется ни при каких неотрицательных переменных.

2.6. Алгоритм отыскания оптимального решения задачи ЛП

Этот алгоритм выполняется после того, как найдено опорное решение, то есть когда в симплекс-таблице все свободные члены ≥ 0.

Алгоритм 3:

1) Просматриваем элементы строки L (несмотря на свободный член). Если все элементы ≤ 0, то данная симплекс-таблица уже соответствует оптимальному решению, иначе переходим к п. 2.
2) В строке L выбираем наибольший положительный элемент и соответствующий столбец объявляем генеральным.
3) В генеральном столбце находим положительные элементы. Если таких нет, то задача не имеет оптимального решения, иначе переходим к п. 4.
4) Среди положительных элементов генерального столбца выбирается тот, отношение к которому соответствующего свободного члена минимально:

$$\min_i\left(\frac{b_i}{a_{ij}}\right),\ b_i \geq 0,\ \alpha_{ij} > 0.$$

5) По выбранному генеральному элементу осуществляем симплекс-преобразование (алгоритм 1) и переходим к п. 1.

Примечание к пункту 2:

Если в генеральном столбце нет положительных элементов, то задача не имеет оптимального решения – целевая функция уменьшается неограниченно (движется в открытую область). Это значит, что в исходной модели не учтено какое-то важное ограничение.

Схема алгоритмов решения задач ЛП показана на рис. 10:

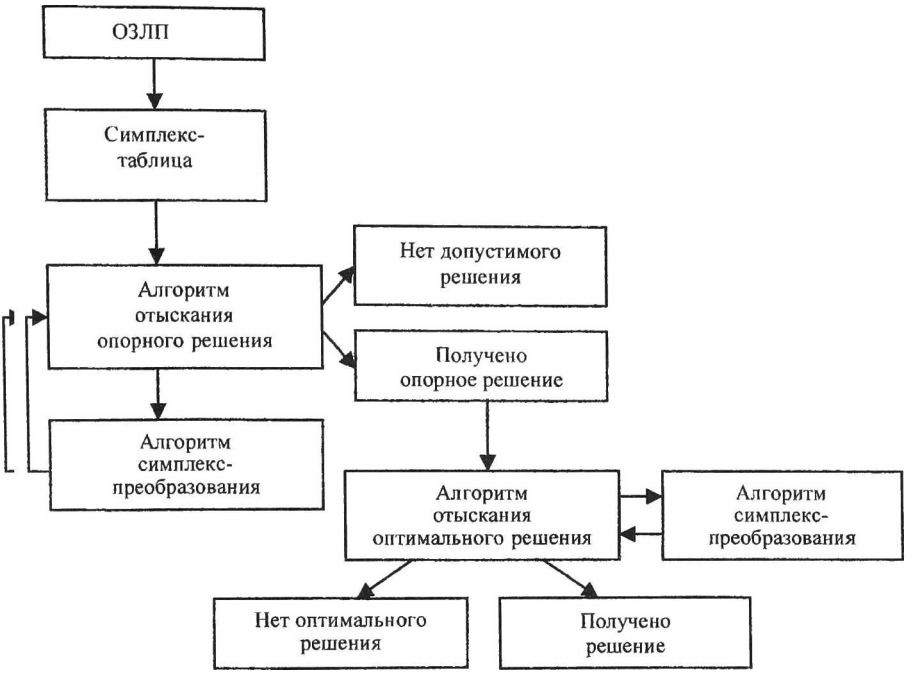


Рис. 10

Пример:

Решить задачу ЛП

$$L = x_1 + x_2 \rightarrow \max;$$
$$x_2 \le x_1 + 2;$$
$$x_2 \ge 1;$$
$$x_1 \le 4;$$
$$(x_1, x_2) \ge 0.$$

Решение:

Приведем задачу к ОЗЛП:

$L' = -x_1 - x_2 \rightarrow \min$; $L' = 0 - (x_1 + x_2)$;

$y_1 = x_1 - x_2 + 2$; $y_1 = 2 - (-x_1 + x_2)$;

$y_2 = x_2 - 1$; $y_2 = -1 - (0x_1 - x_2)$;

$y_3 = 4 - x_1$; $y_3 = 4 - (x_1 + 0x_2)$.

Таблица 3

	1	X_1	X_2
L'	0 -1	1 0	1 1
y_1	2 -1	-1 0	1 1
y_2	-1 1	0 0	-1 -1
y_3	4 0	1 0	0 0

Таблица 4

	1	X_1	y_2
L'	-1 -4	1 -1	1 0
y_1	1 4	-1 1	1 0
X_2	1 0	0 0	-1 0
y_3	4 4	1 1	0 0

2/1=2; -1/-1=1(min).

В таблице 4 получено опорное решение: $x_1=y_2=0$,

$y_1=1$, $x_2=1$, $y_3=4$, $L'=-1$.

В таблице 4 выполняем алгоритм отыскания оптимального решения:

Таблица 5

	1	y_3	y_2
L'	-5 -5	-1 -1	1 -1
y_1	5 5	1 1	1 1
X_2	1 5	0 1	-1 1
X_1	4 0	1 0	0 0

Таблица 6

	1	y_3	y_1
L'	-10	-2	-1
y_2	5	1	1
X_2	6	1	1
X_1	4	1	0

В таблице 6 получено оптимальное решение:

$y_3 = y_1 = 0$, $y_2 = 5$, $x_2 = 6$, $x_1 = 4$, $L' = -10$, $Lmax = 10$.

Так как в данной задаче всего две свободных переменных, то можно рассмотреть ее геометрически на плоскости (рис. 11).

Так как $y_1=0$, $y_3=0$, то решение находится в точке, в которой выполняются равенства: $x_1-x_2+2=0$ и $4-x_1=0$.

На рисунке 11 показана последовательность перехода от вершины к вершине в соответствии с работой алгоритма симплекс-метода. Цифрами обозначены соответствующие симплекс-таблицы.

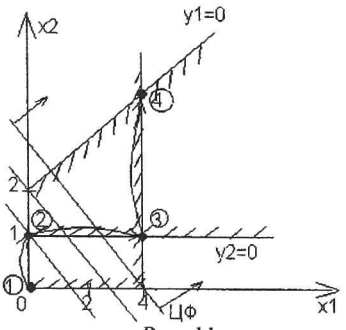


Рис. 11

2.7. Алгоритм получения первого базисного решения с использованием симплекс – процедуры (метод искусственного базиса)

Выше было показано, что алгоритмы симплекс-метода требуют, чтобы исходная система ограничений-равенств была приведена к базису, т.е. какие-нибудь базисные переменные должны быть выражены через свободные. Оказывается, что первый базис можно так же получить, используя алгоритм симплекс-метода. Для этого для каждого ограничения-равенства вводится искусственная переменная z_i (i=1,m). Каждое ограничение-равенство

$$\beta_i - (g_{i1} \cdot x_1 + \ldots + g_{in} \cdot x_n) = 0$$

формально записывается так:

$$z_i = \beta_i - (g_{i1} \cdot x_1 + \ldots + g_{in} \cdot x_n),\ i=1,m\,,$$

т.е. переменные z_i тождественно равны нулю. Последнее равенство определяет так называемый искусственный базис.

Вводим в рассмотрение дополнительную целевую функцию

$$L_o = z_1 + z_2 + \ldots + z_m \rightarrow \min\,,$$

для чего суммируем коэффициенты при переменных x_j всех равенств z_i и получаем ОЗЛП, готовую для решения симплекс-методом.

Решая эту задачу на основе вышеприведенных алгоритмов последовательно выводим из базиса все искусственные переменные z_i. При этом, как только переменная z_r становится свободной, то столбец z_r исключается из симплекс-таблицы.

Пример. Пусть исходную задачу ЛП мы привели к форме ОЗЛП и получили систему ограничений-равенств

$x_1 + 2x_2 = 4,$

$x_1 - x_2 + 2x_3 = 2.$

Найдем первый базис, т.е. разрешим базисные переменные (их всего будет 2) относительно свободных (n-m=3-2=1). Для этого запишем

$z_1 = 4 - (x_1 + 2x_2),$

$z_2 = 2 - (x_1 - x_2 + 2x_3),$ (2.10)

$Lo = z_1 + z_2 = 6 - (2x_1 + x_2 + 2x_3) \rightarrow \min.$

Запишем симплекс-таблицу.

Таблица 7

	1	X_1	X_2	X_3
L_0	6 -4	2 -2	1 2	2 -4
Z_1	4 -2	1 -1	2 1	0 -2
Z_2	2 2	(1) 1	-1 -1	2 2

В таблице 7 находим генеральный элемент по алгоритму нахождения оптимального решения, получаем пару для симплекс-преобразования и переходим к таблице 8, в которой исключаем столбец z_2.

Таблица 8

	1	X_2	X_3
L_0	2 -2	3 -1	-2 2
Z_1	2 2/3	(3) 1/3	-2 -2/3
X_1	2 2/3	-1 1/3	2 -2/3

В таблице 8 находим генеральный элемент и пару $z_1 \leftrightarrow x_2$, переходим к таблице 9.

Таблица 9

	1	X_3
L	0	0
X_2	2/3	-2/3
X_1	8/3	4/3

Таблица 9 соответствует оптимальному решению вспомогательной задачи (2.10) и дает выражение базисных переменных x_1, x_2 относительно свободной переменной x_3, а именно:

$x_2 = 2/3 + 2/3x_3;$

$x_1 = 8/3 - 4/3x_3.$

Теперь можно приступать к выполнению симплекс-алгоритма, записав вместо строки L_0 целевую функцию задачи, выраженную через свободные переменные. Чтобы такая запись получалась автоматически можно в таблицу 7 кроме строки L_0 записать строку целевой функции исходной ОЗЛП и выполнять описанный выше алгоритм, помня, что генеральный элемент не может быть ни в строке L_0, ни в строке ЦФ.

2.8. Вырожденная задача ЛП

При использовании симплекс-метода некоторые свободные члены могут быть равны нулю. Это значит, что в вершине, которой соответствует СТ, равны нулю не k переменных, а больше (равна нулю и базисная). В этом случае при выборе генерального элемента отношение b_i/a_{ij} будет минимальным именно для этой строки. Ясно, что соответствующую свободную переменную увеличить никак нельзя и целевая функция при переходе в новую вершину не меняется, т.к. мы остаемся фактически в этой же вершине, хотя формально перешли в новую.

Такая задача называется вырожденной задачей. Вырождение отрицательно сказывается на эффективности вычисления. Признаком вырожденности является равенство 0 некоторых свободных членов. В этом случае может произойти 2 отрицательных явления:

1. «Пробуксовка». Переходим в новую вершину, где равна нулю другая совокупность переменных, а на самом деле остаемся в той же точке. Значение ЦФ не меняется.

2. «Зацикливание» алгоритма. Через некоторое количество операций мы можем прийти к первоначальной таблице, следовательно, если алгоритм не менять, то зацикливаемся.

Существует несколько способов борьбы с зацикливанием:

- Использование степени свободы алгоритма. Например, если в алгоритме сказано: «берем первый по порядку…» - то, если обнаруживается зацикливание, можно взять «второй по порядку». Аналогично, если сказано «берем любой…».
- «Зашумление» коэффициентов задачи ЛП. Прибавляем матрицу малых случайных величин к матрице А. Получаем А + ξ, где А – матрица коэффициентов; ξ – матрица очень малых случайных величин. Тогда матрица коэффициентов СТ не будет содержать одинаковых чисел и после вычислений появление нулей будет маловероятным. Подробнее этот сложный вопрос освещен в специальной литературе.

2.9. Двойственная задача ЛП

Важным вопросом анализа полученного решения ЛП является анализ чувствительности решения к параметрам модели (коэффициентам целевой функции, свободным членам ограничений, коэффициентам a_{ij}). Теория этого вопроса тесно связана с так называемой двойственной задачей ЛП. Двойственная задача ЛП получается из прямой и она имеет физический смысл, когда прямая задача – задача об использовании ресурсов. Имеются ресурсы m типов в количестве $b_1, \ldots b_m$. Для изготовления одного изделия j – го типа расходуется a_{ij} сырья i-го типа. Каждое j – е изделие продается по цене c_j. Ставится задача максимизации стоимости проданных изделий.

$$L = \sum_{j=1}^{n} c_j x_j \to \max, \tag{2.11}$$

$$\sum_{j-1}^{n} a_{ij} x_j \le b_j, i = \overline{1,m}, \tag{2.12}$$

$$x_j \ge 0, \forall j.$$

Можно рассмотреть эту проблему по-другому. Пусть некоторые величины Δ_i – стоимости единицы i – го ресурса. Нужно определить эти стоимости так, чтобы выполнялись неравенства, выражающие то, что стоимость затраченного ресурса на j – е изделие была бы не меньше стоимости изделия и при этом общие затраты на ресурсы были минимальны.

$$L^* = \sum_{i=1}^{m} b_i \Delta_i \to \min, \tag{2.13}$$

$$\sum_{i=1}^{m} \alpha_{ij} \Delta_i \geq c_j, \quad j = 1,...,n, \tag{2.14}$$

Задача (2.13), (2.14) называется двойственной по отношению к задаче (2.11), (2.12) Она получается из прямой (2.11), (2.12) так: коэффициенты целевой функции двойственной задачи – это правые части прямой задачи и наоборот; а коэффициенты a_{ij} – суммируются по другому индексу (матрица коэффициентов двойственной задачи это транспонированная матрица прямой задачи). Направление оптимизации целевой функции меняется на противоположное.

Основное свойство задачи ЛП: оптимальные значения целевой функции прямой и двойственной задачи ЛП совпадают:

$$L^*_{опт} = \sum_{j=1}^{n} c_j x_j^* = \sum_{i=1} b_i \Delta_i^*$$

Оптимальное решение задачи ЛП находится в седловой точке:

$\max L$ по $x_j = \min\ L^*$ по Δ_i.

Существует и двойственный симплекс–метод решения задачи ЛП, который в ряде случаев эффективнее прямого [1, 10].

Экономический смысл двойственной задачи: какова цена единицы каждого ресурса Δ_i, чтобы при заданных количествах ресурсов b_i и стоимости единицы продукции c_j обеспечить минимум общей стоимости затрат.

Δ_i называется двойственной оценкой или теневой ценой. Если решение прямой задачи найдено, то Δ_i – есть коэффициенты, стоящие в строке L в последней оптимальной СТ.

Если $\Delta_i > 0$, то она показывает, насколько увеличивается значение целевой функции прямой задачи, если соответствующие запасы сырья i – го типа увеличиваются на единицу.

Соответствующая дополнительная переменная y_i при этом равна нулю. Это означает, что i – е ограничение выполняется точно (как равенство) и весь ресурс b_i тратится полностью.

При малом изменении коэффициентов модели оптимальное решение задачи ЛП не меняется, такое свойство называется устойчивостью. На рисунке 12 показано, что для ЦФ1 решением является точка А, для ЦФ3 *(при малом изменении коэффициентов c_j) также точка А, а для ЦФ2 решением уже будет точка В.

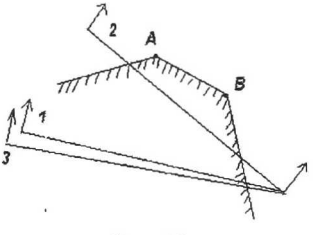


Рис. 12

Существуют специальные программы, которые позволяют проанализировать пределы изменения коэффициентов c_j, при которых оптимальное решение не изменяется, а значение целевой функции меняется. Аналогично проводится анализ по правым частям b_i.

Современные алгоритмы ЛП способны с использованием современных компьютеров решать задачи ЛП очень большой размерности. Многие из них учитывают архитектуру компьютера.

Наиболее широко известны блочные методы ЛП. При большой размерности матрица А будет содержать много нулей. В этом случае задачу можно решать по блокам. Вот почему параметрами сложности решаемой задачи ЛП являются: число переменных, число ограничений и число ненулевых коэффициентов ограничений. Число итераций симплекс- метода, за которое находится решение в бóльшей степени зависит от числа ограничений. Современные программы ЛП имеют удобный интерфейс, позволяющий моделировать и решать небольшие задачи в удобной «математической» или табличной форме. Для решения задач ЛП большой размерности используются режимы работы с крупными базами исходных данных. Например, программа LINDO, а также приложение «Поиск решения» (SOLVER) программы Microsoft EXCEL.

3. ТРАНСПОРТНЫЕ ЗАДАЧИ (ТЗ)

3.1. Математическая модель ТЗ по критерию стоимости

Симплекс–метод является общим для любой задачи ЛП. Но существуют специальные задачи ЛП, которые имеют более простой и наглядный способ решения.

Постановка транспортной задачи:

Имеется m пунктов $A_1 \ldots A_m$ отправления, в каждом из которых сосредоточено определенное количество груза $(a_1 \ldots a_m)$. Груз однородный, величина a_i называется запасом. Имеются n пунктов назначения $B_1 \ldots B_n$, где требуется количество груза $(b_1 \ldots b_n)$; величина b_i - заявка. Известна стоимость перевозки c_{ij} - из i – го пункта в j – й пункт.

Требуется так перевезти грузы, чтобы запасы не были превышены, заявки выполнены, и общая стоимость перевозки была бы минимальной.

Введем дополнительное требование – так называемое условие правильного баланса: $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$.

Модель:

x_{ij} - количество груза, перевозимого из i – го в j – й.

$$L = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij} \to \min, \tag{3.1}$$

$$\left.\begin{aligned} \sum_{j=1}^{n} x_{ij} &= a_i, i = \overline{1,m} \\ \sum_{i=1}^{m} x_{ij} &= b_j, j = \overline{1,n} \end{aligned}\right\}, \tag{3.2}$$

$$x_{ij} > 0, \forall i, j$$

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j. \tag{3.3}$$

Особенности модели ТЗ:

1. Наличие условия баланса (3.3) приводит к тому, что задача ТЗ всегда имеет решение.
2. Просуммировав левые части (3.2), получим (3.3), следовательно, одно уравнение в (3.2) линейно зависимо, значит ранг системы (3.2) будет $(n+m-1)$.

3. Коэффициенты при переменных в (3.2) все равны единице.
4. Подсчитаем число свободных и базисных переменных. Всего переменных $m \times n$. Базисных ($m+n-1$). Свободных переменных будет $m \times n - m - n + 1 = k$, ... $k=(m-1)\times(n-1)$. Т.е. практически все переменные свободные. Т.к. ТЗ является задачей ЛП, то в оптимальном решении все свободные переменные должны быть равны нулю, т.е. большинство маршрутов не будут задействованы. Если $m=100$ и $n=100$, всего маршрутов (переменных) 10 000, но не более 199 будут задействованы.

3.2. Нахождение опорного плана транспортной задачи

Все условия транспортной задачи можно записать в так называемой транспортной таблице.

В ней выполняются и все необходимые вычисления.

Пример:

Таблица 10

I	B_1	B_2	B_3	a_i
A_1	**4** 5	1 6	3	11
A_2	2	5 7	2	7
A_3	3	7 3	4 5	8
A_4	2	4	5 4	4
b_i	5	16	9	**30**

Стоимость перевозки

Перевозка (кол-во единиц в маршруте)

Назовем x_{ij} «перевозкой». Решение ТЗ, также как и ЛП состоит из 2 этапов:

1. Находится опорный план, т.е. допустимый план, соответствующий условию «вершины». В нем отличных от нуля перевозок не более ($m+n-1$).
2. Путем последовательного перехода от одного опорного плана к другому получаем оптимальный план.

Метод «северо-западного угла» (СЗУ) для нахождения опорного плана ТЗ

Он заключается в заполнении плана перевозок, начиная с самих верхних (северных) пунктов отправления за счет самых левых (западных) пунктов назначения. Величина перевозки определяется выражением:

$$x_{ij} = \min\{a'_i, b'_j\}, \qquad (3.4)$$

где a_i' и b_i' - остатки запаса и заявки после предыдущей итерации. Этот способ обеспечивает линейную независимость столбцов. Свойство допустимого плана: сумма перевозок по строке равна запасу, а по столбцу – заявке. План называется невырожденным, если отличных от нуля перевозок ровно $m+n$ -1. В вышеприведенном примере $m + n - 1 = 4 + 4 - 1 = 7=7$, т.е. план невырожденный.

Выполненные действия показаны в таблице 10

Определим стоимость плана:

$$L=4\times5+5\times4+1\times4+2\times2+3\times1+5\times3+5\times2=76.$$

Часто бывают такие транспортные задачи, где количество ненулевых перевозок (базисных клеток) меньше, чем $m + n - 1$.

Пример:

Таблица 11

	B_1	B_2	B_3	a_i	
A_1	3	4	(0)	7	+0
A_2			1	1	
A_3			2	2	
b_j	3	4	3		

Базисных клеток 4, а должно быть 5. Задача вырожденная. Будем дополнять план до невырожденного: в определенных местах проставим символическую бесконечно малую величину (0), чтобы выполнялась линейная независимость столбцов и условие невырожденности.

Чтобы автоматически определять место для дополнения плана будем избавляться от вырожденности в процессе заполнения плана в соответствии с выражением (3.4). Вырожденность появляется тогда, когда в формуле (3.4) будет $a_i'=b_j'$. В этом случае будем считать, что в строке $a_i'=a_i' + (0)$.

Метод минимального элемента

В методе СЗУ мы не использовали стоимость перевозок. Существуют методы, которые позволяют получить первый план сразу близким к оптимальному (т.е. с достаточно малой общей стоимостью перевозок).

Метод минимального элемента состоит в том, что порядок заполнения плана соответствует возрастанию стоимости. Но при этом запас исчерпывается полностью, т.е. перевозка ставится в соответствии с формулой (3.4) в клетке с минимальной стоимостью в строке, если не исчерпали запас, или в столбце, если не выполнили заявку.

Метод не обязательно гарантирует лучшее решение, чем метод С ЗУ, но, как правило, дает лучшие результаты.

Способ борьбы с вырождением такой же, как и в предыдущем методе.

Пример: Начинаем с самого дешевого маршрута (A_1B_4), далее в столбце B_4 ищем самый дешевый (A_4B_4), затем в строке A_4 минимальная стоимость 2 в маршруте (A_4B_2) и т.д.

Таблица 12

	B_1	B_2	B_3	B_4	a_i
A_1	5	4	3	1 9	9
A_2	2 1	4 2	2	9	3
A_3	8 3	7	1 5	3	8
A_4	4	2 4	3	2 1	5
b_j	4	6	5	10	25

L=9×1+1×2+2×4+3×8+5×1+4×2+1×2=58.

3.3. Оптимизация плана ТЗ, распределительный метод

Если в плане есть свободные клетки, для которых стоимость меньше, чем в базисных клетках, то можно передвинуть какую – либо перевозку в эту свободную клетку, но чтобы баланс не нарушался, нужно из строки или столбца, где будет новая базисная клетка, убрать такое же количество груза. Будем ставить (+) там, где перевозка увеличивается и (-) там, где уменьшается.

Пример:

Таблица 13

5	10 →	1
4 ←	3 6	5

Таблица 14

5	- 10	+	1
4	+ 3	- 6	5

Т.е. необходим перенос по циклу. Назовем циклом замкнутый маршрут, который показывает перенос груза по некоторым клеткам таблицы. В цикле всегда четное число вершин.

Назовем ценой цикла алгебраическую сумму стоимостей перевозок c_{ij}, стоящих в вершинах цикла с соответствующими знаками. Цена показывает, насколько увеличится стоимость перевозок в плане, если по данному циклу переносится единица груза. Если $\gamma<0$, то стоимость уменьшается. При переносе по циклу величины груза k стоимость уменьшается на $\Delta L=k\cdot\gamma$.

Будем рассматривать только правильные циклы: одна вершина в свободной клетке, а все остальные в базисных. В свободной клетке, знак всегда «+», в остальных вершинах цикла знаки чередуются.

Для невырожденной транспортной таблицы с независимыми столбцами справедлива теорема о том, что для любой свободной клетки существует правильный цикл и притом только один.

Таким образом, для любой свободной клетки можно построить правильный цикл и вычислить его цену. Если она больше нуля, то переходим к другой свободной клетке. Если $\gamma<0$, то строим цикл.

Очевидно, по циклу нужно перенести наибольшее количество груза. Эта величина k определяется как наименьшая из перевозок, стоящих в вершинах цикла со знаком «-».

Для нижеприведенного примера цена цикла 1-5+3-10 = -11. Величина k=5. Значит перенеся 5 единиц, мы уменьшаем общую стоимость на величину 5(-11)= -55 единиц.

Таблица 15

- 5	10	+	1
+ 4	3	- 6	5

Таблица 16

	10	5	1
9	3	1	5

Перенос груза по циклу соответствует симплекс–преобразованию. Метод непосредственного отыскания циклов с отрицательной ценой называется распределительным методом. Рассмотрим пример решения ТЗ этим методом

Пример:

Таблица 17

T1	B_1	B_2	B_3	B_4	a_i
A_1	4 2	2 - 1 +	1	1	3
A_2	5	4 3 -	3 2 +	2	5
A_3	1 +	2	1 1 -	4 5	6
b_j	2	4	3	5	

Таблица 18

T2	B_1	B_2	B_3	B_4	a_i
A_1	- 4 1	2 2	1	1 +	3
A_2	5	4 2	3 3	2	5
A_3	+ 1 1	2	1	4 5 -	6
b_j	2	4	3	5	

Для таблицы 17

$m + n - 1 = 3 + 4 - 1 = 6$;

L=2×4+1×2+3×4+2×3+1×1+5×4=49.

Строим правильный цикл для свободной клетки (3,1).

γ_{31}=1-4+2-4+3-1= -3;

k_{31}=1; ΔL = -3.

Для таблицы 18:

L=4+4+8+9+1+20=46.

Строим цикл для (1,4) получаем таблицу 19, далее строим цикл для клетки (3,3) и переходим к таблице 20. В таблице 20 строим цикл для клетки (2,4).

γ_{24}=2-4+1-3= -4;

k_{24}=1; ΔL = -4; L = 34.

Таблица 19

	B_1	B_2	B_3	B_4	a_i
A_1	4	2 -	1	1 +	3
A_2	5	4 +	3 -	2	5
A_3	1	2	1 +	4 -	6
b_j	2	4	3	5	

Дальнейшее решение предлагается проделать самостоятельно.

Очевидно, для того, чтобы процесс остановился, необходимо проверить все свободные клетки и убедиться, что цена циклов для каждой такой клетки больше нуля или равна нулю. С учетом того, что количество свободных клеток равно $(m-1)(n-1)$, этот процесс очень трудоемок.

Таблица 20

	B_1	B_2	B_3	B_4	a_i
A_1	4	2	1	1 3	3
A_2	5 4	4	- 3 1	2 +	5
A_3	1 2	2	+ 1 2	4 2 -	6
b_j	2	4	3	5	

Недостатком распределительного метода является необходимость перебирать циклы до встречи цикла с отрицательной ценой. Для доказательства оптимальности нужно перебрать и построить правильные циклы для всех свободных клеток. Если n=100, m=100, то свободных клеток будет 9801.

Рассмотрим метод, в котором автоматически находятся циклы с отрицательной ценой.

3.4. Метод потенциалов решения ТЗ

Пусть каждый пункт отправления A_i платит за вывоз единицы груза платеж α_i. И каждый пункт назначения B_j платит за привоз единицы груза β_j. Эти платежи передаются посреднику. Таким образом, на маршруте ij он получает $\alpha_i+\beta_j = \tilde{c}_{ij}$. Назовем эту величину псевдостоимостью. Оказывается, что суммарная псевдостоимость плана перевозок есть величина постоянная (не меняется с изменением плана). В этом заключается теорема о платежах.

Для доказательства запишем формулу псевдостоимости и перегруппируем слагаемые:

$$\tilde{L} = \sum_{i=1}^{m}\sum_{j=1}^{n}\tilde{c}_{ij}x_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n}(\alpha_i+\beta_j)\cdot x_{ij} =$$

$$= \sum_{i=1}^{m} \alpha_i \sum_{j=1}^{n} x_{ij} + \sum_{j=1}^{n} \beta_j \sum_{i=1}^{m} x_{ij} = \sum_{i=1}^{m} \alpha_i a_i + \sum_{j=1}^{n} \beta_j b_j = const_{x_{ij}}$$

Теорема о потенциальном плане ТЗ.

Пусть в базисных клетках стоимость перевозок равна псевдостоимости перевозок $c_{ij} = \tilde{c}_{ij}$, $x_{ij} \geq 0$, а в свободных клетках $\tilde{c}_{ij} \leq c_{ij}$, $x_{ij} = 0$. Тогда такой план является оптимальным и улучшен быть не может.

$$\left.\begin{array}{l} c_{ij} = \tilde{c}_{ij}, x_{ij} \geq 0 \\ \tilde{c}_{ij} \leq c_{ij}, x_{ij} = 0 \end{array}\right\}. \qquad (*)$$

Доказательство:

$L^* = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}^*$. Знак (*) означает, что выполняется условие (*).

Т.е. необходимо показать, что стоимость плана X^* является нижней границей и определяется только исходными данными задачи.

$L^* = \sum\sum \tilde{c}_{ij} x_{ij}^* = \underset{\{x_{ij}\}}{\text{const}}$ (т.к. в базисной $c_{ij} = \tilde{c}_{ij}$, а в пустой равно нулю).

Перейдем к другому произвольному плану

$$\{x_{ij}^*\} \to \{x_{ij}'\} : L' = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}' .$$

В этом плане перевозка из базисной клетки попала в свободную, где

$$c_{ij} \geq c_{ij}' \Rightarrow L' \geq L^* = \sum_{i=1}^{m} \sum_{j=1}^{n} \tilde{c}_{ij} x_{ij}^* = \text{const} .$$

Т.е. любое изменение плана от $\{x_{ij}^*\}$ приводит к увеличению стоимости, следовательно план $\{x^*_{ij}\}$ является оптимальным.

Идея метода потенциалов основывается на этой теореме и заключается в следующем:

1) Составим любой опорный план и, если это необходимо, то дополним до невырожденного.

2) Потребуем в базисной клетке $c_{ij} = \tilde{c}_{ij}$. Запишем в левом верхнем углу базисной клетки величины псевдостоимостей, равные стоимостям, стоящим в правом верхнем углу. Таких клеток $m + n - 1$.

3) Находим неизвестные нам платежи α_i и β_j из условий, что их сумма известна для каждой базисной клетки.

$$\left.\begin{array}{l} \alpha_i + \beta_j = c_{ij} \\ изв. - неизв. \\ \ldots \end{array}\right\} m + n - 1$$

В такой системе в каждой строке две неизвестных, а всего их $(m + n)$. Т.к. равенств всего $(m + n - 1)$, то решений бесконечное множество. Поэтому будем брать произвольное α_1 и полагать, например, $\alpha_1=0$. Тогда можно начать решение со строки, где содержится α_1, и так, последовательно переходя в строки с известным платежом, найти все платежи.

4) Найдя все платежи, подсчитываем $\alpha_i + \beta_j = c_{ij}$ для свободных клеток.

5) Проверяем условие (*) для свободных клеток. Там, где $\tilde{c}_{ij} > c_{ij}$, свободная клетка дает цикл с отрицательной ценой, т.к. $\gamma_{ij} = c_{ij} - \tilde{c}_{ij}$.

6) Выбираем цикл с наименьшим (наиболее отрицательным) γ_{ij} и составляем цикл. По циклу переносим k единиц груза и т.д. до выполнения условия (*).

Пример:

Опорный план построим по методу минимального элемента, вычисляем платежи и определяем псевдостоимости в свободных клетках (таблица 21). Видим, что в клетке (3,2) цена цикла отрицательная. Строим цикл, переходим в таблицу 22.

$$\gamma_{32} = -4,$$
$$k_{32} = 1; \Delta L = -3.$$

Таблица 21

	B_1	B_2	B_3	B_4	a_i	α_i
A_1	-2 4	3 2	-2 1	1 1 3	3	0
A_2	-1 5	4 4 4 -	-1 3	2 2 1 +	5	1
A_3	1 1 2	6 2 +	1 1 3	4 4 1 -	6	2
b_j	2	4	3	5		
β_j	-2	3	-2	1		

Таблица 22

	B_1	B_2	B_3	B_4	a_i	α_i
A_1	2 4	3 2	2 1	1 1 3 +0	3	0
A_2	3 5 3	4 4	3 3	2 2 2	5	1
A_3	1 1 2	2 2 1	1 1 3	0 4	6	-1
b_j	2	4	3	5		
β_j	2	3	2	1		

Заметим, что если при вычислении k минимальных перевозок две и более, то при переносе по циклу появятся дополнительные свободные клетки, т.е. план вырождается. Поэтому в этом случае считаем, что в одной такой клетке k, а в других k +(0) ((0) лучше поставить там, где дешевле маршрут).

Таблица 23

	B_1	B_2	B_3	B_4	a_i	α_i
A_1	1 4	2 2	1 1	1 1	3	0
		3		0		
A_2	2 5	3 4	2 3	2 2	5	1
				5		
A_3	1 1	2 2	1 1	1 4	6	0
	2	1	3			
b_j	2	4	3	5		
β_j	1	2	1	1		

План таблицы 23 оптимальный, т.к. выполняется условие (*).

3.5. Решение ТЗ с неправильным балансом

До сих пор мы считали, что в ТЗ выполняется условие правильного баланса, то есть $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$. На практике это никогда не выполняется. Но можно ТЗ с неправильным балансом привести к правильному балансу и показать, что оптимальный план, найденный для ТЗ с правильным балансом, является оптимальным и для ТЗ с неправильным балансом.

Рассмотрим 2 случая:

1. ТЗ с избытком запаса: $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$.

Будем считать, что излишки запасов якобы перевозятся в некоторый фиктивный пункт назначения.

Чтобы учесть фиктивность перевозки, достаточно считать, что стоимость перевозки в фиктивный пункт $B_ф$ равна нулю, $c_{iф}$=0. Тогда, если какие-то перевозки $x_{iф}$>0, то $\sum c_{iф} x_{iф} = 0$.

Таким образом, чтобы решать ТЗ с избытком запаса, нужно к таблице прибавить дополнительный столбец $B_ф$, заявка которого $b_ф = \sum a_i - b_j$, а $c_{iф}$=0, $i = \overline{1,m}$. Тогда для новой таблицы будет выполняться условие правильного баланса. После этого задача решается любым способом.

2. ТЗ с избытком заявок: $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$

Необходимо ввести фиктивный пункт отправления ($A_ф$), в котором якобы есть недостающий запас $a_ф = \sum b_j - a_i$, при этом $c_{фj}$=0, $j = \overline{1,n}$. После этого задача решается обычным способом.

Заметим, что при таком подходе в итоговом оптимальном плане не все заявки будут выполнены.

Иногда задача решается по-другому. Пусть $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$ Тогда можно вычислить коэффициент дефицита:

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

После этого каждую заявку можно скорректировать:

$$b_j' = b_j k,$$

$$\sum b_j' = \sum a_i.$$

При этом можно решать задачу с учетом приоритетов на заявку. Величина приоритетов j-го пункта назначения (d_j) может быть от 0 до 1 ($0 \le d_j \le 1$).

$$\sum_{j=1}^{n} d_j = 1.$$

Чем меньше d_j, тем выше приоритет.

$\Delta = \sum b_j - \sum a_i$,

Δ – величина дефицита.

$b_j' = b_j - d_j \Delta$.

Пример:

Рассмотрим ТЗ с избытком запасов (58<68), поэтому введем фиктивный столбец, находим опорный план методом минимального элемента, а далее находим цикл с отрицательной ценой (2,1).

Таблица 24

T1	B_1	B_2	B_3	a_i	$B_ф$
A_1	6	5 11	1 14	35	0 10
A_2	4 +	3 21 -	2	21	0
A_3	7 8 -	3 4 +	4	12	0
b_j	8	36	14		10

Если цикл с отрицательной ценой видно сразу, то не обязательно применять метод потенциалов.

k=8, γ=-3

Для таблицы 24 применим метод потенциалов:

Таблица 25

T2	B_1	B_2	B_3	$B_ф$	A_i	α_i
A_1	6 6	5 5 11	1 1 14	0 0 10	35	0
A_2	4 4 8	3 3 13	-1 2	-2 0	21	-2
A_3	4 7	3 3 12	-1 4	-2 0	12	-2
b_j	8	36	14	10		
β_j	6	5	1	0		

Видно, что во всех свободных клетках псевдостоимости $\leq c_{ij}$, следовательно, нет цикла с отрицательной ценой. План - оптимальный.

3.6. ТЗ по критерию времени, типы критериев

Пусть в ТЗ задано время t_{ij}, за которое груз можно перевезти из i – го в j-й пункт. Оно не зависит от количества перевозимого груза. Заданы запасы a_i, заявки b_j и необходимо составить такой план перевозок, чтобы общее время окончания перевозок было бы минимальным.

Т.к. перевозки осуществляются параллельно, окончание всех перевозок определяется длительностью маршрута с самой большой величиной t_{ij} среди занятых маршрутов (базисных клеток).

Таким образом, критерий в нашей задаче

$$T = (\max_{x_{ij}\geq 0} t_{ij}) \to \min . \tag{3.5}$$

Ограничения те же:

$$\left.\begin{aligned} \sum_{i=1}^{m} x_{ij} = b_j, j = \overline{1,n} \\ \sum_{j=1}^{n} x_{ij} = a_i, i = \overline{1,m} \end{aligned}\right\} \tag{3.6}$$

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \,. \qquad (3.7)$$

(3.5) – (3.7) – математическая модель ТЗ по критерию времени. Видно, что целевая функция $T(x_{ij})$ имеет нелинейный вид. При изменении плана $\{x_{ij}\}$ критерий T может не меняться. Критерии такого рода называют гарантирующими или максиминными (минимаксными). Они требуют оптимизации по так называемому «узкому месту». В этом смысле рассмотренный в предыдущих параграфах критерий общей суммарной стоимости перевозок является интегральным критерием.

Т.к. целевая функция $T(x_{ij})$ имеет нелинейный вид, то для решения ТЗ по критерию времени нельзя применить теорию линейного программирования. Для ее решения можно построить последовательность нескольких задач линейного программирования. Мы рассмотрим более простой эвристический алгоритм.

Эвристическими алгоритмами называют такие алгоритмы, которые обеспечивают разумность поиска оптимального решения, но не основываются на строгой математической теории и в связи с этим не вполне гарантируют результат (не доказана сходимость к оптимальному решению, не изучена точность и т.п.). Тем не менее, эвристические процедуры чрезвычайно распространены, например, при игре в шахматы, при составлении компьютерных программ и т.д.

Метод запрещенных клеток

1. Находим любой опорный план (необязательно невырожденный).
2. Вычисляем величину критерия для полученного плана $T_1 = \max\limits_{\{x_{ij}^1\}} t_{ij}$.
3. Запрещаем все маршруты в свободных клетках, у которых время $t_{ij} \geq T_1$.
4. Зафиксируем в плане занятую клетку с максимальным t_{ij} (узкое место).
5. Ищем цикл, при помощи которого можно «разгрузить» эту клетку и переносим по этому циклу перевозку величиной k. Циклы не обязательно должны быть правильными. Причем может быть, что освободить такую клетку можно только за несколько циклов .
6. Получаем план с новым значением $T_2 = \max\limits_{\{x_{ij}^2\}} t_{ij}$. Запрещаем свободные клетки, для которых $t_{ij} \geq T_2$ и переходим к п.4.

Повторяем до тех пор, пока будет невозможно составить ни одного цикла, разгружающего занятую клетку с максимальным временем.

Пример.

Для нижеприведенной ТЗ по критерию времени составляем первый опорный план, получаем T_1=3. Запрещаем все свободные клетки, у которых $t_{ij} \geq 3$. Используем неправильный цикл, получаем новый план, в котором T_2=2 .

Таблица 26

T1	B_1	B_2	B_3	a_i
A_1	3 19	1 1	0 +	20
A_2	2 +	1 10	2 30 -	40
A_3	3	2	1 15	15
b_j	19	11	45	

$T_1 = 3$

Таблица 26

T2	B_1	B_2	B_3	a_i
A_1	3	1 1	0 19	20
A_2	2 19	1 10	2 11	40
A_3	3	2	1 15	15
b_j	19	11	45	

$T_2 = 2$

Видно, что перевозку со временем t_{ij}=2 уже нельзя перенести ни в одну строку, ни в один столбец, поэтому последняя таблица является оптимальной.

4. ДИСКРЕТНОЕ ПРОГРАММИРОВАНИЕ

При моделировании различных задач возникает требование, чтобы какие-то управляемые переменные принадлежали специальному множеству.

1) Множество целых чисел – число людей, число датчиков, число самолетов и т.д.

2) Множество дискретных значений – это стандартные значения в производственных задачах, которые заранее зафиксированы (количество деталей в изделии, мощность электродвигателя и т.п.).

В таких задачах приходится искать значения переменных, которые не являются любыми действительными числами. Искомые переменные должны принадлежать заранее зафиксированному множеству значений D.

Такого рода задачи принадлежат классу задач дискретного программирования:

$$k = f(x,a) \to \min, \tag{4.1}$$

$$\varphi_i(x,a) \le 0, i = \overline{1,m}, \tag{4.2}$$

$$x \in D. \tag{4.3}$$

В связи с чрезвычайной сложностью решения таких задач в общем виде, мы рассмотрим только наиболее важные методы.

1) Универсальные методы решения задач линейного целочисленного программирования на основе «классических» алгоритмов.

2) Специальные методы (модели) дискретной оптимизации, которые имеют особое практическое значение.

Задачей линейного целочисленного программирования (ЛЦП) называют задачу вида:

$$L = \sum c_j x_j \to \min, \tag{4.4}$$

$$\sum a_{ij} x_j \le b_i, i = \overline{1,m}, \tag{4.5}$$

$$x_j \in Z, j = \overline{1,n}. \tag{4.6}$$

Если только часть переменных должны быть целыми, то задача называется частично целочисленной. Тогда в (4.6) вместо n будет $n_1 < n$.

Во многих задачах дискретность проявляется при моделировании задачи. Это относится к так называемым булевским переменным. В этом случае переменные принадлежат множеству $B = \{0,1\}$. Ограничения также будут иметь

логический характер. Но при этом все действия с этими переменными, определяемые математической моделью, подчиняются обычным правилам арифметики (т.е., например, 1+1=2, а не 0 как в булевой алгебре). Т.о. вместо (4.6) или в дополнение к нему в модели может присутствовать требование

$$x_j \in \{0,1\}. \qquad (4.7)$$

Если все переменные булевские, то задача (4.4), (4.5), (4.7) – задача линейного булевского программирования (ЛБП).

К этим задачам иногда сводят задачи целочисленного и дискретного программирования, т.к. для ЛБП разработаны эффективные быстродействующие алгоритмы (в том числе и с «аппаратной поддержкой»).

Чтобы привести задачу ЛЦП к ЛБП, достаточно каждую переменную представить в виде:

$$x_j = 2^0 x_{j1} + 2^1 x_{j2} + 2^2 x_{j3} + ... + 2^k x_{jk+1},$$

$$x_{ij} \in \{0,1\}.$$

Если x_i ограничено, то можно найти k, т.е. количество двоичных переменных, которыми можно описать исходные целочисленные переменные.

Задачу дискретного программирования можно привести к задаче булевского программирования так. Пусть

$$x_j \in k_{j1}, k_{j2}, k_{j3}, ..., k_{jn}$$

Тогда можно ввести булевские переменные y и заменить

$$x_j = y_1 k_{j1} + y_2 k_{j2} + ... + y_n k_{jn}, y_r \in \{0,1\}, \sum_{r=1}^{n} y_r = 1.$$

4.1. Особенности задач дискретного программирования

Рассмотрим геометрический образ задачи ЛЦП для двух переменных (рис. 13).

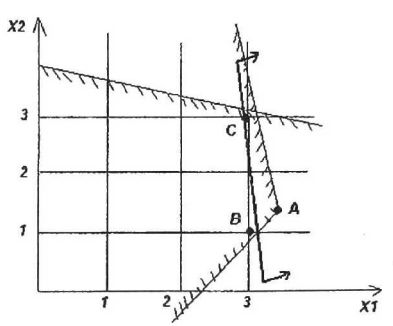


Рис. 13

ОДР этой задачи представляет собой только целочисленные точки внутри ОДР соответствующей задачи ЛП.

Решение задачи ЛП без требования целочисленности будет находиться в точке А (см. главу 2).

Возникает вопрос, нельзя ли решать так: найти решение задачи ЛП (назовем его «непрерывным» решением), а затем округлить до ближайшего целого? Покажем, что ответ на этот вопрос отрицательный.

1) При округлении решения до ближайшего целого может быть получено недопустимое решение.

2) Даже, если при округлении решение является допустимым, может оказаться, что существует другое допустимое целочисленное решение , которое лучше округленного. Из рисунка видно, что ближе к непрерывному решению A целочисленная допустимая точка B, а лучше C. Оптимальное решение: $x^*=(3,3)$. Иногда кроме целочисленности переменной, требуется, чтобы целой была L. Тогда тем более округление не дает правильного результата.

Пример: $z=\frac{1}{6}x_1+x_2\to\max$,

$$-\frac{1}{2}x_1+x_2\le 2,\ \frac{1}{5}x_1+x_2\le 3,\ x_1-\frac{1}{3}x_2\le 4.$$

Ответ: $x_1=1\frac{3}{7}, x_2=2\frac{5}{7}$.

Округлив: $x_1=1, x_2=3$ - это не удовлетворяет решению!

Ответ (целочисленный): $x_1=4, x_2=2$.

Чтобы Z было целым: $x_1=0, x_2=0$.

Таким образом, для нахождения целочисленных, оптимальных решений требуется непосредственно находить их на множестве целых чисел.

4.2. Примеры моделей задач дискретного программирования

Заметим, что обычная ТЗ при целых коэффициентах a_i, b_i дает целочисленное решение. Рассмотрим некоторые наиболее распространенные задачи дискретного программирования.

1. Задача о назначениях (о выборе).

Имеется n работ и n кандидатов на эти работы. Известен эффект того, что i – ая работа будет занята j – м работником c_{ij}. Необходимо распределить работы по работникам так, чтобы суммарный эффект был максимальным.

Управляемые переменные:

$$x_{ij}=\begin{cases}1, i\to j,\\ 0, i\not\to j.\end{cases} \tag{4.8}$$

$$z=\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij}\to\max. \tag{4.9}$$

Т.к. каждая работа может быть занята только одним работником и каждый работник может занимать только одну работу, то

$$\left.\begin{aligned}\sum_{j=1}^{n} x_{ij} = 1, i = \overline{1,n}\\ \sum_{i=1}^{n} x_{ij} = 1, i = \overline{1,n}\end{aligned}\right\} \qquad (4.10)$$

Это частный вид ТЗ. Любой опорный план будет сильно вырожден, в нем будет только n базисных переменных (должно быть $2n-1$).

Эту задачу можно свести к выбору n нулевых элементов, по одному в каждом столбце и каждой строке («правильному выбору»), в некоторой матрице с неотрицательными элементами, у которой в каждой строке и каждом столбце есть нули.

Нам удобно будет предварительно перейти от данной задачи выбора на максимум к задаче выбора с теми же условиями, но на минимум, т. е. от матрицы $C=(c_{ij})$ перейти к матрице $-C=(c_{ij})$ и искать выбор, дающий минимальную сумму элементов.

Теперь перейдем от задачи на минимум с матрицей $-C$ к задаче на минимум с эквивалентной ей матрицей, которая имела бы только неотрицательные элементы, и в каждом столбце и каждой строке которой было бы хотя бы по одному нулевому элементу. Для этого сначала прибавим к каждому столбцу матрицы $-C$ наибольший из элементов соответствующего столбца матрицы C (или, что то же, вычтем элементы каждого столбца матрицы C из наибольшего элемента этого столбца). Получится неотрицательная матрица C_1, в каждом столбце которой есть хотя бы один нуль. Теперь вычтем из каждой строки матрицы $-C_1$ минимальный элемент этой строки. Полученная матрица D и будет неотрицательной матрицей, в каждом столбце и каждой строке которой есть хотя бы один нуль.

Итак, проделываем следующие два последовательных преобразования матрицы $-C$:

$$C = (c_{ij}) \xrightarrow{(1)} C_1 = (c'_{ij}) = (\max_i c_{ij} - c_{ij}) \xrightarrow{(2)} D = (d_{ij}) = (c'_{ij} - \min_j c'_{ij}). \qquad (2.17)$$

Назовем эти преобразования *предварительными преобразованиями*. Если в каждой строке матрицы $C=(c'_{ij})$, полученной после первого преобразования матрицы C, есть хотя бы один нуль, то второе преобразование, конечно, не изменит матрицу C и практически не производится.

Наименьшее возможное значение суммы n элементов неотрицательной матрицы равно, очевидно, нулю. Таким образом, наша задача сводится теперь к выбору в матрице D, или в эквивалентной ей матрице с неотрицательными элементами, n нулевых элементов, по одному в каждой строке и каж-

дом столбце. Покажем, как это можно сделать. Неформальный смысл приводимого ниже алгоритма заключается в последовательных переходах от одного правильного неполного выбора нулей к другому, содержащему на один нуль больше, чем предыдущий, до тех пор, пока не получится полный правильный выбор. При этом на отдельных этапах может потребоваться переход к новой матрице, эквивалентной предыдущей.

Приведем алгоритм решения задачи о назначениях. Пусть уже проделаны предварительные преобразования матрицы эффективностей C данной задачи и получена неотрицательная матрица D, содержащая хотя бы по одному нулевому элементу в каждой строке и в каждом столбце.

1. Отмечаем (например, звездочкой) какой-нибудь нуль в первом столбце матрицы D (0*); отмечаем звездочкой какой-нибудь нуль во втором столбце, не лежащий в той строке, в которой находится 0* из первого столбца (если такой нуль во втором столбце найдется); отмечаем звездочкой один из нулей третьего столбца, лежащий в строке, где нет еще нулей со звездочкой (если такой нуль в третьем столбце найдется); и так далее, пока не пройдем все столбцы матрицы.

Если число отмеченных звездочкой нулей равно n, то процесс окончен: места, занимаемые нулями со звездочкой, соответствуют n переменным x_{ij}, равным 1, в оптимальном решении исходной задачи.

Если нулей со звездочкой меньше n, то переходим к п.5.

2. Помечаем (например, знаком «+» сверху) столбцы матрицы, в которых есть 0*, и считаем эти столбцы *занятыми*.

В ходе процесса будут появляться и *занятые* строки. Элементы, стоящие на пересечении незанятого столбца и незанятой строки будем считать незанятыми; остальные элементы — занятыми.

Если в матрице нет незанятых нулей, то переходим к п. 5.

Если незанятые нули есть, то выбираем первый из них (просматривая поочерёдно строки матрицы слева направо). Отмечаем его каким-нибудь промежуточным значком (например, штрихом — 0'). Если в его строке нет нуля со звездочкой, то переходим к п. 4; если в его строке 0* есть, то переходим к п. 3.

3. Освобождаем (снимаем знак «+» и считаем снова незанятым) столбец, в котором находится 0*, лежащий в той же строке, что и отмеченный только что штрихом нуль. Помечаем (например, знаком «+» справа) строку, в которой находится наш 0', и считаем ее занятой. Возвращаемся ко второй части п. 2 (третий абзац п. 2).

4. Начиная с только что отмеченного 0', строим цепочку из нулей: от этого 0' по столбцу к 0*, от него по строке к 0', и т. д., пока это возможно. Цепочка оборвется (возможно, на первом же 0') на некотором 0'. Снимаем звездочки у нулей из цепочки и заменяем звездочками штрихи у нулей из цепочки. Новый набор нулей со звездочками содержит на один больше, чем предыдущий, и является также правильным.

Снимаем все пометки, кроме звездочек, и возвращаемся ко второй части п. 1 (второй абзац п. 1).

5. Отыскиваем минимальный элемент среди незанятых элементов матрицы (пусть он равен h) и вычитаем его из всех незанятых строк, а затем прибавляем ко всем занятым столбцам. Никакие пометки при этом не снимаются. Получается матрица, эквивалентная предыдущей и содержащая незанятые нули. Возвращаемся к третьей части п. 2 (четвертый абзац п. 2).

Рассмотрим пример использования алгоритма.

Пример. Найти оптимальный вариант назначений, если матрица эффективностей такова:

$$\begin{pmatrix} 2 & 3 & 3 & 5 & 4 \\ 4 & 2 & 4 & 6 & 2 \\ 2 & 2 & 2 & 4 & 3 \\ 4 & 3 & 4 & 3 & 5 \\ 0 & 1 & 0 & 2 & 0 \end{pmatrix}$$

Приводим цепочку матриц, получающихся в процессе решения задачи с соответствующими пометками. Снятие значка отмечено включением его в прямоугольник. Над стрелками переходов от матрицы указаны пункты алгоритма, которые использовались при соответствующих преобразованиях.

$$\begin{pmatrix} 2 & 3 & 3 & 5 & 4 \\ 4 & 2 & 4 & 6 & 2 \\ 2 & 2 & 2 & 4 & 3 \\ 4 & 3 & 4 & 3 & 5 \\ 0 & 1 & 0 & 2 & 0 \end{pmatrix} \xrightarrow{\text{предварит. преобр. (1)}} \begin{pmatrix} 2 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 3 \\ 2 & 1 & 2 & 2 & 2 \\ 0 & 0 & 0 & 3 & 0 \\ 4 & 2 & 4 & 4 & 5 \end{pmatrix} \xrightarrow{\text{предварит. пресбр. (2), п. 1., п. 2., п. 3}}$$

$$\rightarrow \begin{pmatrix} 2 & 0^* & 1 & 1 & 1 \\ 0^* & 1 & 0 & 0' & 3 \\ 1 & 0 & 1 & 1 & 1 \\ 0' & 0 & 0^* & 3 & 0 \\ 2 & 0 & 2 & 2 & 3 \end{pmatrix} \xrightarrow{+\ \text{п. 5 } (h=1),\ \text{п. 2, п. 3, п. 4}} \begin{pmatrix} 1 & 0^* & 0' & 0 & 0 \\ 0 & 2 & 0 & 0 & 3 \\ 0' & 0 & 0^* & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 2 \end{pmatrix} \xrightarrow{+\ \text{п. 1, п. 2, п. 3, п. 4}}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0^* \\ 0 & 2 & 0 & 0^* & 3 \\ 0^* & 0 & 0 & 0 & 0 \\ 0 & 1 & 0^* & 3 & 0 \\ 1 & 0^* & 1 & 1 & 2 \end{pmatrix} \xrightarrow{\text{результат}} \begin{pmatrix} 1 & 0 & 0 & 0 & 0^* \\ 0 & 2 & 0 & 0^* & 3 \\ 0^* & 0 & 0 & 0 & 0 \\ 0 & 1 & 0^* & 3 & 0 \\ 1 & 0^* & 1 & 1 & 2 \end{pmatrix}.$$

Процесс окончен, так как получилось $n = 5$ нулей со звездочкой. Оптимальный вариант назначений: $x_{15}=x_{24}=x_{31}=x_{43}=x_{52}=1$, остальные $x_{ij}=0$, т. е. первый механизм назначается на пятую работу, второй - на четвертую, третий - на первую, четвертый - на третью, пятый - на вторую.

Изложенный алгоритм получил в литературе название венгерского алгоритма (венгерского метода) решения задачи о назначениях. Если в исходной

задаче матрица C не квадратная, то необходимо дополнить ее до квадратной нулевыми элементами.

4.2.1. Задача о покрытии

Примером такой задачи является задача об определении минимального числа спутников, которые бы обеспечивали устойчивую связь для заданных городов. Другой пример – это задача о переводчиках и туристических группах. В группе известны языки, которые они понимают, а у переводчиков – какие языки знают. Нужно обеспечить все группы минимальным числом переводчиков.

Классическая задача о покрытии формулируется так.

Задан граф с ребрами и узлами. Найти такое подмножество ребер, которое инцидентно любой вершине этого графа и при этом, чтобы количество ребер в этом подмножестве (называемым покрытием) было бы минимальным. Тогда если j - номер ребра, а $П$ - множество ребер покрытия, то

$$x_j = \begin{cases} 1, j \in П \\ 0, j \notin П \end{cases}.$$

Построим математическую модель. Целевая функция имеет вид

$$z = \sum x_j \to \min. \tag{4.11}$$

А - матрица инцидентности исходного графа, элементы которой

$$a_{ij} = \begin{cases} 1, i \approx j \\ 0, i \not\approx j \end{cases}.$$

Каждая вершина должна быть инцидентна хотя бы одному ребру покрытия

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1, i = \overline{1, m}. \tag{4.12}$$

Таким образом, мы получили задачу линейного булевского программирования (4.11), (4.12). Решение такой задачи рассмотрим ниже.

4.2.2. Задача о коммивояжёре

Задана сеть городов с расстояниями между ними. Торговец должен обойти все города, вернувшись в начальный город так, чтобы путь его был минимален. Необходимо найти порядок обхода городов. Эта ставшая классической задача находит чрезвычайно широкое применение при оптимизации порядка выполнения операций станков, выполнения заданий операционной системой компьютера и многого другого.

Одна из математических моделей этой задачи имеет вид

$$x_{ij} = \begin{cases} 1, i \to j, \\ 0, i \not\to j, \end{cases} \tag{4.13}$$

c_{ij} – расстояние между городами i и j.

$$z = \sum\sum c_{ij} x_{ij} \to \min. \tag{4.14}$$

Из каждого города торговец должен выти один раз и лишь один раз ойти, поэтому

$$\left.\begin{aligned} \sum_{j=1}^{n} x_{ij} = 1, i = \overline{1,n} \\ \sum_{i=1}^{n} x_{ij} = 1, j = \overline{1,n} \end{aligned}\right\}, \tag{4.15}$$

$$u_i - u_j + n x_{ij} \le n - 1, (1 \le i \ne j \le n), u_i \in R,$$

где u_i и u_j произвольные целые и неотрицательные числа.

Последнее ограничение – это условие цикличности маршрута, именно оно приводит к тому, что эта задача принадлежит к классу чрезвычайно сложных «переборных» задач. При некоторых коэффициентах решение приводит к полному перебору вариантов. Ниже мы рассмотрим общий метод решения подобных задач, а в [10] можно найти алгоритм и пример решения этой задачи.

4.2.3. Задача о раскрое материала

Эта задача впервые была поставлена и изучена нобелевским лауреатом Л.В.Канторовичем. Задачу о раскрое материалов необходимо рассматривать для производства комплектов продукции.

Заданы заготовки некоторого материала в виде нескольких партий различного размера. α_i - количество заготовок i – й партии (i – я партия имеет свой размер). Под размером можно понимать в общем случае различные величины, в простейшем случае «стержневого раскроя» - это длина стержня.

Из заготовок необходимо выкроить детали (рис. 14). Всего деталей n. $j = \overline{1,n}$.

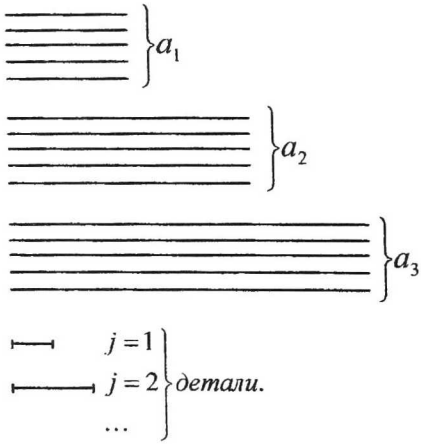


Рис. 14

Из деталей делается комплект. Известно количество деталей j типа, входящих в комплект l_j.

Необходимо так раскроить все заготовки, чтобы количество комплектов, полученных из раскроенных заготовок, было максимальным.

Существует несколько способов моделирования такой задачи (2-индексная и 3-индексная).

2 - индексная модель предполагает, что мы сначала рассматриваем различные способы раскроя (при этом можно рассматривать раскрой деталей сложной формы) Для фиксированного способа можно подсчитать количество деталей j – го типа для каждой i – й заготовки при k – м способе раскрое: a_{ijk}. Например,

$a_{111}=2$, $a_{121}=0$.

$a_{112}=0$, $a_{122}=1$.

Целесообразно отбросить способы раскроя, при которых величины a_{ijk} одинаковы, чтобы уменьшить размерность.

Интересно, что при помощи предварительного вычисления величин a_{ijk} можно решать не только «стержневые» задачи, но и «плоскостные», для этого достаточно перечислить различные способы раскроя плоской заготовки i-го типа на множество плоских деталей из j типов. Далее сама задача оптимального раскроя решается универсальным способом.

Рассмотрим величину x_{ik} - количество заготовок i - го типа, раскраиваемых k – м способом – это управляемые переменные.

Обозначим z_i - количество деталей j – го типа, полученных из всех типов заготовок при всех способах раскроя .

$$\sum_{k=1}^{n}\sum_{i=1}^{m} a_{ijk} \cdot x_{ik} = z_j ,$$

где $[\, z_j / l_j \,]$ - число комплектов, которое можно изготовить из деталей j-го типа. Знак [.] обозначает ближайшее целое с недостатком.

Выберем тип детали, которых меньше всего входит в комплект, в качестве основного, тогда можно максимизировать общее их количество

$$r = \arg\min_j l_j ,$$

$$z_p = \sum\sum a_{irk} \cdot x_{ik} \to \max ,$$

при выполнении условия комплектности:

$$\left.\begin{array}{l} \dfrac{z_1}{l_1} = \dfrac{z_2}{l_2} \\ \dfrac{z_2}{l_2} = \dfrac{z_3}{l_3} \\ \cdots \\ \dfrac{z_m}{l_m} = \dfrac{z_n}{l_n} \end{array}\right\} .$$

Кроме этого количество заготовок ограничено:

$$\sum_{k=1}^{n} x_{ik} \le a_i, i = \overline{1,m} .$$

Вышеприведенная модель требует строгой пропорциональности деталей для производства комплектов. Но это может оказаться слишком строгим ограничением: задача может не иметь решения или же решение не самое лучшее. Поэтому можно поставить задачу в более свободной форме.

Пусть z – число комплектов.

$$z \to \max , \qquad (4.16)$$

$$\frac{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ijk} \cdot x_{ik}}{l_j} \le z, j = \overline{1,n} , \qquad (4.17)$$

$$\sum_{k=1}^{n} x_{ik} \le a_i, i = \overline{1,m}\,. \tag{4.18}$$

В этой модели z дополнительная переменная, причем,так как ограничения (4.17) более «слабые», ОДР более широкая, и оптимальное количество комплектов может быть большим, чем в первой модели. Видно, что обе модели задачи раскроя материалов - это задачи ЛЦП, решение которых подробно рассмотрим ниже. Задача получения величин a_{ijk} решается отдельно для каждого типа заготовок и деталей. Для простых заготовок и деталей (стержень, прямоугольники) существую простые алгоритмы, для изделий сложной формы существуют эвристические алгоритмы и специализированные программные системы.

4.2.4. Задача о ранце

В ранец объема b необходимо упаковать предметы из имеющегося множества предметов разного типа так, чтобы суммарный вес ранца был максимальным. Пусть j – тип предмета, который занимает объем a_j и весит c_j.

Введем управляемые переменные x_j - количество предметов j – го типа, которые будут в ранце. Общий вес ранца

$$z = \sum_{j=1}^{n} c_j x_j \to \max\,. \tag{4.19}$$

Ограничение по объему ранца

$$\sum_{j=1}^{n} a_j x_j \le b\,, \tag{4.20}$$

x_j - целое. (4.21)

Это задача ЛЦП при единственном ограничении.

4.3. Алгоритм решения задачи о ранце

Рассмотрим специальный метод решения задачи о ранце, т.к. в ней лишь одно ограничение. При этом в этом методе будут введены важнейшие понятия, необходимые для разработки общего метода решения задач ЛЦП.

Любое решение задачи о ранце это вектор, например: $\bar{x}(6, 0, 2, 4)$. Возьмем предметы с большим удельным весом в качестве приоритетных. Упорядочим компоненты по удельному весу.

$$\frac{c_j}{a_j}\left|\frac{c_1}{a_1} > \frac{c_2}{a_2} > \ldots \frac{c_n}{a_n}\right..$$

Старшая компонента – с большим удельным весом.

Определим лексикографический порядок таких векторов, это упорядочение векторов по первой компоненте, если они равны, то по второй и т.д.

Больший вектор тот, у которого старшая компонента больше.

Лексикографический максимальный вектор для задачи о ранце определяется так:

$$\bar{x}^0 = (x_1^0, x_2^0, ..., x_n^0),$$

$$x_1^0 = \left[\frac{b}{a_1}\right],$$

$$x_2^0 = \left[\frac{b - a_1 \cdot x_1^0}{a_2}\right],$$

...

$$x_k^0 = \left[\frac{b - \sum_{j=1}^{k-1} a_j \cdot x_j^0}{a_k}\right].$$

Следующий после максимального в лексикографическом ряду будет вектор, который получается так: последнюю ненулевую компоненту уменьшают на единицу, а следующую увеличивают на максимально возможную величину.

Несмотря на то, что лексикографически максимальный вектор дает близкий к оптимальному значению вес ранца, целевая функция для векторов, лежащих ниже, чем $\bar{x}^0$, может быть больше, чем целевая функция для $\bar{x}^0$. В этом проявляется специфика целочисленности, т.к. если из ранца убрать один «тяжелый» предмет, то в сумме два более «легких» могут дать больший вес (они плотнее упакуются).

Если пройти все лексикографически упорядоченные векторы , то можно найти лучшее значение *z*.

Рассмотрим подход, позволяющий отсеять некоторое подмножество вариантов. Следует отметить, что все методы дискретной оптимизации направлены на сокращение перебора вариантов решения. Для этого мы должны быть уверены, что все эти отсеиваемые варианты ни в каком случае не дадут значение целевой функции лучше, чем уже полученное. Для этого используется понятие оценки множества вариантов.

Оценка подмножества вариантов – это величина, имеющая размерность целевой функции, такая, что ни один вариант этого подмножества не даст значения целевой функции лучше, чем данная оценка (δ).

Грубую оценку сделать легко. Вопрос в том, чтобы получить оценку как можно ближе к истинным значениям. Чем точнее оценка, тем больше будем отсеивать вариантов.

Рассмотрим в лексикографическом ряду вектор, стоящий на i – м месте, $\overline{x}^i$. Его последняя ненулевая компонента x_s^i. Рассмотрим вектор y^i, который получается из вектора x^i так: последняя ненулевая компонента уменьшается на единицу $y_s^i = x_s^i - 1$, все дальнейшие – нули.

Сравним y^i и x^{i+1}.

Пример:

$\overline{x}^i = (6, 0, 5, 3, 0, 0)$,

$\overline{x}^{i+1} = (6, 0, 5, 2, 9, 8)$,

...

$y^i = (6, 0, 5, 2, 0, 0)$.

Дадим оценку всех вариантов, находящихся от x^i до y^i. Для этого определим оценку i – го множества в виде:

$$\delta^i = \sum_{j=1}^{n} c_j y_j^i + \frac{c_{s+1}}{a_{s+1}}(b - \sum_{j=1}^{n} a_j y_j^i).$$

Первая сумма – это вес упаковки y^i, а выражение в скобках – объем, оставшийся после упаковки y^i, поэтому оценка дает вес плотно заполненного ранца, что при целых предметах недостижимо.

Теперь, если для всех векторов от $\overline{x}^0$ до $\overline{x}^i$ получено некоторое наибольшее, т.н. рекордное значение целевой функции z^* и $\delta^i \le z^*$, это означает, что не надо рассматривать векторы от $\overline{x}^i$ до $\overline{y}^i$ и нужно взять сразу для анализа вектор $\overline{y}^i$. Если $\delta^i > z^*$, то ничего сказать нельзя, и берем для анализа следующий вектор $\overline{x}^{i+1}$, хотя в сохраненном подмножестве может и не быть лучшего значения целевой функции.

Пример:

$$Z = 2x_1' + 7x_2' + 4{,}5x_3' + x_4' \to \max,$$

$$6x_1' + 3x_2' + 2x_3' + x_4' \le 19.$$

Проверим соотношения $\frac{c_s}{a_s}$ и переобозначим переменные.

$$\frac{7}{3} > \frac{4{,}5}{2} > \frac{1}{1} > \frac{2}{6},$$

$$\begin{cases} 7x_1 + 4{,}5x_2 + x_3 + 2x_4 \to \max, \\ 3x_1 + 2x_2 + x_4 + 6x_1 \le 19. \end{cases}$$

$\bar{x}^0 = (6,0,1,0); z = 6\cdot 7 + 1\cdot 1 = 43 = z^*; s = 3; \delta^0 = 42 + \frac{1}{3}(19-18) = 42\frac{1}{3} < z^*;$

$\bar{x}^1 = (6,0,0,0) = \bar{y}^0; \bar{y}^1 = (5,0,0,0); s = 1; \delta^1 = 35 + \frac{4,5}{2}(19-15) = 35+9 = 44 > z^*;$

$\bar{x}^2 = (5,2,0,0); z = 5\cdot 7 + 2\cdot 4,5 = 44 = z^{**};$

$\bar{y}^2 = (5,1,0,0); s = 2; \delta^2 = 39,5 + 1(19-17) = 41,5 < z^*;$

$\bar{x}^3 = \bar{y}^2 = (5,1,0,0); \bar{y}^3 = (5,0,0,0); s = 2; \delta^3 = 35 + 1(19-15) = 39 < z^{**};$

$\bar{x}^4 = \bar{y}^3 = (5,0,0,0); \bar{y}^4 = (4,0,0,0); s = 1;$

$\delta^4 = 28 + \frac{4,5}{2}(19-12) = 28 + \frac{4,5\cdot 7}{2} = 28 + 15\frac{3}{4} = 43\frac{3}{4} < z^{**};$

$\bar{x}^5 = \bar{y}^4 = (4,0,0,0); \bar{y}^5 = (3,0,0,0); s = 1;$

$\delta^5 = 21 + \frac{4,5}{2}(19-9) = 21 + \frac{90}{4} = 21 + 22\frac{1}{2} = 43\frac{1}{2} < z^{**};$

$\bar{x}^6 = \bar{y}^5 = (3,0,0,0); \bar{y}^6 = (2,0,0,0); s = 1; \delta^6 = 14 + \frac{4,5}{2}(19-6) = 43\frac{1}{4} < z^{**};$

$\bar{x}^7 = \bar{y}^6 = (2,0,0,0); \bar{y}^7 = (1,0,0,0); s = 1; \delta^7 = 7 + \frac{4,5}{2}(19-3) = 38\frac{1}{2} < z^{**};$

$\bar{x}^8 = \bar{y}^7 = (1,0,0,0); \bar{y}^8 = (0,0,0,0); s = 1; \delta^8 = 0 + \frac{4,5}{2}\cdot 19 = 42\frac{3}{4} < z^{**};$

$\bar{x}^9 = (0,0,0,0).$

Ответ: (5, 2, 0, 0), с учетом переобозначения $x' = (0, 5, 2, 0)$.

Примечание: Если коэффициенты целевой функции целые, то истинное значение z может быть только целым, поэтому если оценка получилась дробной, то ее можно усилить (приблизить к истинной величине Z: для $Z \to \max$ уменьшить до ближайшего целого, а для $Z \to \min$ увеличить до ближайшего целого).

4.4. Решение задач ЛЦП методом отсечений Гомори

Алгоритмы решения задач ЛЦП используют в качестве подалгоритма симплекс – метод решения задач ЛП. Рассмотрим идею методов отсечения. Пусть дана задача ЛЦП (рис. 15). Отбросим требование целочисленности и решаем обычную задачу ЛП. Если решение получается целым, то оно является оптимальным. Если решение не удовлетворяет требованиям целочисленности, то применяют условие правильного отсечения. Для этого формируют новое ограничение, которое отсекает только что полученное оптимальное «непрерывное» решение, оставляя допустимыми все целочисленные решения.

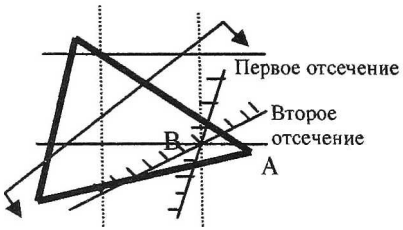


Рис. 15

Точка А – оптимальное «непрерывное» решение. Если его не отсечь, то оно появится на следующем этапе. Первое отсечение оставляет все целочисленные решения и отсекает точку А. Точка В – оптимальное целочисленное решение.

Как получить отсечение, чтобы получить такое решение?

Решая задачу ЛП, мы получаем в симплекс – таблице выражение базисных переменных через свободные.

$$x_i = b_i - \sum_{i \in S} \alpha_{ij} x_j, i = \overline{1,m}. \tag{4.22}$$

Пусть $x_j \in Z$, $j = \overline{1,n}$.

S – множество свободных переменных.

Коэффициенты a_{ij} - любые (целые, дробные).

Обозначим $\{a\}$ – дробная часть числа,

$[a]$ – целая часть числа (с недостатком).

$\{a\} = a - [a] > 0$ – всегда!

$$x_i = [b_i] + \{b_i\} - \sum_{j \in S} [\alpha_{ij}] x_j - \sum_{j \in S} \{\alpha_{ij}\} x_j = [b_i] - \sum_{j \in S} [\alpha_{ij}] x_j + \{b_i\} - \sum_{j \in S} \{\alpha_{ij}\} x_j. \tag{4.23}$$

Покажем, что если решение целое, то

$$\{b_i\} - \sum_{j \in S} [\alpha_{ij}] x_j \le 0. \tag{4.24}$$

(4.22) представляет собой запись симплекс-таблицы для оптимального решения, но решение не целое. Рассмотрим строку с нецелым b_i.

$$\{b_i\} \ne 0, 0 \le \{b_i\} < 1$$

Т.к. $\{\alpha_{ij}\} > 0, x_j > 0 \Rightarrow$, то величина (4.24) может быть только отрицательным целым числом.

Это необходимый признак целочисленности x_i. Это условие можно считать правильным отсечением. Неравенство (4.24) приведем к равенству, введя дополнительную переменную y_{n+1}:

$$y_{n+1} = -\{b_i\} + \sum_{j \in S} \{\alpha_{ij}\} x_j. \tag{4.25}$$

Если это ограничение добавить к предыдущим ограничениям , то оно отсекает полученное непрерывное решение. Т.к. в симплекс – таблице появляется отрицательный свободный член, то надо искать новое опорное и новое оптимальное решения.

Алгоритм Гомори:

1. Решаем задачу ЛП. Находим оптимальное решение. Если в оптимальной симплекс – таблице все свободные члены не меньше нуля и целые, то получено оптимальное целочисленное решение. Иначе переходим ко второму пункту.

2. Находим строку с наибольшей дробной частью $\{b_i\}$ для этой строки по (4.25) составляем правильное сечение и записываем дополнительной строкой в симплекс-таблицу; переходим к п.1.

При целых коэффициентах задача ЛП сходится за конечное число шагов к точному решению.

Если в исходной задаче коэффициенты дробные и есть ограничения неравенства, то дополнительные переменные y_i могут быть дробными и задача не является полностью целочисленной. В этом случае может получиться, что если потребовать $y \in Z$ и построить отсечение (4.25), то целых решений может не быть. Тогда ограничения можно ослабить, считая $x \in Z$, $y \in R$.

$$y'_{n+1} = -\{b_i\} + \sum_{j \in S} \{\alpha_{ij}\} x_j + \sum_{l \in S} \{\alpha_{ij}\} y_l . \qquad (4.26)$$

(4.26) – ослабленное условие.

Если требуется, чтобы значение целевой функции было целым, то строим условие правильного отсечения по строке L.

Пример:

Решить задачу

$$L = -x_1 + 3x_2 \to \min,$$

$$2x_1 + x_2 \le 4,$$

$$4x_1 - 3x_2 \le 2,$$

$$-3x_1 + 2x_2 \le 3,$$

$x_1, x_2 \ge 0$ и целые числа.

Сначала решим данную задачу без требования целочисленности симплекс-методом. Для этого запишем задачу в виде основной задачи линейного программирования (ОЗЛП) в стандартной форме:

$$L = 0 - (x_1 - 3x_2) \to \min,$$

$$y_1 = 4 - (2x_1 + x_2),$$

$$y_2 = 2 - (4x_1 - 3x_2),$$

$$y_3 = 3 - (-3x_1 + 2x_2).$$

Далее заполним симплекс-таблицу (табл. 27).

Таблица 27

	1	x_1	x_2
L	0 $-\frac{1}{2}$	1 $-\frac{1}{4}$	-3 $\frac{3}{4}$
y_1	4 -1	2 $-\frac{1}{2}$	1 $\frac{3}{2}$
→ y_2	2 $\frac{1}{2}$	4 $\frac{1}{4}$	-3 $-\frac{3}{4}$
y_3	3 $\frac{3}{2}$	-3 $\frac{3}{4}$	2 $-\frac{9}{4}$

Это опорное решение, так как в столбце свободных членов при базисных переменных все коэффициенты положительны. Но решение не оптимально, так как в строке функции цели есть положительный коэффициент при свободной переменной. Ищем оптимальное решение.

Делаем все необходимые преобразования в симплекс-таблице для замены y_2 на x_1, и переходим к табл.28.

Это оптимальное решение задачи без требования целочисленности, так как в столбце свободных членов нет отрицательных коэффициентов при базисных переменных, а в строке функции цели L нет положительных коэффициентов при свободных переменных.

Таким образом, $x_{1опт} = \frac{1}{2}; x_{2опт} = 0; L = -\frac{1}{2}$.

Но это решение не удовлетворяет требованию целочисленности. Для построения правильного отсечения и ввода нового ограничения берется первое по номеру нецелочисленная переменная. В данном случае это x_1.

Выпишем x_1 из табл.28:

Таблица 28

	1	y_2	x_2
L	$-\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{9}{4}$
y_1	3	$-\frac{1}{2}$	$\frac{5}{2}$
x_1	$\frac{1}{2}$	$\frac{1}{4}$	$-\frac{3}{4}$
y_3	$\frac{9}{2}$	$\frac{3}{4}$	$-\frac{1}{4}$

$$x_1 = \frac{1}{2} - \left(\frac{1}{4}y_2 - \frac{3}{4}x_2\right),$$

$$\left\{\frac{1}{2}\right\} = \frac{1}{2} - \left[\frac{1}{2}\right] = \frac{1}{2} - 0 = \frac{1}{2},$$

$$\left\{\frac{1}{4}\right\} = \frac{1}{4} - \left[\frac{1}{4}\right] = \frac{1}{4} - 0 = \frac{1}{4},$$

$$\left\{-\frac{3}{4}\right\} = -\frac{3}{4} - \left[-\frac{3}{4}\right] = -\frac{3}{4} - (-1) = \frac{1}{4}.$$

Применяя формулу (4.25) , получим $y_4 = -\frac{1}{2} + \left[\frac{1}{4}y_2 + \frac{1}{4}x_2\right]$ - это новое ограничение, которое определяет правильное отсечение. Перепишем это ограничение в стандартном виде $y_4 = -\frac{1}{2} - \left(-\frac{1}{4}y_2 - \frac{1}{4}x_2\right)$ и допишем его в симплекс-таблицу 29.

Таблица 29

	1	y_2	x_2
L	$-\frac{1}{2}$ $\frac{1}{2}$	$-\frac{1}{4}$ -1	$-\frac{9}{4}$ $\frac{1}{4}$
y_1	3 4	$-\frac{1}{2}$ -2	$\frac{5}{2}$ $\frac{1}{2}$
x_1	$\frac{1}{2}$ $-\frac{1}{2}$	$\frac{1}{4}$ 1	$-\frac{3}{4}$ $-\frac{1}{4}$
y_3	$\frac{9}{2}$ $-\frac{3}{2}$	$\frac{3}{4}$ 3	$-\frac{1}{4}$ $-\frac{3}{4}$
y_4	$-\frac{1}{2}$ 2	$-\frac{1}{4}$ -4	$-\frac{1}{4}$ 1

Для решения этой расширенной задачи без требования целочисленности необходимо найти опорное решение, так как в столбце свободных членов есть отрицательные элементы при базисных переменных. Делаем необходимые преобразования в табл. 29 для нахождения опорного решения и переходим к табл. 30.

Решение оптимально и целочисленно: $x_{1опт} = 0; x_{2опт} = 0; L = 0$.

Геометрическая интерпретация данного примера приведена на рисунке 16.

Таблица 30

	1	y_4	x_2
L	0	-1	-2
y_1	4	-2	3
x_1	0	1	-1
y_3	3	3	-1
y_2	2	-4	1

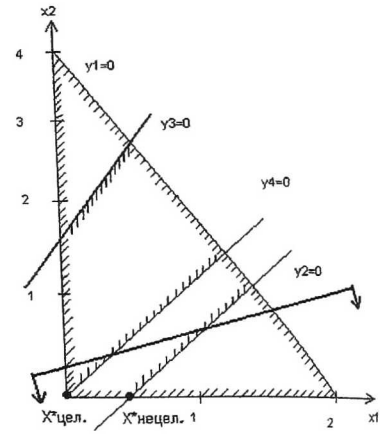


Рис. 16

4.5. Метод ветвей и границ (МВГ)

Метод ветвей и границ (МВГ) – это наиболее широко применяемый метод дискретного программирования. Он относится к комбинаторным методам. МВГ – метод последовательного анализа вариантов (1960 г. – Лэнд и Дойг).

Широкое распространение получил, когда с помощью этого метода был разработан алгоритм решения задачи коммивояжера.

Название МВГ раскрывает основное содержание метода, он состоит из двух процедур:

1. ветвление вариантов решения;
2. установление границ возможного для каждого анализируемого подмножества решений.

В любой задаче все множество решений представляется в виде дерева вариантов, т.е. вводится принцип разбиения множества вариантов на подмножества вариантов.

В каждой задаче может быть свой принцип разбиения. Для МВГ необходимо, чтобы, во-первых, был удобный способ перехода от множества к подмножеству, во-вторых, чтобы варианты не повторялись и чтобы ни один вариант не был упущен.

Переход от одного варианта к другому осуществляется в соответствии с некоторым порядком (например, в лексикографическом порядке).

Исходя из свойств задачи физического смысла целевой функции устанавливаются границы возможного (иногда называемого «обещанием») для каждого подмножества, т.е. придумывается формула оценки для заданного подмножества.

Оценкой множества вариантов называется такое значение ЦФ, которое не может быть улучшено ни на одном из вариантов оцениваемого множества. Оценка множества всегда не хуже оценки входящего в него подмножества. В частности, если в некоторой задаче добавляется ограничение, то оценка для этой задачи может только ухудшиться или остаться неизменной.

Оценка должна быть как можно более точной, т.е. приближаться к истинным значениям как можно ближе. Чем точнее оценка, тем чаще отсеиваются варианты. Действительно, если получено некоторое рекордное значение ЦФ z^*, а наилучшая из оценок оставшихся для проверки подмножеств хуже z^*, то, очевидно, процесс дальнейшего улучшения z можно прекратить. Чтобы это произошло как можно раньше, во-первых, надо чтобы первый рекорд был достаточно «хорошим», а во-вторых, оценки были как можно более точными.

Рассмотрим МВГ применительно к любой задаче дискретного программирования:

$$z = f(\bar{x}) \to \min, \tag{4.27}$$

$$\bar{x} \in G. \tag{4.28}$$

G – некоторая область, которая определяется системой ограничений неравенств, а также типом переменных.

На первом этапе вычисляют нижнюю границу целевой функции. Для этого снимают ограничение на тип переменных. Целевая функция может быть при этом только лучше (или равной), поэтому ее можно принять за оценку.

Если $\xi(G)$ - оценка множества G, то для задачи на min $f(x) \geq \xi(G)$.

На втором этапе производится разбиение (ветвление) множества G. Само разбиение идет в процессе решения и зависит от результатов, которые получены в результате разбиения на непересекающиеся подмножества.

Рассмотрим k-ую итерацию. Для дальнейшего разбиения выбирают так называемое перспективное подмножество (например то, которое имеет наилучшую оценку). Заметим, что при разбиении на следующем уровне получаемые оценки могут быть только хуже (или те же).

Если $G_1 \subset G_0$, то $\xi(G_1) \geq \xi(G_0)$.

Как только на каком-то подмножестве вариантов, получено решение, удовлетворяющее всем ограничениям (в том числе специальному требованию на тип переменной),проверяется условие оптимальности МВГ.

Условие оптимальности МВГ. Если на каком–либо подмножестве $G_\upsilon^{(k)}$ на k – ой итерации получено некоторое решение $\bar{x}^*$ такое, что значение целевой функции равно оценке этого подмножества $f(\bar{x}^*) = \xi(G_\upsilon^{(k)})$, $\bar{x} \in G_\upsilon^{(k)}$, и при этом сама оценка является наилучшей среди всех оценок ещё не разбитых

подмножеств на этой итерации $\xi(G_\upsilon^{(k)}) = \min\{G_i^{(k)}\}$, при этом $\bigcup_i G_\upsilon^{(k)} = G_0$, то решение $\overline{x}^*$ является оптимальным и улучшено быть не может (Рис. 17).

Действительно, в самом множестве G_υ лучшего решения быть не может, а остальные оценки дают «обещание» решений хуже, чем $\overline{x}^*$.

В практических задачах можно ограничиться приближённым решением с точностью до ε.

Пусть получено некоторое решение $\overline{x}'$ и задана некоторая абсолютная точность решения ε:

$\left|f(\overline{x}') - f(\overline{x}^*)\right| < \varepsilon$, где $\overline{x}^*$ - неизвестное нам точное решение.

Если $\Delta = \left|f(x') - \min\{\xi(G_i^{(k)})\}\right| < \varepsilon$, то разница между точным оптимальным значением и приближённым тем более меньше ε.

Если $\Delta < \varepsilon$, то $\delta < \varepsilon$.

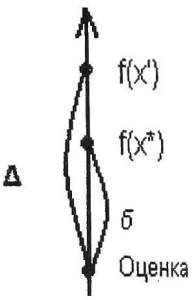


Рис. 17

4.6. Алгоритм МВГ для задачи ЛЦП

$$z = \sum_{j=1}^{n} c_j x_j \to \min, \tag{4.29}$$

$$\sum_{j=1}^{n} a_j x_j \le b_i, i = \overline{1,m}, \tag{4.30}$$

x_j - целое, $j = \overline{1,n_1}$, $n_1 \le n$. (4.31)

Для задачи ЛЦП (4.29), (4.30), (4.31) оценкой множества решений, определяемых (4.30), (4.31), является оптимальное значение для задачи ЛП (4.29), (4.30). На рисунке 18 точка А – решение задачи ЛП, а точка В – решение задачи ЛЦП. Кроме этого, для ЛЦП можно использовать простое правило ветвления вариантов. Получаем следующий алгоритм:

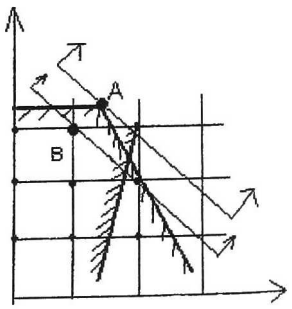


Рис. 18

1. Отбросим (4.31) и найдем решение задачи ЛП. Если решение автоматически удовлетворяет (4.31), то оно является оптимальным. Иначе переходим ко второму пункту.

2. Рассмотрим какую-нибудь нецелочисленную переменную полученного решения $x_l = d_l, l \in \overline{\{1, n_1\}}$. По этой переменной производится ветвление, все множество решений разбивается на два:

$$G_1^{(1)} : x_l \geq [d_l] + 1,$$

$$G_2^{(1)} : x_l \leq [d_l].$$

Пример:

$$x_5 = 4\frac{4}{7} \rightarrow \begin{cases} x_5 \leq 4, \\ x_5 \geq 5. \end{cases}$$

Снова решаем задачу на каждом подмножестве. Если оба решения не удовлетворяют (4.31), то в качестве перспективного выбирают то подмножество, где оценка лучше. Сравнение оценок производится среди всех еще не разбитых подмножеств. Как только на каком-нибудь подмножестве получено целочисленное решение, проверяем условие оптимальности. Если оно выполняется, то получено оптимальное решение. Если не удовлетворяется, то снова выбирают перспективное подмножество и продолжают поиск решения.

1. Если решения задачи ЛП нет, то соответствующее ее множество $G_l = \Theta$ пустое. Ему необходимо приписать оценку $+\infty$ (для $z \rightarrow \min$).

2. Если все коэффициенты целевой функции целые, то оценку множества можно заменить на более сильную. Для задачи на min – на величину $]\xi(G_l)[$ - ближайшее целое с избытком. Для $z \rightarrow \max$ на $[\xi(G_i)]$.

Пример:

$$z = -x_1 - x_2 \rightarrow \min, \tag{4.32}$$

$$\left.\begin{array}{l} 2x_1 + 11x_2 \leq 38 \\ x_1 + x_2 \leq 7 \\ 4x_1 - 5x_2 \leq 5 \end{array}\right\}, \tag{4.33}$$

$$x_1, x_2 \text{ - целое.} \tag{4.34}$$

Решим задачу (4.32, 4.33). Получим

$$\overline{x}_0 = (\frac{40}{9}, \frac{23}{9});\ \xi(G_0) = f(x_0) = -7,$$

$$G_1^{(1)} = \{x_1 \le 4\},$$
$$G_2^{(1)} = \{x_1 \ge 5\}.$$

Решив эти задачи, получим:

$$x_1 = 4;\ x_2 = 2\frac{8}{11};\ \xi(G_1^{(1)}) = \left]-6\frac{8}{11}\right[= -6;\ G_2^{(1)} = O;\ \xi(G_2) = +\infty.$$

$\Rightarrow G_1^{(1)}$ - перспективное множество. Разбив его на два подмножества по x_2, получаем:

$$G_{1,1}^{(1)} : (x_2 \le 2)\ \&\ (1), (2) \ldots$$

$$G_{1,2}^{(2)} : (x_2 \ge 3)\ \&\ (1), (2) \ldots$$

Переобозначим:

$$G_{1,1}^{(1)} = G_1^{(2)};\ G_{1,2}^{(1)} = G_{(2)}^{(2)};\ G_2^{(1)} = G_{(3)}^{(2)}.$$

Решим задачу на $G_1^{(2)}$. Получим

$$\overline{x}_1^2 = (3\frac{3}{4}; 2);\ \xi G_1^{(2)} = \left]-5\frac{3}{4}\right[= -5$$

А на $G_2^{(2)}$ получим :

$$G_2^{(2)} : \overline{x}_2^2 = (2\frac{1}{2}; 3),\ \xi(G_2^{(2)}) = \left]-5\frac{1}{2}\right[= -5,$$

$$G_3^{(2)} = O;\ \xi = +\infty.$$

Берем оценку первую по порядку, т.к. одинаковые:

$G_{1,1}^{(2)}:(x_1 \le 3)$,

$G_{1,2}^{(2)}:(x_1 \ge 4)$.

Переобозначим (получим 4 подмножества).

Решив задачу на $G_1^{(3)} \to \overline{x}_1^{(3)} = (3;2);\ \xi(G_1^{(3)}) = -5$,

$G_2^{(3)} = O;\ \xi(G_2^{(3)}) = +\infty$

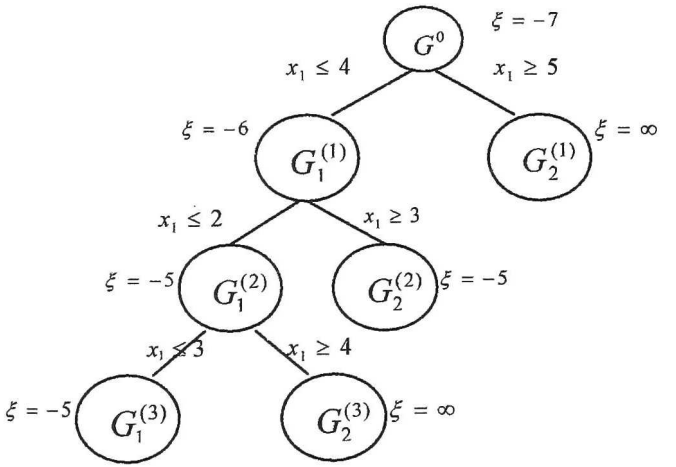
$G_3^{(3)} \to \overline{x}_3^{(3)} = (2\frac{1}{2};3);\ \xi(G_3^{(3)}) = \left]-5\frac{1}{2}\right[= -5,$

$G_4^{(3)} = O; \xi(G_4^{(3)}) = +\infty.$

Получено целочисленное решение $\overline{x}_1^{(3)} = (3,2)$. Применим условие оптимальности.

$f(3, 2) = -5 = \min\{-5, +\infty, -5, +\infty\} = -5.$

Условие оптимальности выполняется. Решение $\overline{x}_1^{(3)} = (3;2)$ является оптимальным $z^* = -5$. Изобразим дерево решений (Рис. 19).



$_1 = 3; x_2 = 2$

Рис. 19

4.7. Алгоритмы решения задач булевского программирования

Задача булевского программирования является наиболее часто встречающейся при проектировании технических средств, автоматизированных систем.

Как показано выше, любую задачу дискретного программирования, в свою очередь, можно привести к задаче булевского программирования.

Для решения задач булевского программирования используется большое число методов прямого поиска. Все они похожи на метод ветвей и границ, так как в них происходит перебор вариантов и анализ подмножеств в целях отсеивания. Большинство алгоритмов используют частичный вид булевской формы. Из-за важности этого существуют специализированные процессоры для решения таких задач.

Рассмотрим задачу линейного булевского программирования

$$L = \sum_{i=1} c_i x_i \to \max(1), \tag{4.34}$$

$$\sum_{i=1}^{n} a_{ij} x_j \le b_i, i = \overline{1,m}(2), \tag{4.35}$$

$$x_j \in \{0,1\}(3). \tag{4.36}$$

Существует 2^n вариантов решения, часть из которых допустима.

При большом n придётся проверять все m ограничений, а проверка каждого ограничения – это трудоемкая операция. Рассмотрим процедуру перебора, положенную в основу аддитивного алгоритма Балаша [8].

Так как внутри метода всегда будет сравнение с рекордом, то необходимо, чтобы первое рекордное значение было достаточно большим. Поэтому переменные исходной задачи преобразовывают в соответствии с возрастанием (для задачи на максимум) коэффициентов целевой функции:

$$c_1 \le c_2 \le c_3 \le \ldots \le c_n .$$

В этом случае, мы будем перебирать с нулевого вектора, начиная справа. Пусть существуют правила, по которым можно выстроить ограничения в таком порядке, чтобы сначала было самое жёсткое из них, а далее мягче.

Затем выбирают первый вектор списка и проверяют ограничения, если какое – то из ограничений не выполняется, то переходим к следующему вектору и т. д. Если для какого-то вектора удовлетворяются все ограничения, то

подсчитывается величина целевой функции и она объявляется рекордом. После этого процедура меняется.

Балаш предложил добавить некоторое фильтрующее ограничение. Идея его в том, что незачем проверять вектор на допустимость, если он не превосходит рекорда Z^*:

$$\sum_{j=1} c_{ij} x_j > Z^*. \tag{4.37}$$

Если ограничение (4.37) не выполняется, то переходим к следующему вектору, в противном случае – к ограничению (4.35). Если новый вектор удовлетворяет всем ограничениям, то, подставляя его в (4.34), получим новый рекорд Z^* и формируем новое фильтрующее ограничение, заменяя Z^* на Z^{**}. Если пройден весь список, то последний рекорд даёт решение. При этом мы экономим на проверке ограничений.

Пример: $z = 3x_1 - x_2 + 2x_3 + x_4 \to \max$,

$2x_1 - x_2 - 4x_3 + x_4 \le 10$,

$-3x_1 + 2x_2 + x_3 + 2x_4 \ge 1$,

$4x_1 - x_2 - 2x_3 \le 4$,
$x_j \in \{0,1\}$,

$z = -x_2 + x_4 + 2x_3 + 3x_1 \to \max$,

$-x_2 + x_4 - 4x_3 + 2x_1 \le 10$, (1)

$2x_2 + 2x_4 + x_3 - 3x_1 \ge 1$, (2)

$-x_2 + 0 \cdot x_4 - 2x_3 + 4x_1 \le 4$. (3)

Составим таблицу и получим первый допустимый вектор (0010); z=2. Составляем фильтрующее ограничение:

$(0) - x_2 + x_4 + 2x_3 + 3x_1 > 2$.

Находим новый рекордный вектор (0110); z=3; новое фильтрующее ограничение

$$(0) - x_2 + x_4 + 2x_3 + 3x_1 > 3 .$$

Наконец находим новое рекордное значение (1111); z=5. Т.к. список исчерпан, то последний рекорд – это оптимальное решение.

$$z_{\max} = 5 .$$

Эффективность процесса можно оценить количеством значков в таблице. В современных алгоритмах (например, в алгоритме Лемке-Шпильберга) порядок перехода от одного вектора к другому меняется в соответствии с анализом коэффициентов модели задачи:

1. Критерий недопустимости – запрещает двигаться к некоторым векторам, которые заведомо недопустимы.

2. Проверяются целевые функции – множеству дается оценка (подробнее см.[7,10,23]).

Таблица 31

$\bar{x}$				1	2	3	(0)	z
0	0	0	0	+	-			2
0	0	0	1	+	-			
0	0	1	0	+	+	+		
0	0	1	1	+	-		+	
0	1	0	0				-	3
0	1	0	1	+	-		+	
0	1	1	0	+	+	+	+	
0	1	1	1	+	-		+	
1	0	0	0				-	
1	0	0	1				-	
1	0	1	0				-	
1	0	1	1	+	-		+	
1	1	0	0				-	5
1	1	0	1				-	
1	1	1	0				-	
1	1	1	1	+	+	+	+	

В алгоритме Балаша используются и другие ограничения, позволяющие отсечь некоторые решения (см. [7]). Рассмотрим некоторые из них.

Рассмотрим правило, по которому можно отсеивать варианты. Пусть N - множество индексов всех двоичных векторов. Подмножество множества N, состоящее из таких элементов j, что каждому j поставлено в соответствие $x_j \in \{0,1\}$, называется частичным решением, а множество индексов обозначим V.

Переменные x_j, номера которых не принадлежат этому подмножеству, называются свободными, множество индексов свободных переменных обозначим S.

Пример: Вектор переменных $(x_1, x_2, x_3, x_4, x_5)$ N=(1, 2, 3, 4, 5), $(x_1, 0, 1, x_4, x_5)$ - частичное решение, V=(2, 3),

$S = N/V,$

S= (1, 4, 5) - номера свободных переменных.

Если частичное решение содержит k переменных, то существует 2^{n-k} различных дополнений частичного решения.

Пусть каким – то образом выбрано частичное решение, содержащее k переменных, тогда проверка 2^{n-k} его дополнений на допустимость по условию (4.35) называется зондированием.

Если до этого получен рекорд Z^* и если удаётся установить, что для какого – то частичного решения не существует допустимого дополнения, приводящего к лучшему рекорду, то говорят, что частичное решение прозондировано, т.е. что это подмножество можно отсеять.

Балаш установил, что если V определяет переменные зондированного решения, а Z^* - ранее достигнутый рекорд, то зондируемое решение x не имеет допустимого дополнения, улучшающего Z^*, если выполняется условие:

$$\sum_{j\in S}\min(a_{ij},0) > b_i - \sum_{j\in\Omega} a_{ij}x_j, \text{ при любом } i=\overline{1,m} \qquad (4.38)$$

Например, пусть задача содержит ограничения:

$$-x_1 + 2x_2 + 6x_3 + 4x_4 - x_5 \le 0,$$

$$x_1 = 1,\ x_2 = 1,\ x_4 = 0,$$

$$\Omega = (1,3,4),\ N = (1,2,3,4,5),\ S = \{2,5\}.$$

Проверим наличие допустимых дополнений для выбранного частичного решения.

$$\sum_{j\in S}\min(a_{ij},0) = -1;\ \sum_{j\in v} a_{1j}x_j = 5;\ b_1 - \sum_{j\in v} a_{1j}x_j = -5;\ -1 > -5.$$

Рассмотрим другое правило отсеивания вариантов. Если для некоторой свободной переменной x_k имеет место соотношение

$$\sum_{j\in S}\min(a_{ij},0) + |a_{ik}| > b_i - \sum_{j\in v} a_{1j}x_j,\ \forall i=\overline{1,m},$$

$$x_k = \begin{cases} 0, & если - a_{ik} \geq 0, \\ 1, & если - a_{ik} < 0. \end{cases} \quad (4.39$$

Кроме этих двух правил можно использовать, так называемый, признак составных ограничений, позволяющий установить факт наличия или отсутствия допустимых дополнений некоторого частичного решения, путём анализа составного ограничения.

В общем случае это комбинация всех строк ограничений.

Например, используется суммирование левых и правых частей ограничений.

Пример:

Пусть задача состоит из такого решения $x_1, x_2, \ldots, x_5$, которое имеет ограничения:

$$- x_1 - x_2 - x_3 \leq - 1,$$

$$2x_2 - 2x_4 + 2x_5 \leq 0,$$

$$x_j = \{0,1\}.$$

Максимизируется функция

$$Z = 2x_1 + x_2 + x_3 \rightarrow \max.$$

Пусть при некотором частичном решении рекорд составлял $Z=2$. Определить, имеет ли частичное решение $x_1=0$ допустимое дополнение, улучшающее достигнутый рекорд.

Прибавим к ограничениям фильтрующее ограничение, то есть

$2x_1 + x_2 + x_3 \geq 3$, получим новую систему ограничений:

$$- x_1 - x_2 - x_3 \leq -1,$$

$$2x_2 - 2x_4 + 2x_5 \leq 0,$$

$$- 2x_1 - x_2 - x_3 \leq -3.$$

Сформируем составное ограничение

$$- 3x_1 - 2x_3 - 2x_4 + 2x_5 \leq -4.$$

)именим к нему признак (4.38)

$$\sum_{j\in S}\min(a_{ij},0)=-4,\quad \sum_{j\in\Omega}a_{ij},x_j=0,\quad -4=-4.$$

Получаем - 4= - 4, следовательно, ограничение выполняется, то есть при) не существует дополнения, улучшающего рекорд Z=2.

На основе полученных правил запишем алгоритм.

Пуст ли основной список задач? Если «да», то вычисления окончены и переходим к п.5. Если «нет» - присвоим счетчику числа задач основного списка α значение α := α -1, т.е. вычеркиваем одну задачу. Счётчику чис-ла итераций $k := k+1$ (в начале, очевидно, $k = 1$).

Существует ли допустимое дополнение при выбранном частичном реше-нии, улучшающее значение рекорда Z^*? Проверим условие (4.38).

Если «да», то определяем эти значения по формуле (4.39) и переходим к п.3. Если «нет» - положим $Z^{(k+1)} = Z^k$ и переходим к п.1.

Содержат ли допустимое дополнение частичного решения и зондируемое решение n переменных?

Если «да» - запоминаем решение, значение рекорда Z^* и переходим к п.1.

Если «нет» - переходим к п.4.

Выбираем любую свободную переменную x_i, не входящую в число до-пустимых переменных. Вводим две задачи в основной список (в одной x_i=0, в другой x_i=1). Положим $Z^k = Z^{(k-1)}$ и переходим к п.1.

Печать результата решения: рекорд и значения переменных либо «задача ения не имеет». Конец алгоритма.

5. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ (ДП)

Это особый метод, который специально приспособлен для оптимизации динамических задач, в которых операция состоит из элементов, сильно влияющих друг на друга. ДП связано с именем Ричарда Беллмана, который сформулировал принцип оптимальности Беллмана. Он позволяет существенно сократить перебор решений в многоэтапных нелинейных задачах.

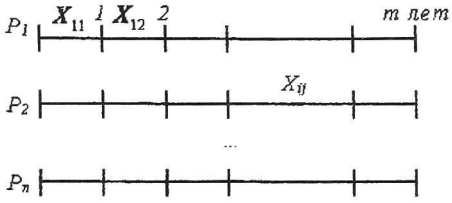


Рис. 20

Рассмотрим экономическую задачу распределения ресурсов: пусть есть начальный капитал k_0. Его можно потратить на несколько предприятий $P_1, P_2, \ldots, P_n$ (Рис. 20). X_{ij} - количество средств вкладываемых в i - м году в j - е предприятие. В результате получается эффект:

$$W_{ij} = f\left(X_{ij}\right).$$

В общем случае это нелинейная функция. Необходимо распределить начальный капитал (ресурс) так, чтобы суммарный эффект от всех предприятий за все года был максимальным:

$$W = \sum f\left(X_{ij}\right) \to \max,$$

$$\sum X_{ij} \le k_0.$$

Так как функция W нелинейная, то получаем задачу нелинейного программирования при очень большом числе переменных (см. главу 6). Решать её сложно, кроме того, часто X_{ij} - дискретные значения. Возникает вопрос : нельзя ли решить задачу последовательно, т. е. найти оптимальное вложение для первого года, второго и т. д., т. е. провести последовательную оптимизацию? В большинстве задач так делать нельзя, т. к. решение, принимаемое на одном шаге, оказывает влияние на последующие шаги. Рассмотрим простой пример.

Проанализируем процесс забега стайеров на 800 метров (рис. 21).

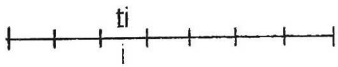


Рис. 21

Каждый бегун имеет запас энергии x_0, который он тратит на каждые 100 метров - величина затрат энергии x_i :

t_i – время бега стайера на отдельном участке i;

T – общее время забега на 800м.

$$T = \sum_{i=1}^{8} t_i \to \min .$$

Если уменьшать каждое t_i отдельно, то за 800 м время стайера будет хуже, так как у него не будет сил продолжать бег.

Значит нужно бежать так, чтобы оставались силы на всю дистанцию.

Очевидно, каждый бегун решает задачу: $\Sigma t_i(x_i) \to \min$; при условии $\Sigma x_i \le x_0$. Казалось бы, чтобы сумма времен t_i была минимальной нужно минимизировать каждое t_i, т.е. бежать как можно быстрее первую стометровку, затем как можно быстрее вторую и т.д. Читатель, вспомнив свой опыт бега, конечно, понял, что такой способ оптимизации не верен. Оказывается, оптимальной величиной x_i и, соответственно t_i, будет такая, которая обеспечит минимальное общее время бега за все 800 метров. Т.о. мы сформулировали принцип «действуй с прицелом на будущее». Другими словами оптимальность «в малом» необходимо понимать через оптимальность «в большом». Это чрезвычайно важный принцип системного подхода к оптимизации систем.

5.1 Принцип оптимальности Р.Беллмана

Метод ДП является наиболее общим методом решения задач оптимального управления. Он применим как для задач с линейной ЦФ, так и с нелинейной , а также в случае, когда управляемые переменные целые числа, при этом сама ЦФ может быть задана таблицей, что наиболее распространено в практических задачах.

Процессы называют динамическими, если результаты на одном участке процесса влияют на другие шаги.

Рассмотрим принцип оптимального управления Р. Беллмана. Он связан с проблемой оптимизации сложной системы, состоящей из многих взаимосвязанных элементов. Элементами могут быть экономические еди-

ницы, которые входят в единую (более общую) систему; узлы сложной технической системы; отдельные участки производства, строительства, боевых операций и т.д.

Возникает вопрос: как нужно управлять отдельными элементами системы, чтобы показатель эффективности всей системы был максимальным?

Выше мы на интуитивном уровне показали, что для оптимизации в целом недостаточно оптимизировать каждый элемент отдельно - это приводит к неверному результату.

Беллман впервые сформулировал принцип оптимальности для такой задачи. То есть, оптимизируя отдельный шаг, необходимо задумываться о его последствиях, приводящих к общему результату. Для решения подобных задач был разработан метод ДП, основывающийся на уравнении Беллмана, которое математически выражает его принцип.

Назовем состоянием системы S один или несколько параметров системы. Например, деньги на лицевом счете предприятия. Обозначим управление на i-м шаге U_i – это некоторое воздействие, которое испытывает система и изменяет свое состояние S.

Если перед i-м шагом состояние системы S и мы принимаем управление U_i, то за i-й шаг мы можем получить некоторый выигрыш, который обозначается $\omega_i(S_i, U_i)$, при этом состояние S переходит в S':

$S \rightarrow S' = \varphi_i(S, U_i)$.

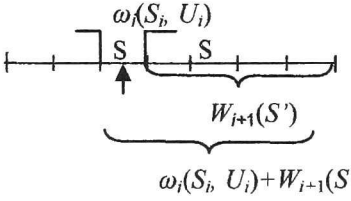


Рис. 22

Считается, что функции $\omega_i(S_i, U_i)$, и $\varphi_i(S, U_i)$ известны.

Беллман ввел понятие условного оптимального выигрыша $W_i(S)$. Эта функция показывает оптимальный выигрыш (наилучший результат), полученный за все шаги от i-го и до конца, если i-й шаг начинается с состояния S. Тогда согласно принципу оптимальности Беллмана, принимая решение на i шаге, мы должны выбрать U_i так, чтобы выигрыш был максимальным от i-шага и до конца.

Принцип оптимальности Беллмана ставит вопрос о том, что такое оптимальность отдельного элемента системы с точки зрения оптимальности всей системы. Принимая решение на отдельном этапе, мы должны выбирать

правление на этом этапе с прицелом на будущее, т. к. нас интересует результат в целом за все шаги.

Любой процесс имеет где-то окончание, т. е. говорят, что он имеет «горизонт планирования». Тогда последний этап «не имеет будущего». Вот именно его можно оптимизировать только из условий данного этапа. После этого приступают к оптимизации (m-1)-го этапа. При этом мы задаём состояние, с которого начинается (m-1)-й шаг (условие). Поэтому функцию $W_i(S)$ называют условным оптимальным выигрышем. Таким образом, процесс оптимизации по методу ДП разворачивается сначала от конца к началу, а затем от начала к концу. В различных задачах может быть известно либо начальное состояние, либо конечное, либо то и другое. Принцип Беллмана нашёл практическое применение в так называемом методе программно-целевого планирования (любое действие планируется как элемент некоторого проекта).

5.2. Решение графовых задач на основе принципа Беллмана

Задача о наборе высоты и скорости летательного аппарата

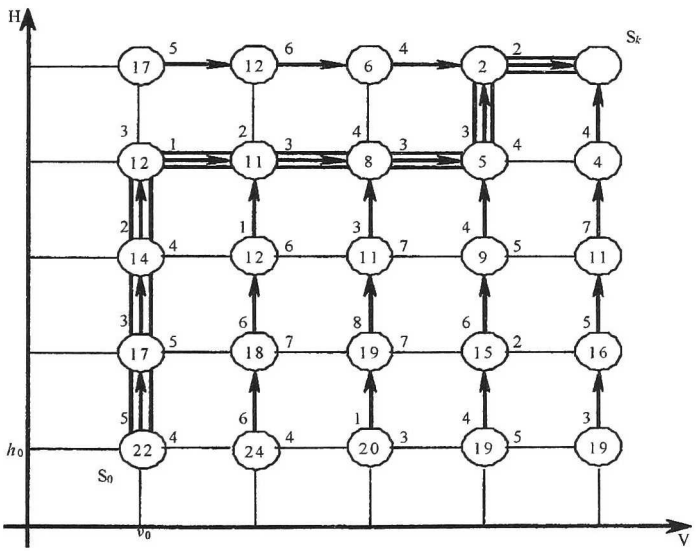


Рис. 23

Летательный аппарат находится на высоте h_0 и летит со скоростью v_0. Необходимо перевести его на высоту h_1 со скоростью v_1. Причём $h_0>h_1$, $v_0>v_1$. Разобьём участок от h_0 до h_1 на n частей:

$$\Delta h = \frac{h_1 - h_0}{n},$$

$$\Delta v = \frac{v_1 - v_0}{m}.$$

Известен расход горючего при переводе системы на Δh при v=const и на Δv при h=const. Таким образом, из каждого состояния есть лишь два управления.

Начиная с конца помечаем все узлы (состояния) величинами условных (для данного узла) оптимальных расходов горючего от этого узла и до конца, а стрелками условные оптимальные управления. Указанные действия в упрощенном виде демонстрируют рассмотренную процедуру решения на основе уравнения Беллмана. Дойдя от конечного состояния до начального и получив 22, мы получим минимальную величину потерь горючего. Идя по стрелкам от начального состояния до конечного, мы получаем безусловные оптимальные управления (показаны двойной линией).

Видно, что любая задача, сводящаяся к поиску минимального пути в графе, решается методом динамического программирования.

5.3. Функциональное уравнение Беллмана

Назовём состоянием системы вектор координат: $S = (\xi_1, \xi_2, ..., \xi_L)$. В некоторых задачах состояние - одна величина. Тогда работу системы можно представить как движение в фазовом пространстве - пространстве состояний. Назовём шаговым управлением - управление на i-м шаге. Рассмотрим процесс управления системой за m шагов. Функция $\omega_i(S, U_i)$ называется выигрышем на i-м шаге. Здесь S-состояние перед i-м шагом, а U_i - управление на i-м шаге.

Величина $\omega_i(S, U_i)$ должна быть известна до начала динамического программирования. Если состояние перед i-м шагом было S и мы приняли какое-то управление U_i, то система перейдёт в новое состояние $S' = \varphi_i(S, U_i)$. Эта функция должна быть так же известна. Если эти функции не заданы, то их надо сформулировать. Введём функцию $W_i(S)$ - условный оптимальный выигрыш. Это выигрыш на всех этапах от i до конца, если i-й шаг начинается с состояния S.

Рассмотрим m шагов. Пусть с $(i+1)$-го шага мы системой управляем оптимально, тогда величина выигрыша будет такая - $W_{i+1}(S')$. Применим на i-м шаге произвольное управление U_i, тогда $\tilde{W}_i(S)$ - неоптимальный выигрыш, т. к. на i-м шаге мы применяем неоптимальное управление U_i. Чтобы от i-го шага и до конца получить оптимальный выигрыш, нужно изменять U_i так, чтобы

$$W_i(S) = \max_{U_i}\{\omega_i(S,U_i) + W_{i+1}(S')\};\quad S' = \varphi_i(S,U_i),$$

$$\underset{\text{неизвестно}}{W_i(S)} = \max_{U_i}\left\{\underset{\text{известно}}{\omega_i(S,U_i)} + \underset{\text{неизвестно}}{W_{i+1}\left[\varphi_i(S,U_i)\right]}\right\}.$$

Это функциональное уравнение Беллмана. Для использования уравнения еллмана начинают с конца:

1. $i=m$,

$$W_m(S) = \max_{U_m}\{\omega_m(S,U_m)\};$$

2. $i=m-1$,

$$W_{m-1}(S) = \max_{U_{m-1}}\{\omega_{m-1}(S,U_{m-1}) + W_m[\varphi_{m-1}(S,U_{m-1})]\}.$$

И так, идя от конца к началу, мы олучаем последовательно:

$$W_m(S),\ W_{m-1}(S),\ldots,\ W_1(S),$$

$$U_m(S),\ U_{m-1}(S),\ldots,\ U_1(S).$$

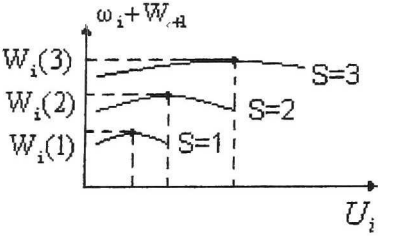


Рис. 24

Придя в начальное состояние $W_1(S)$, ы можем подставить $S=S_0$ и $W_1(S_0)=W_{\max}$ это безусловный выигрыш.

Теперь необходимо получить безусовные оптимальные уравнения, идя от ачала к концу по цепочке:

$$S=S_0 \to U_1(S_0) = U_1^* \to$$
$$\varphi(S_0,U_1^*) = S_1^* \to U_2(S_1^*) = U_2^* \to$$
$$\varphi(S_1^*,U_2^*)\ldots$$

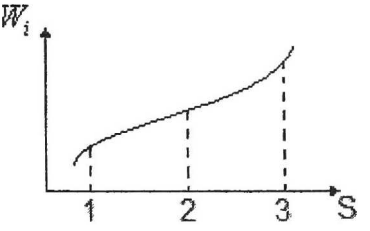


Рис. 25

Итак, в результате мы получаем оптимальное решение:

$$U_1^*, U_2^*, \ldots, U_m^*; W_{\max}.$$

5.4. Задачи распределения ресурсов

5.4.1. Классическая задача распределения ресурсов

Распределение ресурсов - это едва ли не самая распространённая операция. Под ресурсом в общем случае понимают физическую или абстрактную величину, которую система использует для производства полезного продукта. Например: горючее, деньги, время, объём склада. Как правило - ресурс ограничен, поэтому встаёт задача так распределить ресурс между отдельными элементами системы, чтобы суммарный эффект был максимальным. Рассмотрим классическую задачу распределения ресурсов.

Имеется начальное количество ресурсов k_0, которое необходимо распределить между двумя отраслями. Каждая отрасль работает в течении m лет. Если в первую отрасль в i-й год вкладываются средства X_i, то доход $f(X_i)$, если же во вторую вкладываются Y_i, тогда доход $g(Y_i)$. Средства тратятся, принося доход, а новых средств не поступает и полученный доход не вкладывается.

Нас интересует суммарный доход:

$$W = \sum_{i=1}^{m} f(X_i) + g(Y_i).$$

Суммарный выигрыш равен сумме выигрышей на каждом шаге. Состоянием системы является количество средств перед i-м шагом. Так как новых средств не поступает, то ресурсы «тают».

Управление Y_i может быть записано как $Y_i=k-X_i$. После i-го шага в первой отрасли остаются средства $\varphi(X_i)$, а во второй $\psi(Y_i)=\psi(k-X_i)$. Эти функции называются функциями траты. Мы можем составить уравнение Беллмана. В этой задаче на i-м шаге одно управление X_i и одно состояние k.

$$W_i(k) = \max_{X_i}\left\{f(X_i) + g(k-X_i) + W_{i+1}\left[\varphi(X_i) + \psi(k-X_i)\right]\right\};$$

$$i = m: \quad W_m(k) = \max_{X_m}\left\{f(X_m) + g(k-X_m)\right\} \text{ и т. д.};$$

$$W_1(k), k = k_0, W_1(k_0) = W_{\max}; \; X_1(k_0) = X_1^*; Y_1^* = k_0 - X_1^*.$$

Используя функции траты, получим количество средств после i-го шага:

$$\varphi(X_1^*) + \psi(Y_1^*) = k_1^*; \; X_2(k_1^*) = X_2^* \text{ и т.д.}$$

Задача о распределении ресурсов допускает геометрическую интерпретацию:

$$X_1 + Y_1 = k_0 .$$

Распределение на первом шаге - указание точки на гипотенузе. После этого средства тратятся. Распределение средств - движение внутрь треугольника. Рассмотрим частные случаи задач о распределении ресурсов.

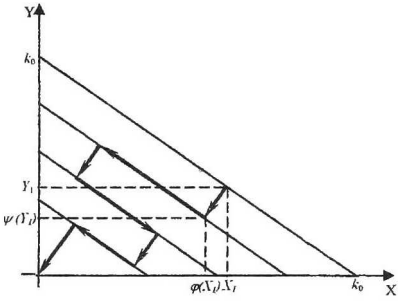


Рис. 26

5.4.2. Неоднородные этапы и распределение ресурсов по n отраслям

Распределение по неоднородным этапам.

Выше мы считали, что все функции одинаковы на всех этапах. Во многих задачах функции меняются от этапа к этапу: $f_i(X_i)$, $g_i(Y_i)$; $\varphi_i(X_i)$ $\psi_i(Y_i)$.

Покажем, что процедура динамического программирования принципиально не меняется. Запишем уравнение Беллмана:

$$W_i(k) = \max_{X_i} \left\{ f_i(X_i) + g_i(k - X_i) + W_{i+1}\left[\varphi_i(X_i) + \psi_i(k - X_i)\right] \right\}.$$

Видно, что при решении достаточно на каждом этапе всего лишь подставлять разные функции.

Распределение ресурсов между тремя и более отраслями.

В этом случае на каждом шаге будет уже n управлений, но одно из них может быть выражено как: $X_i^n = k - \sum_{j=1}^{n-1} X_i^j$. В этом случае в правой части уравнения Беллмана будет две или более переменных, по которым ищется максимум, и задача существенно усложняется.

5.4.3. Распределение ресурсов с резервированием

В такой модели если средства распределяются между двумя отраслями, то какое-то количество средств можно оставить до последующего распределения. В этом случае задача имеет смысл даже для одной отрасли.

Начальное количество средств разделяется на первом этапе на X_1 и на k - X_1 (резерв), на втором этапе подлежат распределению оставшиеся средства и из резерва. Такую задачу можно представить с одной реальной отраслью, а другой - фиктивной (не приносящей доход и не расходующей средства). На рисунке 27 показаны графики распределения ресурсов с резервированием и в дополнение к этому с нулевыми функциями трат.

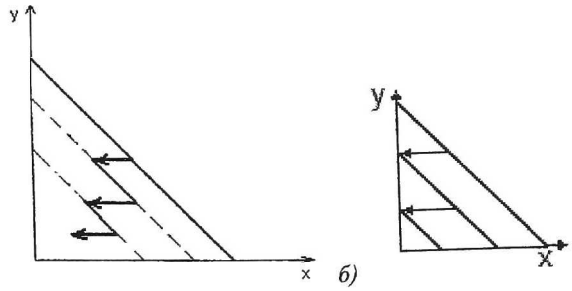


Рис. 27

Решение такой задачи сводится к классической, задав функции дохода и трат в виде:

$f(X), g(Y) = 0$ - функции дохода; $\varphi(X), \psi(Y) = 0$ - функции трат.

Задача с резервированием в одной отрасли при нулевых функциях траты.

Все функции траты $\varphi(x_i) = 0$.

В этом случае задача сводится к более простой:

$$\varphi(x_i) = 0,$$

$$W = \sum_{i=1}^{m} f_i(x_i) \to \max,$$

$$\sum x_i = k_0.$$

Рассмотрим более частный случай - все функции одинаковые на всех шагах.

$$f_i(x) = f(x), \forall i.$$

Пусть эти функции неубывающие, тогда недоиспользование средств невыгодно и в ограничении можно поставить равенство:

$$W = \sum_{i=1}^{m} f(x_i) \to \max, \qquad (5.1)$$

$$\sum x_i \le k_0. \qquad (5.2)$$

Эта задача имеет теоретическое решение:

1 Если функция $f(x)$ неубывающая и выпуклая вверх, то оптимальным распределением является равномерное распределение: $x_i=k_0/m$.

2 Если функция $f(x)$ неубывающая и выпуклая вниз, то оптимальным распределением является такое - все распределение в один этап (элемент) и ничего в другие: $x_r=k_0$, $x_i=0$ для всех i, кроме $i=r$.

Покажем это на графических примерах (рис. 27, 28).

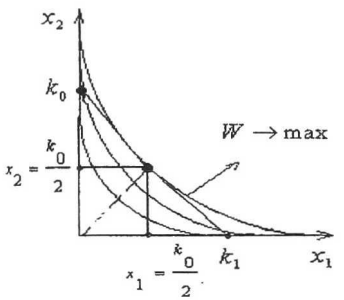


Рис. 28

1. Пусть $f(x)$ – выпуклая вверх, например, $f(x)=\sqrt{x}$, $W=\sqrt{x_1}+\sqrt{x_2}\to\max$, $x_1+x_2=k_0$.

Тогда линии постоянного уровня W – это дуги линий астроид, симметричных относительно биссектрисы координат. Поэтому последнее касание астроиды гипотенузы треугольника при $W \to max$ будет в точке $x_i = k_0/2$. Очевидно, для m шагов будет $x_i = k_0/m$.

Рис. 29

2. Пусть $f(x)$ - выпуклая вниз, например, $f(x)=x^2$, $W=x_1^2+x_2^2\to\max$, $x_1+x_2=k_1$.

Из рисунка видно, что в этом случае решение будет лежать на оси координат, т.е. $x_r = k_0$.

5.4.4. Распределение ресурсов с «вложением доходов»

В классической задаче считается, что полученный доход на i-м шаге в производство не вкладывается, т. е. он отчисляется и подсчитывается как эффект. Во многих задачах полученный эффект можно использовать как ресурс для следующего шага, объединяя его с оставшимся ресурсом. Если ресурс не деньги, то средства нужно привести к единому эквиваленту с оставшимися средствами. Такая модель является развитием классической модели. Так как оставшиеся средства и доход объединяются, то можно ввести единую интегральную функцию - функцию изменения средств $F(X_i)$ - количество оставшихся средств плюс доход после i-го шага, если вложили X_i.

I. $F(X_i)$,

II. $G(Y_i)=G(k-X_i)$.

Где k -количество средств перед i-м шагом.

Выигрыш на i-м шаге зависит от того, как мы подсчитываем доход (эффект) от управления всеми ресурсами. Поставим задачу: максимизировать доход в конце m-го шага. Тогда на всех шагах $i=\overline{1,m-1}$, доход = 0, $W_i=0$. На m-м шаге выигрыш $W_m=F_m(X_m)+G_m(k-X_m)$. Подставив эти выражения в уравнение Беллмана, мы программируем задачу от начала к концу, если имеется начальное количество средств k_0. Здесь функция траты: $k'=F_i(X_i)+G_i(k-X_i)$.

Рассмотрим частный случай: когда F и G неубывающие функции. В этом случае, чем больше значение доход + средства получается в конце i-го шага, тем лучшим условием это будет для проведения (i+1)-го шага. Поэтому можно не заботиться о следующих шагах, достаточно обеспечить максимум на каждом шаге.

Таким образом, процедура оптимизации возможна в одном направлении от начала к концу, т. е. задача динамического программирования вырождается в задачу последовательной оптимизации.

$$\max_{X_1}\left\{F_1(X_i)+G_1(k_0-X_1)\right\}=k_1^*,$$

$$\max_{X_2}\left\{F_2(X_2)+G_2(k_1^*-X_2)\right\}=k_2^*,$$

...

$$\max_{X_m}\left\{F_m(X_m)+G_m(k_{m-1}^*-X_m)\right\}=k_m^*,\quad W_{\max}=k_m^*.$$

Рассмотрим задачу распределения ресурсов с вложением доходов в производство и отчислением. Это наиболее общий случай. Разделим функции дохода и функции траты:

$f(X_i), g(Yi),$
$\varphi(X_i), \psi(Y_i),$

и максимизируем суммарный отчисленный доход + оставшиеся средства после m-го шага. Введём функцию отчисления $r_i(D_i)$; D-доход. Тогда выигрыш на каждом шаге:

$$
\begin{aligned}
&W_i = r_i\left[f(X_i)+g(k-X_i)\right], \quad i=\overline{1,m-1},\\
&W_m = r_m\left[f(X_m)+g(k-X_m)\right]+\varphi(X_m)+\psi(k-X_m),\\
&W=\sum_{i=1}^{m} W_i \to \max,\\
&k'=\varphi(X_i)+\psi(k-X_i)+f(X_i)+g(k-Xi)-r_i\left[f(X_i)+g(k-X_i)\right].
\end{aligned}
\tag{5.3}
$$

Уравнение Беллмана для i-го шага будет выглядеть так:

$$
W_i(k)=\max_{X_i}\left\{r_i\left[f(X_i)+g(k-X_i)\right]+W_{i+1}\left[\varphi(X_i)+\psi(k-X_i)+f(X_i)+\right.\right.
$$

$$
\left.\left.+g(k-X_i)-r_i\left[f(X_i)+g(k-X_i)\right]\right]\right\},\ i=\overline{1,m-1},
$$

для $i=m$ надо учесть (5.3).

Если $r_i=1$, то мы получаем классическую задачу.

5.5. Расширение модели задач динамического программирования

Учёт предыстории процесса

Мы считали, что все функции, как выигрыша, так и траты, зависят от состояния перед i-м шагом, т. е. не зависят от более ранних состояний. Такие процессы называются процессами без памяти. Но иногда при рассмотрении некоторых процессов требуется помнить всю историю происходящего (например, в задачах ремонта и замены оборудования). Такая задача более сложна. Введём расширенное состояние:

$$
S=\left(S,S_{i-1},S_{i-2},\ldots,S_{i-L}\right),
$$

S_{i-L} - состояние за L шагов до i-го. Тогда, считая аргументы векторами, функции не меняются $\omega_i(S,U_i), \varphi_i(S,U_i)$. Но задача становится сложней в вычислительном аспекте. Пусть S имеет k координат и предыстория распространяется на L шагов, тогда в результате число переменных будет $k\times L$. Вот почему подобные задачи практически можно решать если $k\times L\le 3$.

ДП для задач со многими ограничениями

Пусть в задаче распределения ресурсов есть два ограничения (два ресурса).

$$\left.\begin{aligned} W &= \sum_{i=1}^{m} f(X_i) \to \max, \\ \sum_{i=1}^{m} a_{1i} X_i &\le b_1, \\ \sum_{i=1}^{m} a_{2i} X_i &\le b_2. \end{aligned}\right\}$$

Введем два параметра состояния K_1 и K_2. Тогда функции условных оптимальных выигрышей и условных оптимальных управлений будут иметь по два аргумента $W(K_1, K_2)$; $x_i(K_1, K_2)$. Уравнение Беллмана для решения такой задачи имеет вид

$$W_i(K_1, K_2) = \max_{X_i} \left\{ f_i(X_i) + W_{i+1}\left[K_1 - a_{1i} X_i; K_2 - a_{2i} X_i\right] \right\}.$$

При этом величина X_i на каждом этапе варьируется в пределах от нуля до величины d_i :

$d_i = \min(K_1/a_{1i}; K_2/a_{2i})$.

Видно, что для двух ограничений, если каждая из величин K_1, K_2 может принимать 10^2 значений, то функцию $W_i(K_1, K_2)$ приходится табулировать в 10^4 точках. Таким образом, наиболее серьезным препятствием для практического применения ДП является число параметров задачи, что заставило Р.Беллмана заявить о так называемом «проклятии многомерности». Некоторые способы понижения размерности смотри в [10].

Задача с мультипликативным критерием

До сих пор мы считали, что суммарный выигрыш равен сумме выигрышей, полученных на i-х шагах. Но есть задачи, где общий критерий равен произведению критериальных величин на каждом шаге:

$$W = \prod_{i=1}^{m} W_i .$$

В этом случае так же можно применить уравнение Беллмана, но вместо этого можно взять функцию $W' = \ln W$. Оптимальные решения будут одинаковы ввиду монотонности логарифмической функции. Но можно при выводе уравнения Беллмана учесть, что

$$W_i(S) = \max_{U_i} \left\{ W_i \times W_{i+1}(S') \right\},$$

$$W = F(W_1, W_2, \ldots, W_n) = F(W_1) \otimes F(W_2) \otimes \ldots \otimes F(W_n).$$

Т.о. метод ДП применим в случае, когда аргументы в критериальной функции разделяются (смотри вышеприведенное выражение)

Пример: устройство состоит из n узлов. Имеется некоторый ресурс k_0, который может использоваться для повышения надёжности каждого узла. Необходимо так распределить ресурс, чтобы суммарная надёжность была максимальной. Получаем задачу

$$Q = \prod_{i=1}^{m} q_i(X_i) \to \max,$$

где $q(X_i)$ - надёжность каждого узла, $\sum X_i \le k_0$.

Операции не связанные со временем

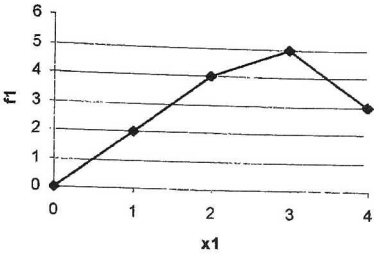
Во многих задачах распределение ресурсов не связано с временными шагами. Ресурс распределяется по объектам. Например, если рассмотреть распределение ресурсов между n объектами, когда на каждый объект задана функция выигрыша, то такая задача эквивалентна рассмотренной нами задаче о распределении ресурсов с резервированием в одной отрасли по n шагам. Это наиболее распространенная задача распределения ресурсов, поэтому приведем ее полное решение методом ДП.

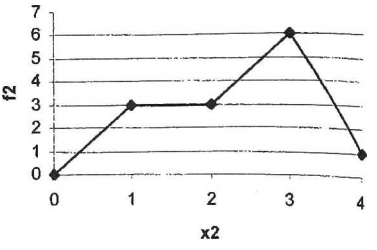
5.6. Пример решения задачи распределения ресурсов

Рассмотрим пример. Завод получил 4 новых станка, которые необходимо распределить между тремя цехами. Известна дополнительная прибыль, которая получится в i-м цехе, если туда поставят x_i станков – $f(x_i)$.

Состоянием является оставшееся (текущее) количество станков (k). Пусть мы распределяем оставшиеся k станков. Начинаем с третьего (последнего) цеха:

$$W_3(k) = \max_{x_3 \le k} \{\underbrace{f_3(x_3)}_{W_3}\}.$$





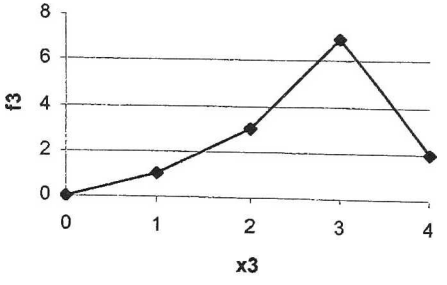


Рис. 30

Таблица 32

k	X_3	$\widetilde{W}_3$	W_3
0	0	0	0
1	0 **1**	0 1	1
2	0 1 **2**	0 1 3	3
3	0 1 2 **3**	0 1 3 7	7
4	0 1 2 **3** 4	0 1 3 7 2	7

Таблица 33

k	$x_3^{опт}$	W_3
0	0	0
1	1	1
2	2	3
3	3	7
4	3	7

Переходим ко второму шагу:

$$W_2(k) = \max_{x_2 \le k}\{f_2(x_2) + W_3(k - x_2)\}.$$

Таблица 34

k	x_2	$\tilde{W}_2$	W_2
0	0	0	0
1	0	1	3
	1	3	
2	0	3	4
	1	4	
	2	3	
3	**0**	7	7
	1	6	
	2	4	
	3	6	
4	0	7	10
	1	10	
	2	6	
	3	7	
	4	1	

Таблица 35

k	$x_2^{опт}$	W_2
0	0	0
1	1	3
2	1	4
3	0	7
4	1	10

Переходим к первому шагу:

$$W_1(k) = \max_{x_1 \le 4}\{f_1(x_1) + W_2(k - x_1)\}.$$

Таблица 36

k	x_1	$\tilde{W}_1$	W_1
4	**0**	10	10
	1	9	
	2	8	
	3	8	
	4	3	

Так как начальное состояние известно, то для первого шага не нужно перебирать оставшееся число станков, и берем k=4.

Ответ:

x_1^*=0, k_1^*=4-0=4 (подставляем k_1^* в табл. 35),

x_2^*=1, k_2^*=4-1=3 (подставляем k_2^* в табл. 33),

x_3^*=3, k_3^*=4-3=1.

5.7. Эффективность динамического программирования

Сравним по числу необходимых операций динамическое программирование с простым перебором всех вариантов для задачи

$$W = \sum_i^n f(x_i) -> \max,$$

$$\sum_i^n a_i x_i < b.$$

Для упрощения расчёта примем $a_1=a_2=\ldots=a_n=1$.

При простом переборе число возможных вариантов решений (при условии целочисленности всех переменных) равно числу способов, которыми можно разместить b одинаковых шаров внутри n урн.

Оно составляет

$$C^{b}_{n+b-1}=\frac{(n+b-1)!}{b!(n-1)!}.$$

При n=5, b=20 имеем $C^{4}_{24}=10626$.

Оценим число операций, требуемых для решения этой задачи методом динамического программирования.

Для вычисления $Wi(k)$ при фиксированном k необходимо провести k+1 вычислений. Поэтому для заполнения одной таблицы $Wi(k)$ необходимо проделать

$$\sum_{k=0}^{b}(k+1)=\frac{(b+1)(b+2)}{2}$$ операций.

Следовательно, для вычисления всех функций $W_1(k),\ldots,W_{n-1}(k)$ необходимо $(n-1)\frac{(b+1)(b+2)}{2!}$ операций. С учётом вычисления функции $W_n(b)$ общее число операций $N_\Sigma=\frac{(n-1)(b+1)(b+2)}{2!}+(b+1)=\frac{(b+1)[(n-1)(n+2)+2]}{2!}$.

При n=5, b=20 имеем N_Σ=945, что в 11 раз меньше, чем при простом переборе.

6. НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

6.1. Особенности задач нелинейного программирования

Нелинейное программирование (НЛП) - это такая задача математического программирования, в которой либо целевая функция, либо функции ограничений, либо они вместе представляют собой нелинейные функции. Термин НЛП применяется только тогда, когда все переменные X-действительные числа. Решение задач НЛП намного сложнее решения задач ЛП, при прочих равных условиях. Рассмотрим особенности задач НЛП, а именно то, где может находиться оптимальное решение. Для этого будем рассматривать целевую функцию двух переменных. В НЛП используют геометрический образ целевой функции как некоторой поверхности, определённой на множестве управляемых переменных. Вид этой поверхности можно представить, изображая в области переменных так называемые линии уровня (рисунок 31)

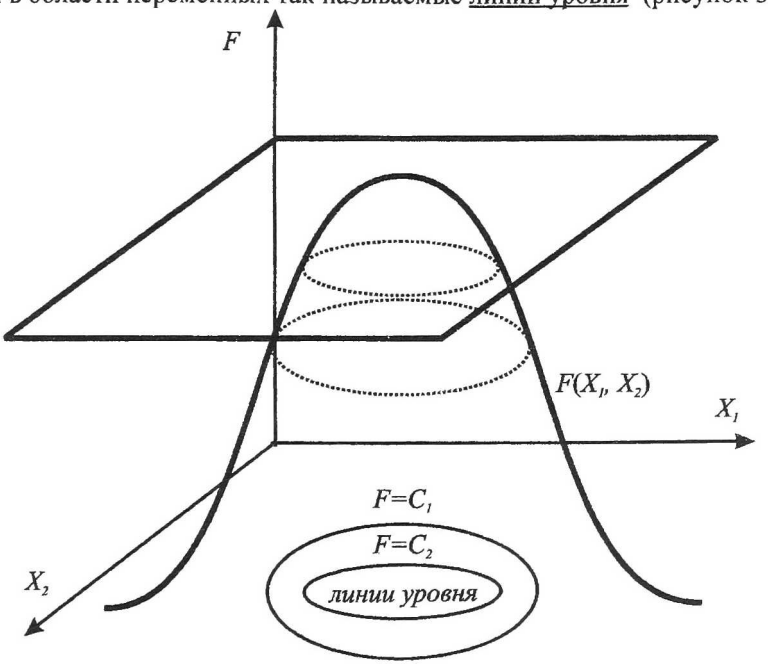


Рис. 31

Особенности задач НЛП проиллюстрируем графически на примере задачи:

$$\begin{cases} F = (x_1 - 1/2)^2 + (x_2 - 1)^2 \to \min, \\ x_1 x_2 \le a, \\ x_1 + x_2 \ge b, \\ 0 \le x_1 \le 3; 0 \le x_2 \le 3. \end{cases}$$

Изобразим графически область допустимых решений (ОДР) и найдем решение для различных значений a и b. Очевидно, что оптимальное решение будет находиться в точке, где линия уровня имеет самую малую величину в области допустимых решений (ОДР). Это можно представить как «последнее касание» линии уровня ОДР по мере уменьшения этого уровня. Линии уровня в этой задаче представляют собой концентрические окружности, а $F \to \min$ соответствует уменьшению их радиуса.

1) a=2, b=2 (рис. 32а);

2) a=1/2, b=2 (рис. 32б).

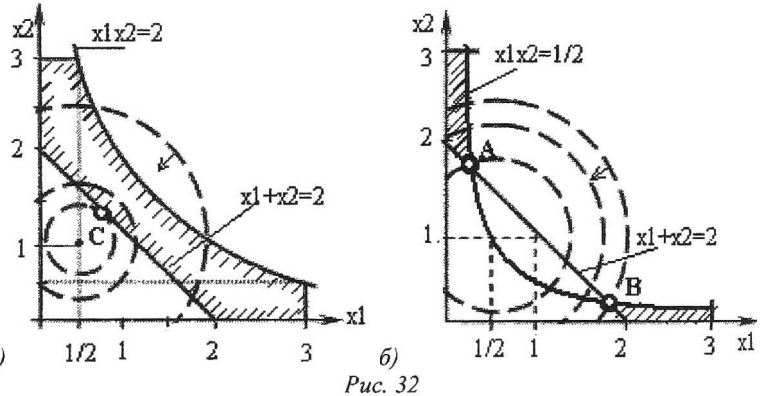


Рис. 32

Решение задачи на рис. 32а находится в точке C, где линия наименьшего уровня целевой функции касается прямой $x_1 + x_2 = 2$. Координаты оптимального решения $x_1 = \frac{3}{4}; x_2 = \frac{5}{4}$ можно найти из простых геометрических соображений.

Решение задачи на рис. 32б качественно отличается от первого случая, так как здесь получаются две несвязанных ОДР, дающих две экстремальные точки А и В, являющиеся точками пересечения линий

$x_1 x_2 = 1/2$ и $x_1 + x_2 = 2$.

Точка B является точкой локального минимума, а точка A точкой глобального минимума. Ее координаты найдем, решив систему уравнений:

$$\begin{cases} x_1 x_2 = 1/2, \\ x_1 + x_2 = 2, \end{cases}$$

$$x_1^A = 1 - \frac{\sqrt{2}}{2} \approx 0{,}3, \quad x_2^A = 1 + \frac{\sqrt{2}}{2} \approx 1{,}7.$$

3) $a=2$, $b=1$.

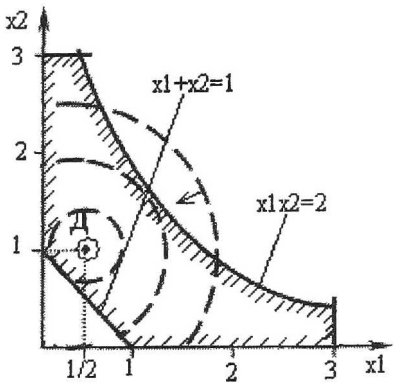


Рис. 33

Решение на рис. 33 находится в точке Д, являющейся координатами центра концентрических окружностей (линий уровня целевой функции).

Из приведенных примеров ясно, что решение задачи НЛП может находиться как внутри ОДР (рис. 33), на одной из границ в произвольной точке (рис. 32а), так и в вершинах ОДР, являющихся точками пересечений функций ограничений (рис. 32б), причем может получиться несколько экстремальных точек.

Графическое изображение ОДР и линий уровня $F(x)$ позволяет решать задачи НЛП с двумя переменными.

6.2. Прямые методы одномерной оптимизации нелинейных функций без ограничений

Пусть на множестве $U \in R$ определена функция $F(X)$. Под *минимизацией функции* $F(X)$ на множестве U будем понимать решение следующей задачи: найти хотя бы одну точку минимума x^* и минимум $F^*=F(X^*)$ этой функции на множестве U.

Многие приближенные методы минимизации применимы только тогда, когда любой локальный минимум $F(X)$ является одновременно и глобальным. Одним из классов функций, удовлетворяющих этому условию, является класс *унимодальных функций.*

Функция $F(X)$ называется *унимодальной* на отрезке $[a;b]$, если она непрерывна на $[a;b]$ и существуют числа α и β, $a \le \alpha \le \beta \le b$, такие, что:

1) если $a<\alpha$, то на отрезке $[a;\ \alpha]$ $F(X)$ монотонно убывает;
2) если $\beta<b$, то на отрезке $[\beta;b]$ F(X) монотонно возрастает;
3) при $x \in [\alpha;\beta]$ $F(X) = F^* = \min\limits_{[a;b]} F(X)$.

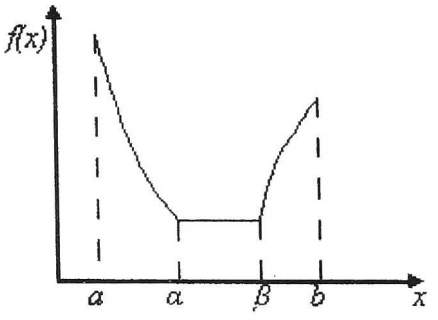


Рис. 34

Для проверки унимодальности функции $F(X)$ на практике обычно используют следующие критерии:

1) *если функция $F(X)$ дифференцируема на отрезке $[a;b]$ и производная $F(X)$ не убывает на этом отрезке,* то *$F(X)$) унимодальна;*
2) *если функция $F(X)$ дважды дифференцируема на отрезке $[a;b]$ и $F(X))\geq 0$ при $x \in [a;b]$, то $F(X))$ также унимодальна.*

Большую группу приближенных методов минимизации функций составляют прямые методы минимизации, основанные на вычислении только значений минимизируемой функции в некоторых точках и не использующие значений ее производных.

Метод перебора (пассивный поиск) является простейшим из прямых методов минимизации. Пусть $F(X)$ унимодальна, и требуется найти какую-либо из точек минимума x^* функции $F(X)$ на отрезке $[a;b]$ с абсолютной погрешностью $\varepsilon>0$. Разобьем $[a;b]$ на n равных частей точками деления $X_i=a+i(b-a)/n$, $i=0, 1, 2, \ldots, n$, где $n\geq(b-a)/\varepsilon$. Вычислив значения $F(X)$ в этих точках, путем сравнения найдем точку x_m, для которой

$$F(X_m)=\min F(X_i), \quad 0\leq i\leq n. \tag{6.1}$$

Далее полагаем $x^*=x_m$, $F^*=F(X_m)$. При этом максимальная погрешность ε_n определения точки X^* равна $\varepsilon_n=(b-a)/n$.

Метод деления отрезка пополам (метод дихотомии) является простейшим последовательным методом минимизации. Он позволяет для любой унимодальной функции $F(X)$ построить последовательность вложенных отрезков $[a;b]$ $[a_1;b_1]$ … $[a_{n-1};b_{n-1}]$ $[a_n;b_n]$, каждый из которых содержит хотя бы одну из точек минимума X^* функции $F(X)$.

Пусть $\varepsilon>0$ – требуемая точность определения точки X^*. Выбрав $\delta=2\varepsilon$, построим последовательности $\{a_n\}$, $\{b_n\}$, $\{X_1^{(n)}\}$, $\{X_2^{(n)}\}$, $n=0, 1, \ldots$, используя рекуррентные формулы $a_0=a$, $b_0=b$;

$$x_1^{(n-1)}=(a_{n-1}+b_{n-1}-\delta)/2,\ x_2^{(n-1)}=(a_{n-1}+b_{n-1}+\delta)/2; \tag{6.2}$$

$$a_n=a_{n-1},\ b_n=x_2^{(n-1)},\ \text{если } f(x_1^{(n-1)})\leq f(x_2^{(n-1)}),$$

$$a_n=x_1^{(n-1)},\ b_n=b_{n-1},\ \text{если } f(x_1^{(n-1)})> f(x_2^{(n-1)}).$$

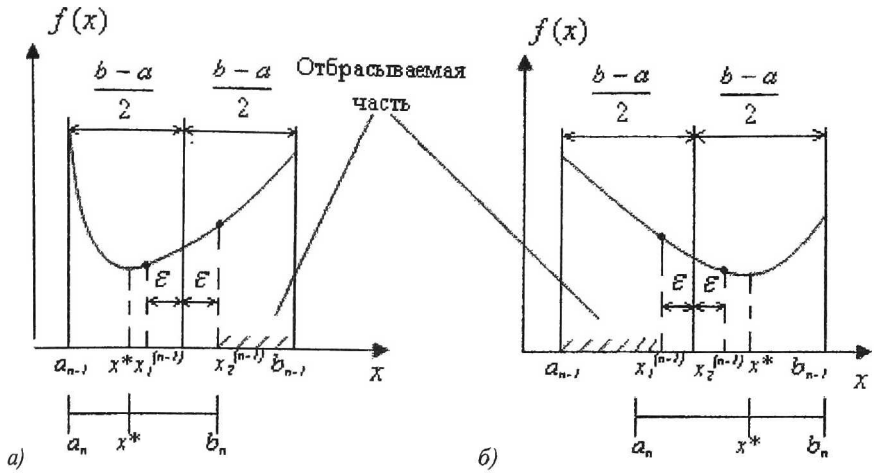


Рис. 35

Переход от отрезка $[a_{n-1};\ b_{n-1}]$ к отрезку $[a_n;\ b_n]$ методом деления отрезка пополам иллюстрируется на рис. 35а, если $F(X_1^{(n-1)}) < F(X_2^{(n-1)})$, и на рис. 35б, если $F(X_1^{(n-1)}) > F(X_2^{(n-1)})$.

Полагая $x^* = (a_n + b_n)/2$, находим X^* с абсолютной погрешностью, не превосходящей величины

$$\varepsilon_n = (b_n - a_n)/2 = (b-a-\delta)/2^{n+1} + \delta/2.$$

Кроме этих методов известны и другие, более эффективные методы, такие как: *метод Фибоначчи, метод золотого сечения, метод поиска по дискретным точкам* [12, 25].

6.3. Градиентные методы многомерной оптимизации

Если ограничения в задаче НЛП отсутствуют, то применяют все методы оптимизации нелинейных функций. Причём, если в задаче с ограничениями решение гарантированно находится внутри области, то его можно найти этими методами, так как эти ограничения не являются активными и их можно отбросить. Рассмотрим наиболее широко применяемые методы оптимизации нелинейных функций многих переменных – это так называемые поисковые методы. Здесь поиск оптимальной точки начинают из начальной точки (ее выбирают из некоторых знаний о примерном месте нахождения решения). Далее в соответствии с некоторым разумным правилом переходят в новую точку с лучшим значением ЦФ и т.д. до тех пор, пока не выполнится некоторое заранее заданное правило остановки, например выполнение условия по точности.

6.3.1. Классический градиентный метод

В этом методе поиска каждая последующая точка получается путем перехода по вектору градиента, т.е. по нормали к линии уровня. Длина шага выбирается из двух основных условий: если длина шага очень большая, то можно перейти точку экстремума, если очень маленькая, то количество шагов от начальной точки до оптимальной будет очень велико. Поиск начинается из начальной точки X^0. Точка $X^{(k+1)}$ ищется по формуле

$$X^{(k+1)}=X^{(k)}\pm\lambda^{(k)}g(X^{(k)}),$$

(+) – если $F(X) \to \max$ и (-) – если $F(X) \to \min$,

где $g(X^{(k)})$ – градиент в точке X^k, $\lambda^{(k)}$ - множитель, определяющий длину шага. В координатах $X_j^{(k+1)}=X_j^{(k)}+\lambda^{(k)}(\frac{\partial F}{\partial X_j})_{X^k}$. Величина $\lambda^{(k)}$ может в простейшем случае не изменяться по мере движения $\lambda^{(k)}=\lambda=\text{const}$.

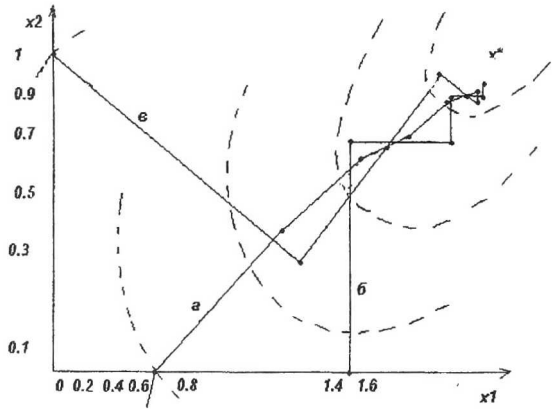


Рис. 36

Теоретически окончание поиска определяется следующим условием: $X^{(r)}$ - точка экстремума, если $g(X^{(r)}) = 0$. На рисунке показаны траектории движения для
а) – классического градиентного метода;
б) – покоординатного метода,
в) – метода наискорейшего спуска.

Для практических вычислений правила остановки могут быть различны. Для простоты можно считать, что X^r является точкой экстремума , если $\max\left|(\frac{\partial F}{\partial X_j})_{x^{(r-1)}}\right| \le \varepsilon$, где ε - заданная малая величина.

Ясно, что этот метод можно применять в том случае, когда можно хотя бы численно вычислить градиент функции в точке. Заметим, что для гладкого оптимума при приближении к оптимальной точке величина градиента автоматически уменьшается, поэтому даже при постоянной величине λ длина шага уменьшается, что является желательным для любого поискового метода.

6.3.2. Покоординатный метод

В этом методе движение из начальной точки производится сначала вдоль первой координаты x_1. Знак направления совпадает с направлением проекции градиента на эту координату (для $F(X) \rightarrow \max$) или противоположен ему, если $F(X) \rightarrow \min$. Конец шага определяется точкой, где $\frac{\partial F}{\partial X_1} = 0$. Затем движение производится по второй координате X_2, затем по X_3 и т.д., и, наконец, по X_n. Затем повторяют движение опять по X_1, X_2 и т.д. Можно использовать следующее правило остановки: $\max \left| \left(\frac{\partial F}{\partial X_j}\right)_{x^{(r-1)}} \right| \le \varepsilon$.

6.3.3. Метод наискорейшего спуска и его модификации

В этом методе из некоторой начальной точки движение осуществляется вдоль направления градиента до тех пор, пока производная по этому направлению не будет равна нулю. Далее из этой точки определяем новый градиент и т. д. Отличие здесь в том, что длина шага из точки X^k определяется из условия, чтобы обеспечить:

$$\min_{\lambda>0} F\left[X^k \pm \lambda g\left(X^k\right)\right].$$

Эту вспомогательную задачу одномерной оптимизации можно решать на основе рассмотренных методов прямого поиска (дихотомии и т.д.)

Наискорейший и покоординатный методы называют методами с длинным шагом. Кроме этого существуют методы, использующие вторую производную для определения длины шага, например, метод Ньютона:

$$X^{k+1} = X^k + \left[F''\left(X^k\right)\right]^{-1} g\left(X^k\right),$$

где $[F''(X^k)]^{-1}$ - обратная матрица вторых производных в точке X^k.

6.4. Метод деформируемого многогранника Нелдера-Мида

Современные методы поиска минимума (максимума) нелинейной функции чрезвычайно разнообразны. Одним из наиболее эффективных является метод Нелдера-Мида. Идея метода состоит в том, что для определения направления движения вычисляются значения целевой функции в вершинах сначала правильного, а затем деформируемого многогранника. Многогранник некоторое тело в n-мерном пространстве. Правильный многогранник называется симплексом, в задачах для двух переменных это правильный треугольник.

Вычисленные значения целевой функции в вершинах симплекса сравниваются, а затем выполняются операции:

1) отражение - поворот симплекса через одну из сторон;
2) растяжение - если идём в правильном направлении;
3) редукция - возврат назад, если перескочили оптимум;
4) сжатие - уменьшение сторон, чтобы движение было с более мелким шагом.

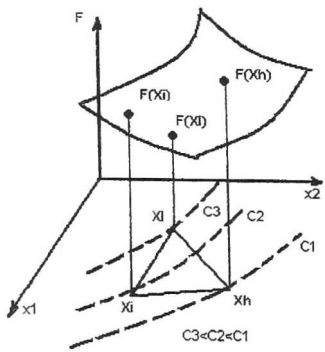


Рис. 37

В этом методе не используется понятие производной функции, что расширяет его возможности. Поиск оптимальной точки ведется путем поворота и деформации многогранника (симплекса) на основе анализа вершин в соответствии со следующими вышеприведенными операциями. Пусть k – номер итерации, тогда алгоритм Нелдера-Мида заключается в следующем:

1. Выбирается начальный симплекс (если в задаче 2 переменные n=2, то симплекс–правильный треугольник). Его вершины $X_i^{(0)}, i=\overline{1,n+1}$. Обозначим $X_h^{(k)}$ - самая худшая точка, т.е. для $F(X)\to\min$, $F\left(X_n^{(k)}\right)=\max F(X_i^k)$; $X_l^{(k)}$ - лучшая точка, т.е. $F\left(X_l^{(k)}\right)=\min F(X_i^k)$;. $X_0^{(k)}$ - центр тяжести всех вершин, исключая X_h^k, его координаты

$$X_0^{(k)}=\frac{1}{n}\left[\left(\sum_{i=1}^{n+1}X_{ij}^{(k)}\right)-X_{nj}^{(k)}\right];\quad j=\overline{1,n}.$$

2. Осуществляют отражение (проектирование) $X_h^{(k)}$ через центр тяжести, новая точка получается так:

$$X_\alpha^{(k)}=X_0^{(k)}+\alpha\left(X_0^{(k)}-X_n^{(k)}\right),$$

где α - коэффициент отражения. Обычно α=1.

- Если $F\left(X_\alpha^{(k)}\right)\le F\left(X_l^{(k)}\right)$, то вектор $X_\alpha^{(k)}-X_0^{(k)}$ растягивается в γ раз и получаем $X_b^{(k)}=X_0^{(k)}+\gamma\left(X_a^{(k)}-X_0^{(k)}\right)$.

- Если для полученной точки $F\left(X_b^{(k)}\right) < F\left(X_l^{(k)}\right)$, то $X_h^{(k)}$ заменяется на $X_b^{(k)}$, и переходим к пункту 2. В противном случае заменяем $X_h^{(k)}$ на $X_a^{(k)}$, и переходим к пункту 2.
- Если $F\left(X_\alpha^{(k)}\right) > F\left(X_h^{(k)}\right)$, для всех $i \neq h$, то вектор $X_h^{(k)} - X_0^{(k)}$ сжимается, получаем точку $X_c^{(k)} = X_0^{k} + \beta\left(X_0^{(k)} - X_h^{(k)}\right)$, $0 < \beta < 1$.

  Точка $X_h^{(k)}$ заменяется на $X_c^{(k)}$, и переходим к выполнению пункта 2.
- Если $F\left(X_\alpha^{(k)}\right) > F\left(X_h^{(k)}\right)$, то все векторы $X_i^{(k)} - X_l^{(k)}$ уменьшаются в λ раз (обычно λ=2) - редукция $X_i^{(k)} = X_l^{k} + \frac{1}{\lambda}\left(X_i^{(k)} - X_l^{(k)}\right)$; $i = \overline{1, n+1}$.

Возвращаемся к пункту 2. Правила остановки могут быть различными, например,

$$\left\{\frac{1}{1+n}\sum_{i=1}^{n+1}\left[F(X_i^r) - F(X_0^r)\right]^2\right\}^{\frac{1}{2}} \leq \delta;$$

для простоты можно использовать правило:

$$\max_{j,l,p}\left|X_{i,j}^r - X_{p,j}^r\right| \leq \varepsilon \qquad \begin{matrix} i = \overline{1, n+1} \\ j = \overline{1, n} \\ p = \overline{1, n+1} \end{matrix}$$

решением будет точка $X_0^{(r)}$.

Например:

$F(X) = -6X_1 + 2X_1^2 - 2X_1X_2 + 2X^2 \to \min$,

выберем параметры:

$\alpha = 1, \beta = \frac{1}{2}, \gamma = 2, \lambda = 2, \varepsilon = 0{,}1$,

решение видно из рисунка 38.

Подробнее смотри в [16].

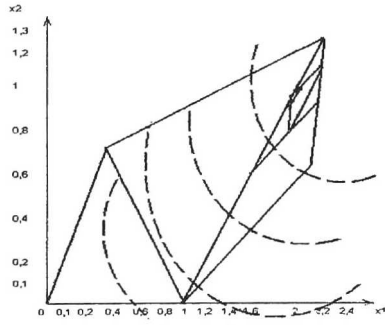


Рис. 38

6.5. Задача НЛП с ограничениями-равенствами

В НЛП особо рассматриваются задачи с ограничениями-равенствами. Это так называемые задачи на условный экстремум. Решение такой задачи производится с использованием функции Лагранжа. Эта функция позволяет

построить новую целевую функцию, которая бы в виде штрафа учитывала ограничения:

$$\Phi(X,\lambda)=F(X)+\sum_{i=1}^{m}\lambda_i\left(\varphi_i(X)-b_i\right), \tag{6.3}$$

где $F(X)$ - целевая функция, λ_i – множители Лагранжа, имеющие смысл коэффициентов чувствительности ограничений.

Таким образом, задача минимизации нелинейной целевой функции с ограничениями-равенствами сводится к минимизации функции Лагранжа без ограничений. Для её решения используется классический приём: записываются необходимые условия существования экстремума:

$$\begin{cases}\dfrac{\partial\Phi}{\partial X_j}=\dfrac{\partial F}{\partial X_j}+\sum\limits_{i=1}^{m}\left(\lambda_i\dfrac{\partial\varphi_i}{\partial X_j}\right)=0, & j=\overline{1,n};\\ \dfrac{\partial\Phi}{\partial\lambda_i}=\varphi_i(X)-b_i=0, & i=\overline{1,m}.\end{cases} \tag{6.4}$$

После выполнения операций дифференцирования получаем систему в общем случае нелинейных уравнений относительно неизвестных X_i и λ_i - множителей Лагранжа.

Решение системы (6.4) дает точки подозрительные на экстремум, и их ещё надо проверить с помощью достаточных признаков экстремума. Рассмотрим некоторые из этих признаков.

1) Пусть X^* - точка подозрительная на экстремум, полученная с помощью выражений (6.4). Рассмотрим матрицу Гессе – матрицу H вторых производных в точке X^*:

$$H=\frac{\partial^2\Phi}{\partial X_i\partial X_j}. \tag{6.5}$$

Если эта матрица определена положительно, то точка X^* -минимум, а если определена отрицательно, то максимум. Если же матрица Гессе знакоопределена, то в этой точке экстремума нет, а если полуопределена, то этот признак не даёт ответа на вопрос.

2) Пусть D_k - главный определитель матрицы Гессе k-го порядка:

а) если $D_1>0, D_2>0, D_3>0,\ldots$ - это достаточный признак минимума;

б) если $D_1<0, D_2>0, D_3<0,\ldots$ - достаточный признак максимума (знаки строгих неравенств чередуются);

в) если же в определителях знаки $\geq$ и $\leq$, то это необходимый, а не достаточный признак и вопрос об экстремуме не решается.

3) Наиболее «сильным» достаточным признаком является следующий. Составим расширенную матрицу Гессе H^B, для чего используем матрицу производных от функций ограничений:

$$Q=\begin{pmatrix} \frac{\partial\varphi_1}{\partial X_1} & \frac{\partial\varphi_1}{\partial X_2} & \cdots & \frac{\partial\varphi_1}{\partial X_n} \\ \frac{\partial\varphi_2}{\partial X_1} & \frac{\partial\varphi_2}{\partial X_2} & \cdots & \frac{\partial\varphi_2}{\partial X_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial\varphi_n}{\partial X_1} & \frac{\partial\varphi_n}{\partial X_2} & \cdots & \frac{\partial\varphi_n}{\partial X_n} \end{pmatrix},\quad H^B=\begin{pmatrix} 0 & Q \\ Q' & H \end{pmatrix},$$

Q' - транспонированная матрица.

Матрица H^B имеет размерность $(m+n)\times(m+n)$.

Признак: точка X^* соответствует максимуму, если начиная с главного определителя порядка $m+1$, последние n-m определителей матрицы H^B образуют знакопеременный ряд, а знак определителя, стоящего на $(n$-$m)$-м месте от конца, совпадает с знаком: если $(-1)^{m+1}$, то max; если $(-1)^m$, то min.

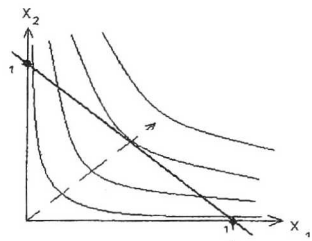


Рис. 39

Например: $n=3$, $m=2$, n-$m=1$, проверяем D_5;
$n=7$, $m=4$, n-$m=3$, проверяем D_9.

Пример:

$$F = X_1X_2 \to \max,\quad X_1 + X_2 = 1.$$

Запишем функцию Лагранжа

$$\Phi(X_1, X_2, \lambda) = X_1X_2 + \lambda(X_1 + X_2 - 1) \to \max.$$

Найдем решение, используя необходимые условия

$$\left.\begin{aligned}\frac{\partial \Phi}{\partial X_1} &= X_2 + \lambda = 0\\ \frac{\partial \Phi}{\partial X_2} &= X_1 + \lambda = 0\\ \frac{\partial \Phi}{\partial \lambda} &= X_1 + X_2 - 1 = 0\end{aligned}\right\}.$$

Получаем решение системы:

$$X_1 = X_2;$$

$$X_1 = X_2 = \frac{1}{2}; \quad \lambda = -\frac{1}{2}.$$

Запишем матрицу H^B:

$$H^B = \begin{pmatrix} 0 & Q \\ Q' & H \end{pmatrix}; \; H = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix};$$

$$H^B = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}; \; n=2; \; m=1; \; m+1=2; \; n-m=1.$$

Начиная со второго определителя (m+1=2) знаки чередуются:

$D_1 = 0$, $D_2 < 0$, $D_3 = 2 > 0$, причем $(-1)^{m+1} = (-1)^{1+1} > 0$.

Проверяем: $D_3 > 0$, значит в точке $X_1 = X_2 = 1/2$ находится максимум.

Геометрически задача на условный экстремум сводится к тому, что решение находится в той точке, где линия (поверхность), определяющая функцию (систему) ограничений, касается какой-нибудь линии уровня (см. рисунок примера).

6.6. Выпуклое НЛП

Теория нелинейного программирования разработана только для одного частного случая выпуклых целевых функций $F(X)$ и ограничений $\psi_i(X)$ и этот раздел соответственно называется выпуклым программированием.

Выпуклое множество обладает тем свойством, что отрезок, соединяющий любые две точки множества, также принадлежит этому множеству.

Таким образом, $C \subset E^n$ выпукло, если из $(X^1, X^2) \in C$ следует, что $\lambda = \theta X^1 + (1-\theta) X^2 \in C$ для любого $0 \le \theta \le 1$.

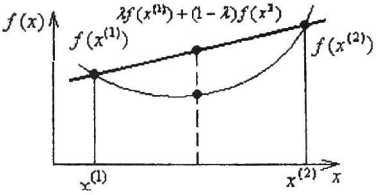


Рис. 40

Важно: если C_i - выпуклое множество, то $C = \cap C_i$ также выпукло.

Т.е. если каждое ограничение в задаче НЛП образует выпуклое множество, то и все ограничения дают выпуклое множество.

Дважды дифференцируемая функция $F(x)$ является выпуклой в том и только в том случае, когда

$$\sum_{i=1}^{n}\sum_{j=1}^{h}\frac{\partial^2 F(X)}{\partial X_i \partial X_j}\Delta X_i \Delta X_j \ge 0$$

для любых $x \in M$, где M - выпуклое множество и ΔX_i и ΔX_j не обращаются в нуль одновременно. Для проверки выпуклости $F(X)$ используется критерий Сильвестра:

Функция $F(X)$ - выпукла, если неотрицательны (для строго выпуклой положительны) все главные миноры матрицы Гессе:

$$\Delta_k = \begin{vmatrix} a_{11} & \dots & a_{1k} \\ & \dots & \\ a_{k1} & \dots & a_{kk} \end{vmatrix};$$

$$a_{ij} = \frac{\partial^2 F}{\partial X_i \partial X_j};$$

$k = 1, 2, \dots n.$

Если все $\Delta_k > 0$, то $F(X)$ - строго выпукла.

Функция $F(X)$ n переменных $\|X_1, X_{21}, \dots, X_n\| = X \in G$ называется выпуклой функцией в выпуклой области G, если для любых двух точек выполняется соотношение

$$F\{\lambda X^{(1)} + (1-\lambda) X^{(2)}\} \le \lambda F(X^{(1)}) + (1-\lambda) F(X^{(2)}), \text{ где } 0 < \gamma < 1.$$

Когда λ – переменная, то промежуточная точка пробегает значения от $X^{(1)}$ до $X^{(2)}$.

Функция строго выпукла, если знак $\le$ можно заменить на $<$.

Т.о. для задачи на min выпуклая функция не может принимать больших значений, чем линейная (для задачи на max меньших значений).

Другим важным свойством функций, выполнение которого позволяет гарантировать решение задачи НЛП является так называемое условие Липшица.

Говорят, что функция $F(X)$ удовлетворяет на отрезке $[a, b]$ условию Липшица, если существует такое число $L>0$ (константа Липшица), что

$$\left|F(X^{\text{I}})-F(X^{\text{II}})\right| \le L\left|X^{\text{I}}-X^{\text{II}}\right|. \qquad (6.7)$$

Для всех X^{I}, $X^{\text{II}} \in [a,b]$.

Для проверки условия Липшица на практике используют следующий фактор:

Если функция $F(X)$ имеет на отрезке [a, b] ограниченную производную, то она удовлетворяет условию (6.7), где

$$L \ge \max_{[a,b]} |F^{\text{I}}(X)|.$$

6.7. Теорема Куна-Таккера для выпуклого НЛП

Функция Лагранжа является основным инструментом для теоретического обоснования решения задачи НЛП. Если задача такова, что

$$\begin{aligned} &F(X) \to \min, \\ &\varphi_i(X) \le 0, i = \overline{1,m}, \end{aligned} \qquad (6.8)$$

все функции обладают свойством выпуклости, то справедлива фундаментальная теорема Куна-Таккера о том, что решение (6.8) находится в так называемой седловой точке функции Лагранжа.

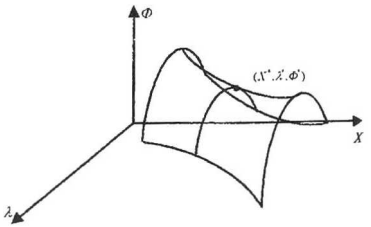


Рис. 41

Функция $\Phi(X, \lambda)$ двух векторных переменных $X=(X_1....X_n)$ и $\lambda=(\lambda_1....\lambda_n)$ имеет в точке (X^*,λ^*) седловую точку, если выполняются соотношения:

$$\Phi(X,\lambda^*) \ge \Phi(X^*,\lambda^*) \ge \Phi(X^*,\lambda) \text{ для всех } X \text{ и } \lambda \text{ из } \varepsilon\text{-окрестности } (X^*,\lambda^*).$$

Для простейшего случая одной переменной X и одного множителя Лагранжа седловая точка геометрически изображена на рисунке 41 (будем счи-

тать, что $X \geq 0$ и $\lambda \geq 0$). В седловой точке максимум функции Лагранжа по λ совпадает с минимумом по X.

Необходимыми и достаточными условиями существования седловой точки функции двух векторных переменных являются соотношения

$$\left.\frac{\partial \Phi}{\partial X_j}\right|_{X^*,\lambda^*} \geq 0, \qquad \left(\frac{\partial \Phi}{\partial \lambda_j}\right)_{X^*,\lambda^*} \leq 0,$$

$$X_i^*\left(\frac{\partial \Phi}{\partial X_j}\right)_{X^*,\lambda^*} = 0, \qquad \lambda_i^*\left(\frac{\partial \Phi}{\partial \lambda_j}\right)_{X^*,\lambda^*} = 0, \tag{6.9}$$

$$X_j^* \geq 0, j = \overline{1,n}, \qquad \lambda_i^* \geq 0, i = \overline{1,n}.$$

Эти соотношения связаны с фундаментальной теоремой нелинейного программирования Куна-Таккера о том, что решение задачи НЛП находится в седловой точке функции Лагранжа.

Соотношения (6.9) являются не только необходимыми, но и достаточными, если кроме условий (6.9) выполняются условия регулярности Слейтера: в области допустимых решений существует такая точка X, в которой все ограничения не активны, т.е. $\varphi_i(X)<0$ для всех i=1,m (в области есть «внутренние» точки).

Теорема Куна-Таккера. Если в задаче НЛП

$$F(X) \to \min; \quad \varphi_i(X) \leq 0, \; i = \overline{1,m}; \quad \Phi(X,\lambda) = F(X) + \sum_{i=1}^{m} \lambda_i \varphi_i(X)$$

все функции выпуклы, а множество допустимых решений, определяемое ограничениями, удовлетворяет условиям регулярности Слейтера, то для оптимальности точки X^* необходимо и достаточно выполнение условий (6.9).

Если все ограничения линейные, то условия Слейтера в теореме не обязательны.

6.8. Квадратичное программирование

Задачей квадратичного программирования (КП) называется такая задача НЛП, когда целевая функция представляет собой сумму линейной и квадратичной формы (переменные не старше второй степени), а все ограничения линейные. Для решения таких задач используется симплекс-метод.

Теорема Куна-Таккера позволяет записать условие Куна-Таккера для задачи КП и найти седловую точку. Рассмотрим задачу КП в виде

$$F(X) = \sum_{j=1}^{n} p_j X_j + \sum_{k=1}^{n}\sum_{j=1}^{n}\left(C_{kj} X_k X_j\right) \to \min,$$

$$\varphi_i(X)=\sum_{j=1}^{n}a_{ij}X_j-b_i\le 0,\quad i=\overline{1,m},\quad j=\overline{1,n},\quad X_j\ge 0.$$

В матричном виде - пусть P, X, B - векторы-столбцы:

$$F=p'X+X'CX\to\min,\quad AX\le b,\quad X\ge 0,$$

C - матрица квадратичной формы, которая должна быть симметрична и положительно-полуопределена, это гарантирует выпуклость F.

Метод Баранкина-Дорфмана непосредственно основан на применении теоремы Куна-Таккера. Запишем функцию Лагранжа в матричной форме для задачи КП:

$$\Phi(X\lambda)=\underbrace{p'X+X'CX}_{F(X)}+\underbrace{\lambda(AX-b)}_{штраф};$$

$$\frac{\partial\Phi}{\partial X}=V=p'+2CX+A'\lambda;\qquad -\frac{\partial\Phi}{\partial\lambda}=Y=-AX+b.$$

Тогда условие Куна-Таккера можно записать в следующем виде:

$$\left.\begin{array}{rl}AX+V=b & \\ 2CX-V+A'\lambda=-p & (*)\\ X\ge 0;\ Y\ge 0;\ V\ge 0;\ \lambda\ge 0 & \end{array}\right\}$$
$$XV+Y\lambda=0 \qquad (**)$$

Неизвестными являются (X, Y, V, λ).

(*) - система линейных уравнений относительно неизвестных; решить её, значит найти допустимые решения задачи ЛП.

(**) - нелинейное условие.

Тогда можно переходить из одного допустимого решения к другому и проверять условие (**).

Введём векторы Z= (X, Y, V, λ) и $\tilde{Z}$ =(V, λ, X, Y), тогда можно записать

$$XV+Y\lambda=\frac{1}{2}Z\times\tilde{Z}.$$

Система (*) также может быть представлена через векторы Z, тогда окончательно условия Куна-Таккера будут иметь следующий вид:

$$\begin{pmatrix} A & E & 0 & 0 \\ 2C & 0 & -E & A' \end{pmatrix} \times Z' = \begin{pmatrix} b \\ -p \end{pmatrix}, \quad Z \geq 0, \tag{6.10}$$

$$T(Z) = Z \times \tilde{Z} = 0. \tag{6.11}$$

Метод Баранкина-Дорфмана заключается в следующем: находится допустимый вектор Z, удовлетворяющий выражению (6.10), и проверяется условие (6.11). Далее выбирается новое базисное решение, причём оно выбирается так, чтобы величина $T(Z)$ всё время уменьшалась. Для этого используется модифицированная симплекс-таблица, в которой генеральный элемент находится из условия минимизации выпуклой функции $T(Z)$:

$$T(Z) = Z \times \tilde{Z} \to \min, \tag{6.12}$$

т. е. мы решаем задачи (6.10) и (6.12), вместо (6.10) и (6.11). В симплекс-таблице записываются в качестве базисных переменных все переменные. Запишем строку базисных переменных:

$$Z_g = d_{g_0} + \sum_{h=1}^{n} d_{g_h} \times t_h, \quad g = \overline{1,2N},$$

где $2N = 2(n+m)$, а t_h - свободные переменные.

Если строка соответствует свободным переменным, то в строке одна единица и остальные нули. Для выбора генерального элемента используются дополнительные элементы, которые записываются в дополнительные строки:

$$\alpha_j = \sum_{i=1}^{N} \left(d_{ij} \times d_{i+N,0} + d_{i+N,j} \times d_{i0} \right),$$

$$\beta_j = 2 \sum_{i=1}^{N} d_{ij} \times d_{i+N,j},$$

$$\Theta_j = \min_g \frac{d_{g_0}}{\left| d_{g_j} \right|}, \text{ при } d_{g_j} < 0 \quad (\arg\min = r),$$

$$K_j = 2\alpha_j + \Theta_j \beta_j$$

Можно показать, что новое значение после замены j-й свободной переменной будет $T_j=T+\Theta_j K_j$. По определению $\Theta_j>0$, поэтому, чтобы T уменьшалось, K_j должно быть меньше нуля. Рассмотрим величину K_j: величина β_j - вторая производная функции $T(Z)$, которая выпукла вниз, и поэтому $\beta_j>0$, значит K_j может быть отрицательной, если $\alpha_j>0$, значит в качестве генерального надо выбрать этот j-й столбец, строку надо выбрать ту, для которой вычисляется величина Θ.

Пример:

$$F(x) = -6x_1 + 2x_1^2 - 2x_1x_2 + 2x_2^2 \to \min ,$$

$$x_1 + x_2 \le 2 ,$$

$$x_1 \ge 0 ,$$

$$x_2 \ge 0 .$$

Запишем все матрицы, которые нам нужны:

$$A = (1,1); \quad B = 2; \quad C = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix};$$

$$p = \begin{pmatrix} -6 \\ 0 \end{pmatrix}; \quad n = 2; \quad m = 1; \quad N = 3.$$

Запишем условие Куна-Таккера, определяемое выражением (1):

$$\begin{pmatrix} (1 & 1) & 1 & (0 & 0) & 0 \\ \begin{pmatrix} 4 & -2 \\ -2 & 4 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ V_1 \\ V_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \\ 0 \end{pmatrix}.$$

Перемножим матрицы, получим систему

$$x_1 + x_2 + y_1 = 2 ,$$

$$4x_1 - 2x_2 - V_1 + \lambda_1 = 6,$$

$$-2x_1 + 4x_2 - V_2 + \lambda_1 = 0.$$

Чтобы записать симплекс-таблицу, надо выразить базисные переменные через свободные. Для получения первого базисного решения используется любой метод, например, метод Гаусса. Можно использовать метод искусственного базиса (см. главу 2), который использует симплекс-процедуру. Затем нужно получить опорное решение, и только после этого записать модифицированную симплекс-таблицу. Для простых задач можно подобрать первое базисное решение так, чтобы сразу получить опорное решение (свободные члены >0). Для нашего примера

$$Y_1 = 2 - X_1 - X_2,$$

$$X_1 = 6 - 4X_1 + 2X_2 + V_1,$$

$$V_2 = 6 - 6X_1 + 6X_2 + V_1.$$

Запишем симплекс-таблицы по методу Баранкина-Дорфмана.

Таблица 37

	1	X_1	X_2	V_1
X_1		*1*		
X_2			*1*	
Y_1	*2*	*-1*	*-1*	
V_1				*1*
V_2	*6*	*-6*	*6*	*1*
λ_1	*6*	*-4*	*2*	*1*
α_j	*24*	*-14*	*4*	*2*
β_j		*8*		
Θ_j		*1*		
K_j		*-20*		

Таблица 38

	1	V_2	X_2	V_1
X_1	*1*	*-1/6*	*1*	*1/6*
X_2			*1*	
Y_1	*1*	*1/6*	*-2*	*-1/6*
V_1				*1*
V_2		*1*		
λ_1	*2*	*2/3*	*-2*	*1/3*
α_j	*4*	*1*	*-6*	*1*
β_j			*8*	
Θ_j			*1/2*	
K_j			*-8*	

Примечание. Алгоритм симплекс-преобразования аналогичен рассмотренному в главе 2, только: «...генеральную строку умножаем на $-\lambda$, а столбец на λ, отмечаем в строке нижние элементы, а в столбце верхние». Порядок строк в таблице нарушать нельзя.

В первой таблице $T = a_0 = 24$, а во второй 4. Чтобы не вычислять всю третью таблицу подсчитаем a_0 для следующей таблицы по формуле

$$T = 2*4+1/2\ (-8) = 0,$$

значит следующая таблица оптимальная. Вычислим только необходимые элементы (свободные члены), получим: $X_1=3/2$; $X_2=1/2$; $Y_1=V_1=V_2=\lambda_1=0$. Оптимальное значение $F = -11/2$.

Недостаток метода: иногда встречаются задачи, когда все $K_j>0$. Значит, мы будем переходить в новую вершину, а значение T будет увеличиваться. В этом случае идём на временное ухудшение T (проходим через «мёртвую зону»). В методе Франца-Вульфа это учтено [1,7,12,16].

6.9. Методы возможных направлений

Наиболее распространенным классом методов решения выпуклых задач НЛП являются градиентные методы, которые отличаются как способом получения последующей точки, так и способом «обработки» ограничений.

В методах возможных направлений переход от точки X_k к точке X^{k+1} осуществляется по направлению S_k, не обязательно вдоль градиента. При этом движение вдоль S_k должно быть таким, что бы новая точка принадлежала области допустимых решений D. Направление, удовлетворяющее этому условию, называется возможным или допустимым:

$$X^{k+1} = \left(X^k - \lambda S^k\right) \in D. \tag{6.13}$$

Таких направлений множество; среди возможных направлений выбирают такое, для которого скалярное произведение градиентов в точке k и этого направления больше нуля (для задачи минимизации меньше нуля). Такое направление называется подходящим. Таких направлений в общем случае также множество. Г.Зойтендейк предложил выбирать то, которое максимально увеличивает значение целевой функции.

Рассмотрим задачу выпуклого НЛП:

$$F(x) \to \max(min), \tag{6.14}$$

$$\varphi_i(x) \le 0, i = \overline{1,m}. \tag{6.15}$$

Обозначим область допустимых решений через D, пусть x^0- начальная точка, а X^* - единственное решение задачи, $X^0 \in D, x^0 \notin x^*$.

Переход из точки $X^{(k)}$ в $X^{(k+1)}$ осуществляют по такому направлению $S^{(k)}$, чтобы при $\lambda>0$ луч $x^{(k)}+\lambda S^{(k)}$ принадлежал D.

Обозначим $g_0(X)$ и $g_i(X)$ градиенты функций $F(X)$ и $\varphi_i(X)$ в точке X соответственно. Обозначим $I(X)$ множество индексов I таких, что $\varphi_i(X)=0$; $I(X)$ указывает номера ограничений, выполняющихся в точке X точно ($I(X)$ – активные ограничения).

Пусть S- некоторый вектор, $S\neq 0$, такой, что для всех $i\in I(x^0)$ выполняется неравенство $\left(g_0{}'(x^0),S\right)<0$.

Знак (•)' обозначает транспортированный вектор (матрицу).

Тогда можно определить $\lambda>0$, чтобы $x^0+\lambda S\in D$. Для этого сначала определим $\lambda'\le \min_{i\in I(x^0)} d_i$, где d_i – корень уравнения $\varphi_i\left(x^0+d_iS\right)=0$ (если нет решения, то $\alpha_i=+\infty$). Затем определим $\lambda''\le \min_{i\in J(x^0)} C_i$, где C_i – корень уравнения $\left(g_i{}'(x^0+C_i,S),S\right)=0$ (если нет решения, то $C_i=\infty$). Таким образом, λ' выбирается из условия невыхода из области D, а λ'' из условия попадания в точку экстремума. Тогда, если $\lambda\le\min\left(\lambda',\lambda''\right)$, то $x^0+\lambda S\in D$. Направление S, удовлетворяющее условию (6.15), и называют *возможным* (допустимым) в точке x^0. То из возможных направлений, для которого, кроме этого, выполняется условие $\left(g_0{}'(x^0),S\right)>0$, называется *подходящим*, так как при движении в этом направлении значения целевой функции улучшаются, не выходя из D (для задачи на минимум $\left(g_0{}'(x^0),S\right)<0$).

Далее находится направление S из подходящих направлений, которое обеспечивает максимально возможное увеличение целевой функции, т.е. на каждом шаге решается вспомогательная задача

$$\left(g_0{}'(x^0),S\right)\to\max, \tag{6.16}$$

$$\varphi_i\left(x^{(k)}+S\right)\le 0, i\in I\left(x^0\right). \tag{6.17}$$

Для нормализации длины вектора S используют следующие условия:

$$N_1: S'S\le 1,\ N_3:\begin{cases} S_i\le 1 \text{ для } Sg_i>0\\ S_i\ge 1 \text{ для } Sg_i<0\end{cases},$$

$$N_2: -1\le S_j\le 1, \forall j=\overline{1,n},\ N_4: g'S\le 1\ldots,$$

$$N_5:\varphi_i\left(x^{(k)}+S\right)\le 0, \forall i=\overline{1,m}.$$

Видно, что условие (6.17) уже содержится в N_5. Точка X^* является решением, как только $(g_0'(x^*), S) = 0$. Метод Зойтендейка можно применять для любых выпуклых функций $F(x)$ и $\varphi_i(x)$, но он может не дать решения за конечное число шагов. Для задачи квадратичного программирования можно с использованием принципа сопряженности обеспечить сходимость за конечное число шагов.

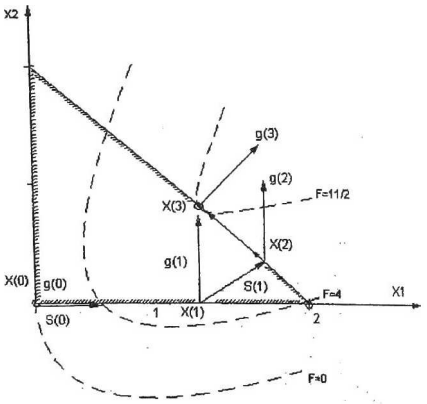


Рис. 42

Рассмотрим этот случай. Если $x^{(k)}$ есть внутренняя точка, т.е. $\lambda^{(k-1)} = \lambda'' < \lambda'$, то новое искомое $S^{(k)}$ должно удовлетворять, помимо условия нормировки, еще одному условию сопряженности $\left(S^{(k)}\right)' CS^{(k-1)} = 0$, где C - положительно полуопределенная матрица квадратичной формы для задачи квадратичного программирования:

$$Q - p'x - x'cx \to max, \tag{6.18}$$

$$a_i x \le b_i, i = \overline{1,m}. \tag{6.19}$$

Если для определения $S^{(k-1)}$ такое условие уже использовалось, то оно остается, так что все условия сопряженности имеют вид

$$\left(S^{(k)}\right)' CS^h = 0, \quad \text{для } h = k-1, k-2, \ldots, k-r.$$

Сопряженность последующего направления предыдущему и обеспечивает конечность шагов при поиске решения для задачи квадратичного программирования. Это условие позволяет совместить траекторию движения с осями поверхности целевой функции.

Рассмотрим пример с нормирующими условиями N_5 для задачи квадратичного программирования (6.18), (6.19). Для нее:

$$g_0\left(x^{(k)}\right) = p - 2cx^{(k)} = g^k.$$

Длина очередного шага определяется из соотношения:

$$\lambda^k = \min\left(\lambda'', \lambda'\right).$$

Величина λ'' определяется из условия попадания в экстремум, т.е.

$$\left(S^{(k)}\right)' g\left(x^{(k)} + \lambda'' S^{(k)}\right) = 0.$$

Подставим выражение градиента, получим

$$S^{(k)\prime}\left[p - 2c\left(x^{(k)} + \lambda'' S^{(k)}\right)\right] = S^{(k)\prime} p - S^{(k)\prime} 2cx^{(k)} - S^{(k)\prime} 2c\lambda'' S^{(k)} =$$

$$= S^{(k)\prime} g\left(x^{(k)}\right) - S^{(k)\prime} 2c\lambda'' S^{(k)} = 0.$$

Отсюда получаем

$$\lambda'' = \frac{g^{k\prime} S^{(k)}}{2S^{(k)\prime} cS^{(k)}}.$$

Величина λ' для N_5 всегда равна 1.
На рис.42 изображена иллюстрация поиска решения для примера

$$F(x) = 6x_1 - 2x_1^2 + 2x_1x_2 - 2x_2^2 \to \max,$$

$$x_1 + x^2 \le 2,$$

$$x_1, x \ge 0.$$

Для решения неквадратичных задач НЛП принцип сопряженности уже неприменим и для построения алгоритма, сходящегося к решению, приходится применять процедуру изменения длины шага. Рассмотрим такой алгоритм по методу возможных направлений Зойтендейка. При выборе подходящего направления будем принимать во внимание не только те ограничения, которые в данной точке выполняются точно, но и те, которые выполняются «почти точно». В противном случае λ может измельчаться вдали от точки минимума. Для $\varepsilon > 0$ и $x \in D$ определим теперь ε – активное индексное множество $I_\varepsilon(x)$. Это те i, для которых $\varphi_i(x) + \varepsilon \ge 0$, а также i=0, т.е. целевая функция $F(x)$. Для выбора λ и S в точке $x^{(k)}$ для получения следующей точки $x^{(k+1)} = x^{(k)} + \lambda S$ в соответствии с методом Зойтендейка необходимо выполнить условия невыхода из D, нормировки длины вектора λS и условия большего возрастания (убывания для задачи на минимум) целевой функции. В алгоритме, излагаемом ниже, последнее требование заменится требованием выбора направления, имеющего наибольший угол к касательным с ограничениями вблизи точки $x^{(k)}$, что дает возможность двигаться с большим шагом, не выходя за пределы D, а длина этого шага выбирается из условия наибольшего увеличения (уменьшения) $F(x)$. В качестве нормировки выберем

N_2 . Тогда для задачи минимизации $F(x) \to min$ для заданного ε в точке x необходимо определить величину $\Delta(\varepsilon, x)$ из условия

$$\Delta(\varepsilon, x) = \min_{S \in N_2} \max_{i \in I_\varepsilon(x)} (g_i(x), S),$$

где $N_2 = \{|S_j| \le 1, j = \overline{1,n}\}$, $g_0(X)$- градиент $F(X)$.

Более подробно алгоритм решения и примеры можно найти в [7, 16].

6.10. Метод проекции градиента

На каждой итерации этого метода предусмотрена процедура возврата очередного приближения градиентного спуска $x^{(k+1)} = x^{(k)} - \alpha_k f'(x^{(k)})$ на допустимое множество U , если $x^{(k+1)} \notin U$. Такой возврат производится посредством проектирования $x^{(k+1)}$ на U, т.е. замены $x^{(k+1)}$ на ближайшую точку множества U.

Определение. Пусть заданы замкнутое множество $U \subset N_n$ и точка $z \in E_n$. Точка $z_U = P_U(z)$ называется *проекцией* точки z на множество U, если

$$\rho(z_U, z) = \min_{x \in U} \rho(x, z)$$

где $\rho(x, y) = \|x - y\|$ – расстояние между точками x и y в пространстве E_n .

Очевидно, для точки $z \in U$ проекция $P_U(z)$ совпадает с z.

Таким образом, в методе проекции градиента последовательные приближения $x^{(k)}$ к точке минимума X^* целевой функции $f(x)$ на множестве U вычисляются по формулам

$$x^{(k+1)} = P_U\left[x^{(k)} - \alpha_k f'(x^{(k)})\right], \; k = 0,1,\ldots, x^{(0)} \in U. \tag{6.20}$$

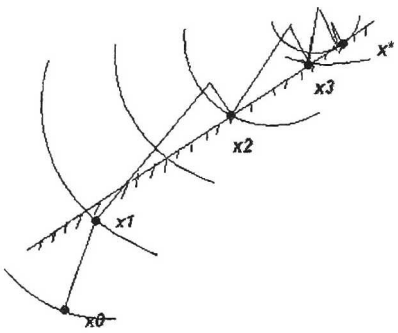


Рис. 43

В зависимости от способа вычисления a_k из (6.20) различают несколько вариантов метода проекции градиента, самыми распространенными из которых являются следующие.

1) a_k находится, как в методе наискорейшего спуска безусловной минимизации, т.е. $\Phi_k(a_k) = \min_{a>0} \Phi_k(a)$, где $\Phi_k(a) = f\left[x^{(k)} - af'(x^{(k)})\right]$.

В предположении, что градиент $f'(k)$ целевой функции удовлетворяет на множестве U условию Липшица, т.е.

$$\|f'(x') - f'(x'')\| \le L\|x' - x''\| \tag{6.21}$$

для всех x' и $x'' \in U$, полагают $a_k \equiv a, k = 0,1,\ldots,$ где a - произвольное число из интервала $(0;2/L)$. Если известна минимальная константа Липшица L из (6.21), то выбирают, как правило, $a=1/L$.

Вычисления по формуле (6.20) завершаются при выполнении одного из неравенств $\|f'(x^{(k)})\| \le \varepsilon$ или $\|x^{(k)} - x^{(k-1)}\| \le \varepsilon$, где величина $\varepsilon > 0$ определяет точность решения задачи. Окончательно полагают $x^* \approx x^{(k)}, f^* \approx f(x^{(k)})$.

Отметим, что определение проекции $P_U(z)$ для точки $z \notin U$ является самостоятельной задачей нелинейного программирования:

$$f(x) = \|x - z\|^2 = \sum_{j=1}^{n}(x_j - z_j)^2 \to min, \quad x \in U \quad , \tag{6.22}$$

решение которой может вызвать затруднения.

В частном случае, когда множество U определяется лишь линейными ограничениями, задача (6.22) представляет собой задачу квадратичного программирования. Ее решение может быть найдено за конечное число шагов, как описано выше.

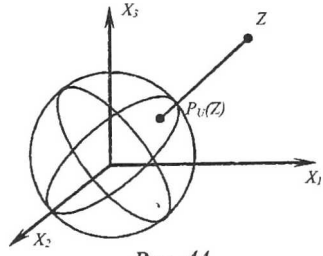


Рис. 44

Особый интерес при использовании метода проекции градиента представляют такие множества U, для которых задача проектирования (6.22) решается в явном виде (U - шар, параллелепипед и т.д.).

Иногда, как в примере с шаром, нахождение кратчайшего расстояния приводит к задаче на условный экстремум:

$$\Phi(X,\lambda)=\sum_{j=1}^{n}(X_j-Z_j)^2+\lambda\sum_{j=1}^{n}(X_j-R_0)^2;\qquad \frac{\partial\Phi}{\partial X_j}=0;\qquad \frac{\partial\Phi}{\partial\lambda_j}=0.$$

В случае с шаром:

$$P_U(Z)=\begin{cases} Z,\ \sum_{j=1}^{n}Z_j^2\le R_0^2;\ (Z\in U),\\ \dfrac{R_0 Z}{\sqrt{\sum_{j=1}^{n}z^2}},\ \sum_{j=1}^{n}Z_j^2>R_0^2;\ (Z\notin U). \end{cases}$$

Если область допустимых решений U параллелепипед, то проекция точки определяется очень просто. Пусть область U определяется условиями: $a_1<x_1<b_1$,

$a_2<x_2<b_2$,

$a_3<x_3<b_3$.

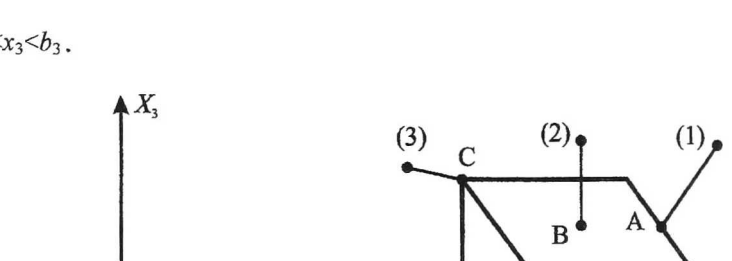


Рис. 45

Если только одна из координат $X_j\in[a_j,b_j]$, а остальные нет, то проекцией на параллелепипед будет точка на ребре (т. *A*).

Если две из координат X_j принадлежат области $X_j\in[a_j,b_j]$, $X_k\in[a_k,b_k]$, то проекцией на параллелепипед будет точка на грани (т. В).

Если все три координаты $X_1\in[a_1,b_1], X_2\in[a_2,b_2], X_3\in[a_3,b_3]$, то проекцией на параллелепипед будет точка в вершине (т. *C*).

6.11. Методы штрафных и барьерных функций

Идея методов штрафных и барьерных функций заключается в том, что исходная задача НЛП с ограничениями заменяется последовательностью задач безусловной оптимизации (т.е. без ограничений) за счет введения новой функции, учитывающей невыполнение ограничений в виде «штрафа»: $C(X)=F(X)+ P(X)$, где $F(X)$ – целевая функция, $P(X)$ – функция, равная в идеале $F(X)$ в области допустимых решений U и равная бесконечности вне этой области. Практически вместо $P(X)$ подбирается последовательность функций $P_K(X)$, зависящих от $к$, так, чтобы $\lim P_K(X)=P(X)$, при $к \to \infty$.

На рис. 46 показано решение задачи $F(x) \to min$, $a<x<b$.

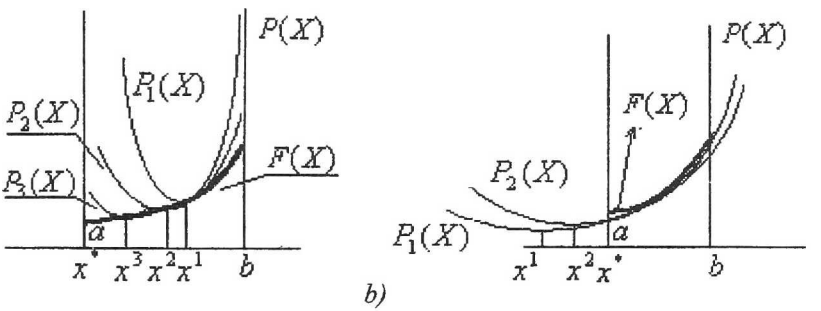


Рис. 46

В методах штрафных функций последовательность точек стремится к решению в допустимой области извне и функции штрафа изменяются вне области U, поэтому они называют методами с внешней точкой (рис. 46а). В методах барьерных функций функция $P(X)$ аппроксимируется барьерными функциями изнутри области (для них ограничения являются барьером), поэтому эти методы называются методами с внутренней точкой (рис. 46b).

Рассмотрим *метод штрафных функций* для задачи НЛП

$F(X) \to \min$, $X \in U$. Заменим эту задачу последовательностью задач безусловной оптимизации

$$C_k(X)=F(X) + P_k(X) \to \min, \quad k=1, 2, 3, \ldots, \tag{6.23}$$

где $P_k(X)$ – функции, которые с ростом k во все большей степени учитывают ограничения, определяемые допустимым множеством U исходной задачи. В методе штрафных функций функции $P_k(X)$ подбираются так, чтобы при больших k функция $C_k(X)$ мало отличалась от $F(X)$ при $X \in U$ и быстро возрастала при удалении точки от допустимого множества U. Последовательностью штрафных функций множества U называется последовательность функций $\{P_k(X)\}$, обладающих свойством

$$\lim P_k(X) = \begin{cases} 0, & если X \in U, \\ +\infty, & если X \notin U. \end{cases}$$

Рассмотрим один из вариантов метода штрафных функций приближенного решения задачи нелинейного программирования

$$F(X) \rightarrow \min, \tag{6.24}$$

$$g_i(X) = 0, i = 1, ..., l,$$
$$g_i(X) \le 0, i = l+1, ..., m,$$

считая, что функции $F(X)$, $g_i(X)$, $i=1,...,m$, заданы во всем пространстве E_n.

Положим

$$P_k(X) = kp(X),\ k=1,2,3,\ldots \tag{6.25}$$

где

$$p(X) = \sum_{i=1}^{} g_i^2(x) + \sum_{i=l+1} [g_i^+(x)]^2,$$

$$g_i^*(X) = \begin{cases} 0, & \text{если } g_i(X) \le 0, \\ g_i(X), & \text{если } g_i(X) > 0. \end{cases}$$

Равенство (6.25) определяет последовательность штрафных функций допустимого множества исходной задачи (проверьте!).

При определенных условиях последовательность решений задач безусловной минимизации (6.23), (6.25) сходится к решению x^* задачи (6.24), поэтому для достаточно больших k полагают $x^*=x^{(k)}$, $F^*=F(X^{(k)})$.

Критерием достижения требуемой точности решения задачи (6.24) может служить неравенство

$$||X^{(k)}-X^{(k/2)}|| \le \varepsilon,$$

где $\varepsilon>0$ – число, характеризующее точность, k – четное число.

Если в задаче (6.24) $F(X)$ – выпуклая квадратичная функция, а $g_i(X)$, $i=1, \ldots, m$, - линейные функции, то точное решение вспомогательной задачи (6.23) можно найти из системы линейных уравнений $dF(X)/dX_j=0$, $j=1, \ldots, n$, определяющих стационарную точку функции $F_k(X)$.

Пример

Минимизировать $-x_1x_2$ при ограничениях

$$g_1 \equiv \quad x_1 + x_2^2 - 1 \le 0,$$
$$g_2 \equiv -x_1 - x_2 \le 0.$$

В качестве штрафной функции применим следующее:

$$p_i(g_i) = \left[(g_i + |g_i|)/2\right]^2, \quad p(X) = \sum p_i\left[g_i(x)\right] \text{ и } P_k(X) = k\, p(X).$$

Тогда

$$C_k(X) = -x_1 x_2 + k\left[\frac{(x_1 + x_2^2 - 1 + |x_1 + x_2^2 - 1|)}{2}\right]^2 + k\left[\frac{(-x_1 - x_2 + |-x_1 - x_2|)}{2}\right]^2$$

будет функцией, минимизируемой методом внешней точки.

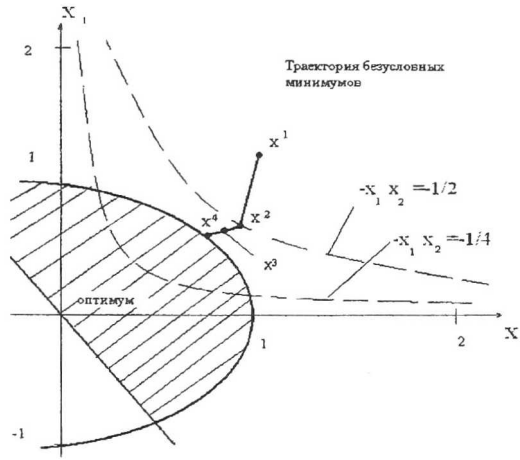


Рис. 47

Значение $x(k)$ для примера

Таблица 39

	k	$x_1(k)$	$x_2(k)$
k_1	1,0	0,89	0,64
k_2	2,0	0,77	0,62
k_3	3,0	0,73	0,61
k_4	10,0	0,67	0,58
оптимум		$2/3$	$\sqrt{3}/3$

В таблице приведены значения $x(k)$ для четырёх различных значений k. Из рисунка видно, что при возрастании k они приближаются к оптимуму из недопустимой области.

Методы внутренней и внешней точки существенно отличаются друг от друга, и не только тем, что в одном из них движение происходит в допустимой области, а в другом - в недопустимой.

Метод барьерных функций

В методе барьерных функций исходная задача НЛП также сводится к последовательности задач безусловной минимизации (6.23), но функции $P_k(X)$ выбираются таким образом, чтобы при больших k функции $C_k(X)$ из (6.23) мало отличались от $F(X)$ во внутренних точках x допустимого множества U. В то же время при приближении точки $X \in U$ к границе множества U эти функции должны неограниченно возрастать.

Определение. Пусть множество $U \subset \varepsilon_n$ задано. Последовательность функций $\{P_k(X)\}$, определенных во всех внутренних точках множества U, называется последовательностью барьерных функций этого множества, если выполняются условия:

1. $\lim_{k\to\infty} P_k(X) = 0$ для любой фиксированной внутренней точки x множества U.
2. $\lim_{r\to\infty} P_k(X^{(r)}) = +\infty$ для любой последовательности $\{X^{(r)}\}$ внутренних точек множества U, сходящейся к какой–либо граничной точке этого множества.

Рассмотрим некоторые варианты метода барьерных функций решения следующей задачи нелинейного программирования:

$$F(X) \rightarrow \min,$$

$$g_i(X) \le 0, i = 1..., m. \tag{6.26}$$

Положим

$$P_k(X) = \frac{1}{k} p(X), \; k=1, 2..., \tag{6.27}$$

где

$$p(X) = \sum_{i=1}^{m} \left| g_{i(x)} \right|^{-k}, \; k > 0, \tag{6.28}$$

или

$$p(X) = -\sum_{i=1}^{m} \ln\left[-g_{i(x)}\right]. \qquad (6.29)$$

Выражения (6.27) – (6.29) определяют последовательности барьерных функций допустимого множества U задачи (6.26) (проверяйте!).

Пусть $X^{(k)}$ – решение задачи безусловной минимизации (6.23), где функция $P_k(X)$ определена равенствами (6.27), (6.28) или (6.29). Полагая $X^* \approx X^{(k)}$, $F^* \approx F(X^{(k)})$ для достаточно большого k, находим приближенное решение задачи нелинейного программирования (6.24) методом барьерных функций. Для контроля достигнутой точности решения можно использовать критерий (6.4).

Пример. Минимизировать $X_1 + X_2$ при ограничениях

$$g_1 = +X_1^2 - X_2 \le 0,$$

$$g_2 = -X_1 \le 0.$$

В качестве функции $p(g_i)$ возьмём логарифмическую функцию. Обозначим $1/k=r$. Тогда $P_r(X))=rp(X)$,

$$p(X) = -\sum_{n=1}^{m} \ln(-g_i(X)) \cdot$$

$$C(X,r) = X_1 + X_2 - r\ln(-X_1^2 + X_2) - r\ln X_1 \cdot$$

Этот простой пример можно решать аналитически, учитывая, что $C(X, r)$ дважды дифференцируема. Необходимые условия первого порядка принимают вид:

$$1 + \frac{r(2X_1)}{-X_1^2 + X_2} - \frac{r}{X_1} = 0,$$

$$1 - \frac{r}{-X_1^2 + X_2} = 0.$$

Решая эту систему, получаем

$$X_1(r) = \frac{-1 \pm \sqrt{1+8r}}{4} \cdot$$

Поскольку значение $X_1(r)$ должно быть положительно, будем рассматривать лишь один корень $X_1(r) = \frac{-1+\sqrt{1+8r}}{4}$. Тогда

$$X_2(r) = \frac{(-1+\sqrt{1+8r})^2}{16} + r \cdot$$

Эти значения $X_1(r)$ и $X_2(r)$ удовлетворяют достаточным условиям и поэтому дают локальный минимум задачи.

В таблице приведены вычисленные значения $X(r)$ для четырёх различных r. Этот пример изображён на рисунке. В пределе при $r_k \to 0$ минимизирующие точки приближаются к решению (0,0).

Значения r и $x(r)$ из примера

Таблица 40

	r	$X_1(r)$	$X_2(r)$
r_1	1,000	0,500	1,250
r_2	0,500	0,309	0,595
r_3	0,250	0,183	0,283
r_4	0,100	0,085	0,107

В этой задаче при каждом значении r существует только *один* локальный минимум. Это происходит так потому, что исходная задача имела единственное решение. Оказывается, что в задачах со многими локальными минимумами (при выполнении вышеупомянутого условия регулярности Слейтера) существует последовательность безусловных локальных минимумов, сходящаяся к каждому из условных локальных минимумов [24,1,12,22].

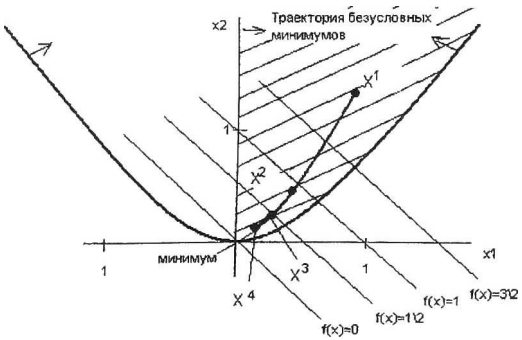


Рис. 48

На рисунке показан пример решения задачи с помощью безусловной логарифмической функции. Допустимая область заштрихована.

6.12. Метод скользящего допуска

Метод скользящего допуска применяется при общей постановке задачи нелинейного программирования:

Минимизировать $F(X)$, $X \in E^n$ при ограничениях

$$h_i(X) = 0, \quad i = 1,....,m,$$
$$g_i(X) \geq 0. \quad i = m+1,...,p.$$

Где функции $F(X)$, $h_i(X), g_i(X)$ могут быть как линейные, так и нелинейные. Алгоритм скользящего допуска позволяет улучшить значения целевой функции как за счет информации, получаемой в допустимых точках пространства решения, так и за счет информации, которую удается получить при прохождения через точки, лежащие вне допустимой области, но близкие к допустимым. Интервалы, в пределах которых точки можно считать почти допустимыми, в ходе поиска постепенно сокращаются, так что в пределе (по мере приближения к искомому решению) учитываются только допустимые точки $\{\overline{X}\}$.

В качестве метода получения последовательности точек, улучшающих значение целевой функции, используется метод Нелдера-Мида (см. п.6.4). При такой стратегии задачу можно заменить более простой, но имеющей то же самое решение задачей минимизации $F(X)$, $X \in E^n$ при ограничениях

$$\Phi^{(\kappa)} - T(X) \geq 0,$$

где $\Phi^{(\kappa)}$ - значение критерия скользящего допуска на κ-м этапе;

$T(X)$ - мера степени нарушения ограничений рассматриваемой задачи.

В качестве Φ выбирают положительно-определенную убывающую функцию координат вершин деформируемого многогранника Нелдера и Мида.

Функция Φ служит критерием допуска для нарушения ограничений задачи и прекращения процедуры оптимизации.

Варианты выбора Φ многочисленны.

Например:

$$\Phi^{(\kappa)} = \min\left\{\Phi^{(\kappa-1)}, \frac{m+1}{r+1}\sum_{i=1}^{r+1}\left\|X_i^{(k)} - X_{r+2}^{(k)}\right\|\right\},$$

$$\Phi^{(0)} = 2(m+1)t,$$

где t – размер исходного многогранника (например, длина высоты);
m – число ограничений равенств;

$X_i^{(k)}$ – вектор i-й вершины многогранника;

R= (n-m) – число степеней свободы целевой функции $F(X)$ в E^n;

$X_{r+2}^{(k)}$ – вектор, задающий «центр тяжести» многогранника при r =n.

По мере приближения к оптимальной точке $\Phi^{(\kappa)}$ устремляется к нулю «сверху».

В пределе многогранник деформируется в точку

$$\lim_{x\to x^*}\Phi^{(\kappa)} = 0.$$

Функционал

$$T(X) = \left[\sum_{i=1}^{m} h_i^2(X) + \sum_{i=m+1}^{p} U_i g_i^2(X)\right]^{1/2},$$

где $U_i = 0$ при $g_i(X) \geq 0$, и $U_i = 1$ при $g_i(X) < 0$,

$T(X) \geq 0$ при $\forall X \in E^n$.

Если $T(X^{(\kappa)}) = 0$, то $X^{(\kappa)}$ – допустима; если $T(X^{(\kappa)}) > 0$, то $X^{(\kappa)}$ – не допустима при $T(X^{(\kappa)}) > \Phi^{(\kappa)}$; и $X^{(\kappa)}$ почти допустима при $0 \leq T(X^{(\kappa)}) \leq \Phi^{(\kappa)}$.

Это условие и определяет набор точек по допустимости.

Общая схема алгоритма выглядит так: при заданном $\Phi^{(\kappa)}$ в точке $X^{(\kappa+1)}$ может быть два варианта:

1) $T\left(X^{(\kappa+1)}\right) > \Phi^{(\kappa)}$ – перемещение считается разрешенным.

2) $T\left(X^{(\kappa+1)}\right) > \Phi^{(\kappa)}$ – необходимо найти другую точку $X^{(\kappa+1)}$, ближе к допустимой области, чтобы $T(X^{(\kappa+1)})$ уменьшилась. Для этого можно в ал-

горитме Нелдера-Мида считать точку X^{k+1} худшей, или же возвращать ее на ближайшую границу.

T(X) не влияет на сходимость алгоритма, так как на заключительных этапах поиска точки $X_i^{(\kappa)}$ являются внутренними , т.е. $T(X_i^{(\kappa)}) = 0$.

Если X^* является граничной точкой, то поиск вдоль границы происходит за счет улучшения $F(X)$ при условии

$$\Phi^{(\kappa)} - T(X_i^{(\kappa)}) \geq 0.$$

Степень нарушения ограничений по мере приближения к искомому решению уменьшается.

К моменту прекращения оптимизационного поиска выполняются условия:

$$\Phi^{(\kappa)} \leq \varepsilon,$$

$$F(X_l^{(\kappa)}) \leq F(X^* \pm \varepsilon),$$

$$\varepsilon - T(X_l^{(\kappa)}) \geq 0.$$

7. ОСОБЕННОСТИ СОВРЕМЕННОЙ ТЕОРИИ ПРИНЯТИЯ ОПТИМАЛЬНЫХ РЕШЕНИЙ

Методы поиска оптимальных решений рассматриваются в разделах математики (методах оптимизации), исследования операций, математическом программировании и в классическом понимании связаны с поиском экстремумов функций и функционалов при функциональных ограничениях на переменные. Решение здесь - математический объект. Причём здесь вопросы оценки критериев (многих) и изучение того, почему такой-то критерий можно считать приемлемым не изучается (о критериях не спорят!).

На практике задача выбора решения намного сложнее.

1. В некоторых задачах сами варианты проведения операций представляют собой небольшое число отдельных альтернатив, не всегда однородных. Например: вариант 1 - строить большой завод, вариант 2- строить небольшой завод, а потом расширять производство. Поэтому в теории принятия решений (ТПР) рассматривается как самостоятельный вопрос задания множества альтернатив, на котором ищется оптимальное решение.
2. Во многих задачах множество альтернатив не ясно с самого начала, например, альтернативы при игре в шахматы. Поэтому в ТПР говорят о генерации альтернатив в процессе решения задачи. При игре в шахматы приходится генерировать альтернативы в процессе игры и оценивать их.
3. Оценка ценности каждой альтернативы во многих случаях не сводится к простому сравнению количественного параметра (сравнению чисел). В связи с этим в ТПР применяют специальные методы измерения полезности альтернатив.
4. В большом числе практических задач системы или операции не могут быть оценены единственным показателем эффективности. Сами системы являются очень часто многоцелевыми. Вот почему критерий не может быть скалярной величиной, а является вектором. Оказывается, что когда по одному показателю один вариант лучше, по другому он может оказаться хуже, и поэтому возникает проблема их сравнения. Этот раздел современной ТПР называют многокритериальной или векторной оптимизацией.
5. Во многих задачах, особенно в задачах автоматизации управления, присутствуют так называемые нечёткие переменные, и нечеткие критерии. Появляется специальный раздел ТПР – принятия решений в расплывчатых ситуациях, в котором рассматриваются вопросы «человекоподобного» восприятия информации для принятия решения и создаются автоматизированные экспертные системы поддержки принятия решений.
6. Иногда встречаются задачи, в которых присутствуют несколько сторон, преследующих разные интересы (боевые действия, поведение государств, конкурирующие фирмы), которые принимают решения в одной и

той же системе. Такие ситуации называются конфликтными. Модель для принятия решений и принцип оптимальности в этих случаях является совершенно иными, чем в рассмотренных задачах математического программирования. Этими вопросами занимается теория игр.

7. На практике одни решения приводят к необходимости принимать следующие решения и т.д., т.е. приходится изучать многоэтапные решения, в том числе и в условиях случайных возмущений.
8. При принятии решений в условиях неопределенности можно поставить и решить задачу устранения части неопределенности за счет проведения «эксперимента». Например, это выражается в виде разведки (военной или экономической), метеонаблюдений, радиолокации и т.п.
9. Во многих задачах принимаются так называемые групповые решения, при этом альтернативы оцениваются группой экспертов – специалистов в данной предметной области. На основе такой экспертизы путем специальной обработки экспертных данных можно принимать решения в случаях, когда альтернативы трудно формализуемы. Такой метод получил название метода экспертных оценок.
10. Вот почему современная теория принятия решений представляет собой обобщение всей методологии получения решений в самых общих и практически важных случаях (много критериев; неопределённость принципа оптимальности, нечёткость шкалы ценности, условия риска, неповторимость ситуации, сложная логика предпочтений между альтернативами и другое).

Сказанное характерно для анализа оптимальных решений при построении современных автоматизированных систем управления и экспертных систем.

7.1. Общая постановка задачи принятия решения

Задачей принятия решения назовём пару (Ω, P) , где Ω – множество вариантов, P – принцип оптимальности. Решением задачи (Ω, P) является множество $\Omega_p \leq \Omega$, полученное с помощью принципа оптимальности P. Отсутствие хотя бы одного из элементов лишает смысла задачи в целом. Если нет Ω не из чего выделять Ω_p. Если нет P, то найти Ω_p невозможно. Математическим выражением принципа оптимальности P служит функция выбора C_p. Она сопоставляет любому подмножеству $X \leq \Omega$ его часть $C_p(X)$. Решением Ω_p исходной задачи и является множество $C_p(X)$.

Задачи принятия решений различаются в зависимости от имеющейся информации о множестве Ω и принципе оптимальности P. В общей задаче принятия решений как Ω, так и P могут быть неизвестными.

Информацию, необходимую для выделения Ω_p получают в процессе решения. Задачу с известными Ω называют задачей выбора, а задачу с известными Ω из P - общей задачей оптимизации.

Таким образом, задача выбора и задача оптимизации являются частными случаями общей задачи принятия решений.

В общем случае для задачи выбора необязательно необходимо полное восстановление принципа оптимальности, а можно ограничиться только информацией, достаточной для выделения Ω_p. Общая задача оптимизации не обязательно требует минимизации одной или нескольких числовых функций, так как может быть и так, что другим способом выделяется множество лучших элементов, то есть выделяются значения $C_p(\Omega)$ при заданных Ω и C_p. Если C_p - скалярная функция на множестве Ω, то получаем обычную оптимизационную задачу.

Элементы множества Ω называются альтернативами или вариантами. Принцип оптимальности P задаёт понятие лучших альтернатив: лучшими считают альтернативы, принадлежащие $C_p(\Omega)$. В практических задачах альтернативы обладают многими свойствами, оказывающими влияние на решение.

Пусть некоторое свойство альтернатив из Ω выражается числом, то есть существует отображение φ: $\Omega \rightarrow E_1$. Тогда такое свойство называется критерием, а число $\varphi(x)$ – оценка альтернативы x по критерию. Одновременный учёт отдельных свойств может быть затруднительным. При этом можно выделить группы свойств, которые агрегируют в виде аспектов. Аспект – сложное свойство альтернатив, которое одновременно учитывает все свойства, входящих в группу. В частном случае аспект может являться критерием.

Пусть все свойства $k_1, \ldots, k_m$, учитываемые при решении задач (Ω, P) являются критериями. Критериальным пространством называют пространство E_m,координаты которого есть оценки соответствующих критериев.

Пример: при определении маршрута перевозок альтернативы-маршруты. Диспетчер учитывает свойства: протяженность, загрузка, энергоёмкость, безопасность, техническое обслуживание и другое.

Техническое обслуживание – это число станций, сроки выполнения ремонтных работ и так далее. Следовательно, техническое обслуживание - это аспект.

Стоимость маршрута состоит из стоимости топлива, обслуживания и т.д. То есть это тоже аспект. Однако, так как стоимость можно вычислить, то это критерий (но так будет не всегда).

Сформулируем схему процесса принятия решения.

Формируется множество Ω, его подготавливают для последующего решения задачи выбора (при назначении на должность сначала готовят список кандидатов, а затем назначают лицо из этого списка). Для формирования Ω используют условия возможности и допустимости альтернатив, которые определяют конкретными ограничениями задачи. При этом считают известным универсальное множество альтернатив Ω_y. Таким образом, задача формирования Ω является задачей выбора (Ω_y, P_1) , где P_1 – принцип оптимальности, выражающий условия допустимости.

Множество $\Omega=C_{p1}(\Omega_y)$, полученное в результате решения, называют исходным множеством альтернатив (ИМА) (Ω_y – все специалисты, Ω - специалисты, удовлетворяющие требованиям).

Таким образом, общая задача принятия решения сводится к решению двух последовательных задач выбора.

В процессе решения участвует лицо, принимающее решение (ЛПР), эксперты, консультанты.

ЛПР – человек, имеющий цель, служит мотивом постановки задачи. ЛПР имеет полномочия и несёт ответственность. Основная функция ЛПР выделять Ω_p. ЛПР даёт информацию о принципе оптимальности Р.

Эксперт - не несёт ответственности, он даёт оценки альтернатив, необходимые для формирования ИМА и решения задачи выбора.

Консультант – специалист по теории выбора и ПР. Он разрабатывает модель, процедуру принятия решений , организует работу ЛПР и экспертов.

В простейшем случае задачу (Ω, Р) решает непосредственно ЛПР.

Прикладные результаты ТПР имеют вид алгоритмов решения задач. Часть алгоритмов может быть реализована вручную, но основная часть ориентирована на диалоговый режим работы ЭВМ или на применение в экспертных системах.

7.2.Классификация задач принятия решений

Для классификации задач принятия решений представим более подробно задачу ПР в виде семёрки:

$$\text{ПР} = (Z, \Omega, K, S, F, G, P).$$

Z - Описание предметной области, в которой принимается решение.

Ω - Множество альтернатив.

K - Множество критериев оценки альтернатив.

S - Множество шкал измерения по критериям.

F - Отображение множества альтернатив в множество критериев оценок $F(X) = Y$.

G - Система предпочтений.

P - Решающее правило, отображающее систему предпочтений.

Любой из элементов этой семёрки может служить классификационным признаком задач ПР. Однако в основном для классификации используются

тип отображения F, мощность множества K, тип системы предпочтений G и характеристики области допустимых решений $X \in \Omega$.

Отображение X в множество критериальных оценок может быть детерминированным (рис. 49), вероятностным (рис. 50) либо неопределённым (рис. 51). Поэтому задачи принятия решений можно разделить на ПР в условиях определённости, риска и неопределённости.

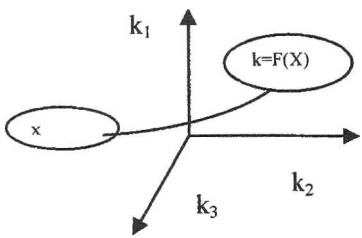


Рис. 49

ПР в условиях определённости предполагает детерминированное (взаимно однозначное) соответствие между X и k.

ПР в условиях риска соответствует вероятностному отображению X в k.

ПР в условиях неопределённости не предполагает соответствия X и k.

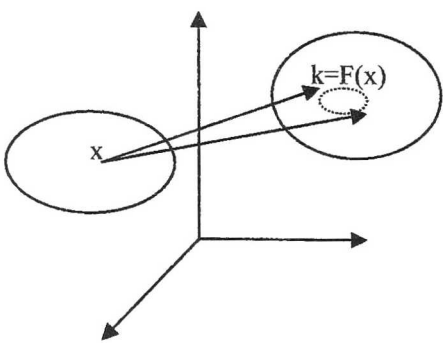


Рис. 50

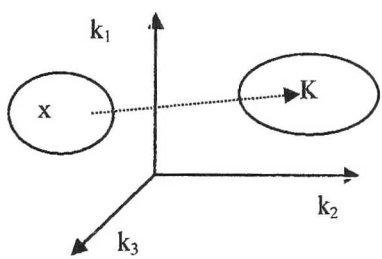


Рис. 51

Если k – скаляр, то это однокритериальная задача ПР. Если k – вектор, то это задача многокритериальной (векторной) оптимизации.

Отметим, что задача ПР со скалярным критерием является тривиальной задачей оптимизации в случае детерминированного отображения F. Как только решающий элемент имеет дело с ПР при вероятностном характере отображения F, либо априори неизвестном (неопределённом) F, то задача ПР даже со скалярным критерием сразу перестаёт быть тривиальной (см. рисунок 52):

а) выбор при одном критерии в условиях определённости;

б) выбор при одном критерии в условиях риска.

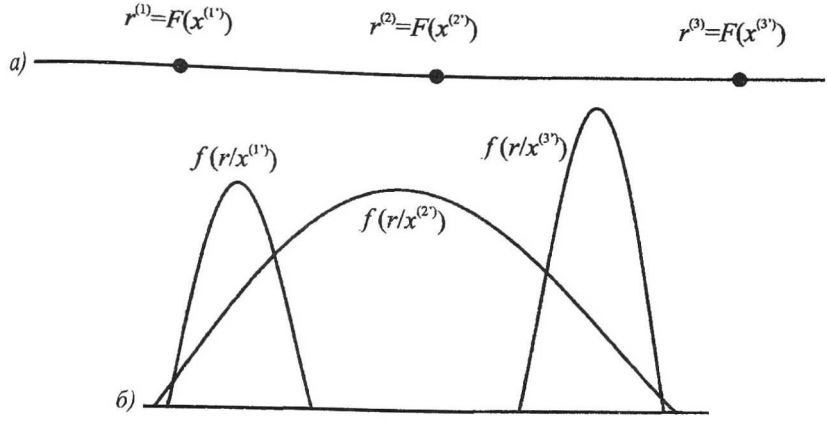


Рис. 52

7.3. Многокритериальная оптимизация

В главах 1-6 мы рассмотрели задачи, в которых требуется выбрать решение, доставляющее максимум (или минимум) единственного показателя эффективности (критерия) k. На практике часто возникает случай, когда эффективность операции приходится оценивать не по одному, а сразу по нескольким показателям $k_1, k_2, \ldots, k_L$.

Примеры:

1. Оценка деятельности завода: прибыль, средняя зарплата, объем производственных фондов.
2. Оценка учебы студента: оценки по предметам.
3. Военная операция: потери, вероятность выполнения задачи.

Одни из этих показателей необходимо сделать больше, другие меньше. Как правило, эффективность больших по объему, сложных операций, а также сложных многоцелевых систем не может быть исчерпывающим образом охарактеризована с помощью одного показателя: на помощь ему приходится привлекать и другие, дополнительные показатели.

Главной особенностью этой ситуации является то, что требования ко всем показателям в реальных системах несовместимы или противоречивы. Как правило, требование max k_1 не обращает ни в максимум, ни в минимум другие показатели k_2, k_3, ... Поэтому широко распространённая формулировка «достижения максимального эффекта при минимальных затратах» является некорректной. Корректными являются следующие формулировки:

1. Достижение максимального эффекта при заданных затратах.
2. Достижение заданного эффекта при минимальных затратах.

Для удобства сравнения значений векторов $k=(k_1, k_2, \ldots k_L)$ – иногда удобно предварительно привести все показатели $k_1, k_2 \ldots k_L$ к стандартному виду, чтобы все критерии минимизировались и чтобы они имели безразмер-

ный масштаб измерения. Векторный критерий k считается стандартным, если он удовлетворяет условию $k_i \geq 0$, и чем меньше k_i, тем лучше операция (система). Таким образом, идеальной является система, у которой $k_i=0$. Нестандартный показатель можно привести к стандартному.

Если $k_i \rightarrow$ max, то

$$k^{ст}{}_i = k^{max}{}_i - k_i.$$

Если $k^{max}{}_i = \infty$, то $k^{ст}{}_i = 1 / k_i$. Стандартный критерий можно задать в виде отношения: $k^{ст}{}_i = (1 - k_i / k^{и}{}_i)$, где $k^{и}{}_i$ – идеальное значение критерия (либо определенное значение, либо max k_i).

При этом в L-мерном пространстве задаётся вектор k, каждая компонента которого изменяется от 0 до 1. Полностью идеальной системе соответствует $k=0$.

Прежде всего, анализ векторов, соответствующих альтернативам из области допустимых альтернатив, позволяет заранее отбросить явно нерациональные варианты решений, уступающие лучшим вариантам по всем показателям. Этот этап векторной оптимизации называется безусловной оптимизацией.

Пусть анализируется боевая операция, оцениваемая по двум показателям:

k_1 – вероятность выполнения боевой задачи;

k_2 – стоимость израсходованных средств.

Очевидно, первый из показателей желательно обратить в максимум, а второй в минимум. Пусть предлагается на выбор конечное число различных вариантов решения, обозначим их X_1, X_2, ..., Xn. Для каждого известны значения k_1, k_2 . Изобразим каждый вариант решения в виде точки на плоскости с координатами k_1, k_2. Когда X пробегают ОДР, точки k_1, k_2 заполняют критериальное пространство решений.

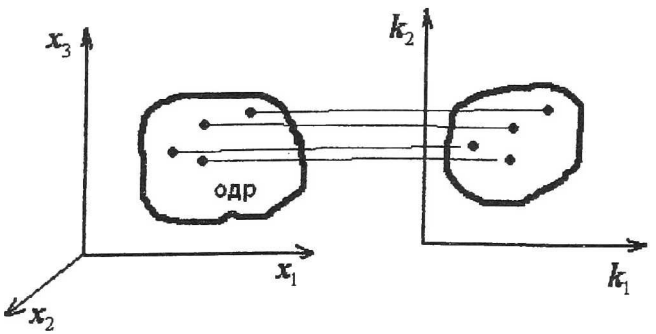


Рис. 53

Итак, если некоторая операция оценивается несколькими критериями, каждое из которых число, то такая задача называется задачей многокритериальной или векторной оптимизации. Для стандартных k эта задача имеет вид

$$k=(k_1, k_2, \ldots, k_L) = \overline{f}(X, A) \rightarrow \min, \tag{7.1}$$

$$\varphi_i(X, A) \le 0, i = 1...m. \tag{7.2}$$

Существует принципиальная трудность в объективной (безусловной) оценке альтернатив при двух или более критериях. Она связана с проблемой сравнения двух векторов.

Безусловным критерием предпочтения называют критерий, основанный на сравнении компонент вектора. Два вектора критериев k^A и k^B безусловно сравнимы , если для любой l-й компоненты вектора выполняются неравенства

$$k_l^A >= k_l^B \ , \quad l=1, 2, 3, \ldots, L.$$

Если все $k_l \rightarrow \min$, то альтернатива В безусловно предпочтительнее (лучше) А. Это означает, что А никак не может быть оптимальной и поэтому должна быть отброшена. Если же знаки неравенств для различных компонент вектора различны, то такие альтернативы безусловно несравнимы. Рассмотрим пример.

Необходимо выбрать лучших по успеваемости студентов А, Б, В, Г, Д. Критериями служат оценки по предметам M, N, O, P.

	M	N	O	P
А	4	4	5	3
Б	5	5	4	3
В	4	5	3	3
Г	5	4	5	4
Д	4	5	4	3

Проведем безусловное сравнение векторов оценок. Студент А не сравним с Б. В безусловно хуже Б – отбросим В. Д безусловно хуже Б – отбросим Д. А безусловно хуже Г – отбросим А. Остались две альтернативы Б и Г, которые безусловно несравнимы.

Определение. Множество безусловно несравнимых альтернатив, оставшихся после отбрасывания всех безусловно худших альтернатив, называется множеством Парето. Парето-оптимальное множество еще называют областью компромиссов. Ясно, что множество Парето можно получить в результате анализа критериального пространства.

Таким образом векторная оптимизация включает два этапа.

1. Безусловная оптимизация. Здесь анализируется критериальное пространство, отсеиваются безусловно худшие варианты и получают множество Парето.

2. Условная оптимизация. Так как множество Парето, как правило, состоит из более чем одной точки, то для получения единственного решения необходимо применить дополнительные принципы оптимальности (условия согласования критериев).

7.4. Определение множества Парето

Если целевая функция векторная функция векторного аргумента и каждому малому изменению переменных X соответствует малое изменение каждого критерия, то критериальное пространство – это область с внутренними точками и границей (не обязательно замкнутая). Пусть есть только два критерия (k_1, k_2) и $k_1 \rightarrow$ max; $k_2 \rightarrow$ max.

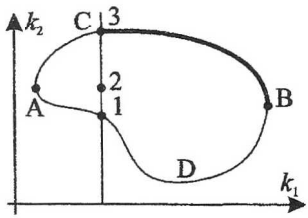


Рис. 54

Рассмотрим область возможных значений критериев (рис. 54). Ясно, что точки, лежащих на одной вертикали или на одной горизонтали всегда, безусловно, сравнимы (точки 1, 2, 3). Получается, что все внутренние точки можно отсеять. Дуга AB состоит из лучших точек по критерию k_2 при постоянном k_1.

Но часть точек дуги AB , а именно точки дуги AC можно отбросить, так как они хуже точек CB. Поэтому точками множества Парето будут являться точки, находящиеся на дуге CB. При анализе характерными являются точки касания вертикалей и горизонталей к области критериев (точки A, B, C, D).

Существует несколько алгоритмов нахождения множества Парето. Они зависят от того, в каком виде задано множество вариантов критериев. В практических задачах сначала приходится получать само множество критериев, затем получать множество Парето в критериальном пространстве и только затем можно определить Парето-оптимальное множество в области допустимых решений X.

Существуют два основных метода нахождения множества Парето:

1. Метод обхода конусом. Этот метод применяется для непрерывной области критериев. Назовем конусом пространственный угол, образуемый лучами, идущими из общей вершины и ограниченными в каждой плоскости углом 90°. Направление ограничивающих лучей соответствует направлению оптимизации критериев.

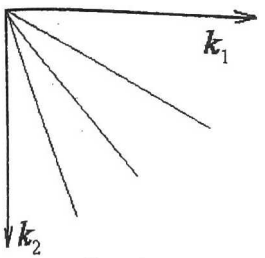


Рис. 55

Пусть для двух критериев: $k_1 \rightarrow$ max (↑) и $k_2 \rightarrow$ min (↓), тогда конус имеет вид, изображенный на рисунке 55.

Для получения множества Парето достаточно установить вершину конуса на все точки границы области критериев, если при этом ни один луч не пересекает область, то вершина лежит в точке Парето, а если пересекает, то нет.

Пример: $k_1 \rightarrow$ min, $k_2 \rightarrow$ min.

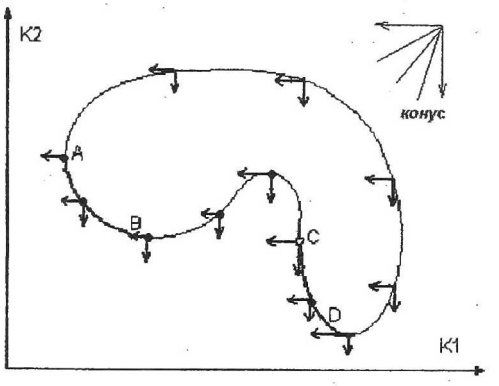


Рис. 56

Множество Парето: [AB]∪ (CD].

2. Метод прямоугольников. Этот метод применяют, когда критериальное пространство представляет собой отдельные точки или табличные значения. Сформулируем алгоритм для случая двух критериев, когда $k_1 \rightarrow$min и $k_2 \rightarrow$min, а критериальное пространство – точки на плоскости.

1) Фиксируем самые левые точки, если их несколько, выбираем среди них самую нижнюю. Эта точка является точкой Парето, фиксируем ее.

2) Выберем самую нижнюю точку, если их несколько, то выбираем самую левую. Это точка Парето, фиксируем ее.

3) Через полученные точки проводим вертикаль и горизонталь. Отбрасываем сами точки Парето, точки лежащие на границе полученного прямоугольника и точки вне прямоугольника.

4) К внутренним точкам полученного прямоугольника применяем алгоритм из пункта 1.

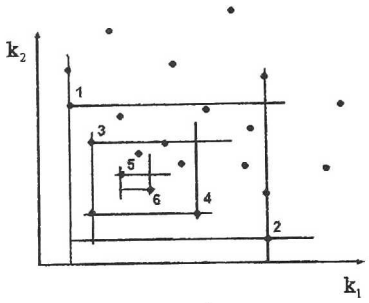


Рис. 57

Алгоритм заканчивается тогда, когда внутри прямоугольника не останется ни одной точки.

Множество Парето:

1, 2, 3, 4, 5, 6.

Вышеизложенный алгоритм легко обобщается на случай большего числа критериев, при других направлениях оптимизации, а также на случай табличного задания критериев.

7.5. Методы условной многокритериальной оптимизации

После нахождения множества Парето, если количество точек в нём ≥2, встаёт вопрос о выборе единственного решения. Так как точки на множестве Парето обладают таким свойством, что каждая из них лучше по одному критерию, но хуже по другому, то объективно предпочесть эти точки невозможно. Условная оптимизация предполагает введение условия согласованности в компонентах критерия, которое всегда является хотя и разумным, но субъективным. Подобные условия еще называют схемой компромиссов. Рассмотрим несколько простейших из них.

Принцип равенства критериев. Рассматривают множество Парето для стандартных критериев. Тогда оптимальной считается та точка множеств Парето, где $k_1=k_2=\ldots=k_L$ (см. рисунок 58).

Принцип равного влияния. Здесь оптимальной считается точка множества Парето, являющаяся точкой касания прямой $k_1+k_2 \rightarrow \min$ и области Парето (точка C, см. рисунок 59).

Принцип минимакса. Оптимальной считается точка множества Парето, где обеспечивается $\min\max k_i^{ст}$ (см. рисунок 59).

Кроме указанных простейших приемов применяются более сложные.

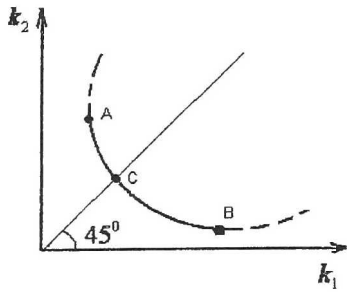


Рис. 58

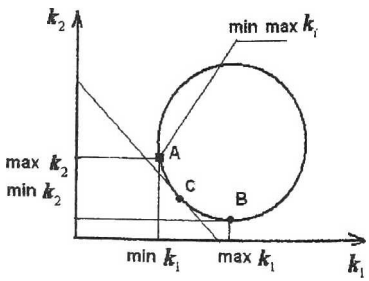


Рис. 59

1) Метод скаляризации. Здесь формируется некоторая скалярная функция многих переменных:

$$K_0 = f(K_1, K_2, \ldots, K_L).$$

K_0 - это скалярная величина, которая обобщает все критерии. Наиболее распространённой разновидностью функции f является линейная функция

$$K_0 = \sum_{i=1}^{L} \alpha_i K_i \to \min(\max).$$

a_i -вес каждой компоненты критерия, чем эта величина больше, тем большее влияние оказывает этот критерий. Для нестандартных критериев веса могут иметь размерность, причем если $K_i \to \max$, то $a_i \geq 0$, а когда $K_i \to \min$, то $a_i \leq 0$. Подобная комплексная оценка зачастую затруднительна. На практике часто используют и другие способы искусственного объединения нескольких показателей в один обобщённый показатель (критерий). В качестве обобщённого критерия, например, берут дробь:

$$K_0 = (k_1 \times k_2 \times \ldots \times k_m) / (k_{m+1} \times \ldots \times k_L),$$

где $k_1 \ldots k_m$ – увеличиваются,

$k_{m+1} \ldots k_L$ - уменьшаются.

Вольное использование подобных критериев чревато опасностями и может привести к неправильным рекомендациям.

Общим недостатком «составных критериев» является то, что недостаток эффективности по одному показателю всегда можно скомпенсировать за счёт другого (например, малую вероятность выполнения боевой задачи – за счёт малого расхода боеприпасов и т.п.).

На этот счет известна шутка Л.Н.Толстого, что если представить оценку человека в виде дроби:

$\dfrac{\textit{Достоинство человека}}{\textit{Мнение о себе}}$, то можно получить самую высокую оценку в случае $\dfrac{\textit{Мало достоинств}}{\textit{Нет мнения о себе}} = \infty$.

2) Метод главного критерия. В этом методе сначала выбирают главный критерий и объявляют его единственным критерием. Все остальные критерии становятся управляемыми переменными, на которые накладываются ограничения, что бы они были не хуже заданной величины k_i^0. Например, если все $k_i \to \min$ и главный k_1, то получаем однокритериальную задачу математического программирования.

$$k_1 = f(X) \to \min,$$

$\Phi_i(X) < 0, \quad i = 1,2,3,...m,$

$k_2(X) < k_2^0$, $k_3(X) < k_3^0$, ... , $k_L(X) < k_L^0$.

Тогда, эти ограничения войдут в комплекс заданных условий.

Например, в промышленности можно потребовать, что бы прибыль → max, план по ассортименту – выполнен, а себестоимость – не выше заданной.

При бомбардировке – понесённый ущерб противника максимальный, потери и стоимость операции не выходили за известные пределы.

Здесь все показатели эффективности, кроме одного, переводятся в разряд ограничений. Варианты решения, не укладывающиеся в заданные границы, сразу же отбрасываются как недопустимые.

Такой метод чаще всего используется при оптимизации сложных технических систем (самолетов, автомобилей, бытовой техники и т.д.).

3). Метод последовательных уступок. Проранжируем все критерии и расположим их в порядке убывающей важности: $k_1, k_2 \ldots k_L$. Пусть все критерии нужно максимизировать. Сначала отбросим все критерии, кроме k_1, и найдем допустимое решение, обращающее в максимум k_1. Как правило при этом другие критерии не получают своих наилучших значений. Поэтому исходя из практических соображений и точности, с какой известны исходные данные, назначается уступка Δk_1, которую мы согласны допустить для того, чтобы обратить в максимум k_2. Наложим на k_1 ограничение, чтобы он был не меньше чем $k_1^* - \Delta k_1$ (k_1^* - максимально возможное значение k_1), и при этом ограничении ищем решение, обращающее в максимум k_2. Далее снова назначается уступка в показателе k_2, ценой которой находится максимум k_3 и т.п.

$$k_2 \to \max,$$
$$k_{1\max} - \Delta k_1 < k_1 < k_{1\max}.$$

На втором этапе решаем задачу:

$$k_3 \to \max,$$
$$k_{1\max} - \Delta k_1 < k_1 < k_{1\max},$$
$$k_{2\max} - \Delta k_2 < k_2 \le k_{2\max}.$$

И так далее.

Такой способ хорош тем, что здесь сразу видно, ценой какой «уступки» в одном показателе приобретается выигрыш в другом. Заметим, что иногда даже при малом Δk «свобода» выбора позволяет достичь существенной оптимизации k_2. Это зависит от вида границы критериального пространства (см. рисунок 60).

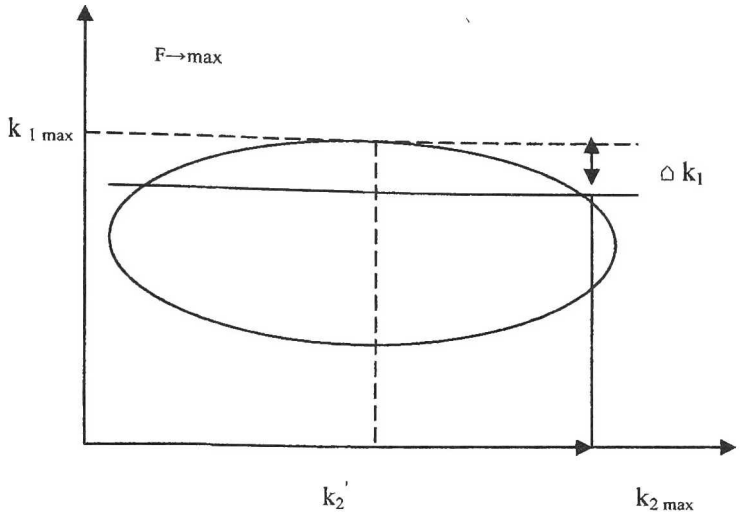


Рис. 60

Очевидно, успех такого метода зависит от того, насколько «тупой» максимум. Процедура может быть повторена для других ΔK.

4. Метод последовательной оптимизации. В некоторых случаях критерии системы не слишком связаны друг с другом, т.е. улучшение одного критерия, по крайней мере в ограниченных пределах, не «мешает» другому. Это характерно для сложных систем, состоящих из отдельных достаточно автономных подсистем (например: процессор компьютера, система электропитания, система охлаждения, система отображения), когда отдельные критерии относятся к этим подсистемам. Расположим все критерии в порядке убывания важности. Сначала оптимизируют систему по важнейшему показателю, отбросив все остальные, затем по второму, и т. д.

Следует отметить, что в вышеприведенных методах используется процедура упорядочения критериев по важности. Эта самостоятельная проблема является достаточно сложной и поэтому решается на основе методов экспертных оценок, изложенных в 10-й главе.

8. ИГРОВЫЕ МОДЕЛИ ПРИНЯТИЯ РЕШЕНИЙ

Математические модели, рассмотренные нами, выражают стремление получить максимальный результат для какой-то одной системы (фирмы, отрасли, страны). Если эта система состоит из каких-то других элементов (подсистем), то считается, что их целевые функции совпадают, то есть эти подсистемы работают «все как один». Во многих задачах это не так. Приходится анализировать ситуации, когда присутствует несколько сторон, преследующих различные интересы (цели). Это и боевые действия, и внешняя экономическая деятельность государств и многое другое. При переходе к рыночной экономике и возникновении конкуренции как основного рычага развития экономики постановка задачи с отождествлением целей является неполной, а иногда и неправильной. Интересы отдельных фирм сталкиваются на одном рыночном пространстве и иногда настолько различны, что конкуренция приобретает форму борьбы, и поэтому невозможно оценить результат принимаемого решения единообразно. Такого рода ситуации называются конфликтными и могут быть описаны моделями, которые называются играми. Модели конфликтных ситуаций, где присутствуют несколько сторон, преследующих различные интересы, называются игровыми моделями. Игровая модель представляет собой особый вид модели для принятия оптимальных решений. Теория, описывающая конфликтные ситуации с количественной стороны, называется теорией игр.

8.1. Основные понятия теории игр

Игра – математическая модель ситуации, некоторая упрощённая схема, где зафиксированы все участвующие стороны, правила развития данной ситуации, определённые выигрыши после каждого хода, правила окончания игры. Из этого следует, что основными в игровой модели являются следующие элементы:

- в игре могут быть две или более стороны, называемые игроками, которые преследуют различные интересы;
- в игре фиксируют правила игры: возможные действия игроков, ситуация выигрыша и его величина, правила остановки игры;
- игры бывают парные, когда есть только две стороны, и множественные, когда участие в игре принимают три и более сторон;
- игры бывают коалиционные, когда часть игроков соединяют свои интересы и действуют как один игрок.

Мы рассмотрим только парные игры как наиболее важные для моделирования реальных явлений. Результаты анализа таких моделей позволяют сформулировать принципы и подходы к выработке оптимальных решений в условиях конкуренции.

В парной игре два игрока: А – это «мы» и В – «противник». Игры называются антагонистическими, или с противоположными интересами, если игрок А выигрывает ровно столько, сколько проигрывает В. В сумме выигрыш равен 0, поэтому такую игру называют игрой с нулевой суммой. Но существуют также парные игры и с ненулевой суммой (неантагонистические).

Ходы в игре могут быть личные и случайные. Личный ход зависит от сознательного решения стороны, а случайный ход - результат случайного механизма, который иногда применяется специально, а иногда случайно вовлекается в игру. В ходе проведения игры каждый игрок имеет какие-то альтернативы поведения, которые называют стратегиями игроков. В общем случае стратегия игрока – это совокупность правил, определяющих выбор вариантов действий при каждом личном ходе игрока в зависимости от сложившейся ситуации. Применение в каждой игре стратегии однозначно определяет исход игры. Если количество стратегий конечно - игра конечная, в противном случае - бесконечная игра. В теории игр считается, что игра повторяется многократно и игроков интересует средний выигрыш. Задачей теории игр является обоснование оптимальных стратегий обоих игроков.

8.2. Платежная матрица антагонистической игры, принцип минимакса

Будем считать, что у игрока А всего m стратегий, а у игрока В – n стратегий.

$A_1 A_2 \ldots\ldots A_m$,

$B_1 B_2 \ldots\ldots B_n$.

Таблица 41

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

Если игрок А применяет свою стратегию A_i, а В применяет B_j, то выигрыш в игре составляет a_{ij}. А старается увеличивать выигрыш, а В – уменьшить эту величину. Величины a_{ij} образуют матрицу, которая называется платежной матрицей .

Рассмотрим пример простейшей игры «Поиск».

Игрок А прячется в двух местах I и II. Игрок В ищет игрока А в местах I и II. Если В находит А, то А платит ему 1 рубль, если не находит, то В платит 1 рубль игроку А.

Платежная матрица – Таблица 42.

Таблица 42

	B_1	B_2
A_1	-1	1
A_2	1	-1

Основой для анализа игр является анализ платежной матрицы. Поэтому игры, для которых могут быть построены платежные матрицы, называются матричными играми.

Для игр с полной информацией, т. е. когда один игрок знает, как поступил второй, всегда можно построить платёжную матрицу.

Основным принципом, которым необходимо пользоваться для объективной оценки игровой ситуации, является следующий подход: противник настолько же разумен, как и мы, и делает все, чтобы использовать свои возможности против нас, т. е. активно противодействует в выборе оптимального решения. Задачей теории игр является обоснование оптимальных стратегий обоих игроков.

Если игрок А будет применять любое сколь угодно сложное правило чередования мест I и II, то В все равно разгадает этот закон чередования и будет выигрывать. Ниже мы обоснуем оптимальные стратегии игроков А и В. Рассмотрим другие простые примеры игр.

Игра «Три пальца».

Таблица 43

	B_1	B_2	B_3
A_1	2	-3	4
A_2	-3	4	-5
A_3	4	-5	6

Игрок А и игрок В одновременно показывают один, два или три пальца. Выигрыш равен сумме, причём выигрывает А, если сумма очков четная, и В, если сумма нечетная. Построим платёжную матрицу.

Игра «Конкуренты».

Таблица 44

	B_1	B_2	B_3
A_1	-2	-2	1
A_2	1	0	2
A_3	0	-1	1

Фирма А может поставить на рынок три товара A_1, A_2, A_3, а фирма В три своих конкурирующих товара B_1, B_2, B_3. Если фирма А поставит товар A_i, а фирма В - товар B_j, то в результате выигрыш в доходах фирм определяется приведенной платежной матрицей.

Возникает вопрос – как на основе анализа матрицы найти оптимальное решение игры, и что понимать под оптимальным решением игры?

Рассмотрим так называемый принцип минимакса при решении игры.

Ясно, что игроки должны действовать с разумной осторожностью, т. е. необходимо, чтобы мы больше всего реагировали на опасные для нас ходы со стороны противника. По отношению к игроку А рассуждаем так, что если он применит стратегию A_1, то можно зафиксировать самый плохой для него выигрыш: $a_1 = \min a_{ij}$.

Для стратегии A_2: $a_2 = \min a_{2j}$, и так далее.

Для A_m: $a_m = \min a_{mj}$.

Тогда естественно, что игрок А выберет ту стратегию, для которой величина a_i будет максимальной (по строкам в платежной матрице).

Величина $a = \max_i \min_j a_{ij}$ называется нижней ценой игры. Она показывает, меньше какой величины не будет выигрыш игрока А при осторожном подходе к игре. Аналогично рассуждая относительно игрока В, получаем величины:

$b_j = \max a_{ij}$ - максимальные величины в столбцах, и затем

$b = \max_i \min_j a_{ij}$ - верхняя цена игры.

Эта величина показывает, больше какой величины не может быть выигрыш в игре.

Получим значения a и b для вышеприведенных игр.

Для игры «Поиск» $a = -1$; $b = 1$.

Для игры «Три пальца» $a = -3$; $b = 4$.

Для игры «Конкуренты» $a = b = 0$.

Эта игра обладает седловой точкой, так как в ней максимин совпадает с минимаксом. Принятие стратегий, соответствующих седловой точке, взаимоприемлемо для обоих игроков, поэтому их можно принять за оптимальное решение игры. Таким образом, пара стратегий (A_2, B_2) является оптимальным решением игры «Конкуренты». Оптимальный выигрыш в этой игре называется ценой игры и равен 0.

Если игра имеет седловую точку, то говорят, что игра решается в чистых стратегиях. Решение, соответствующее седловой точке, обладает свойством устойчивости: если один игрок примет оптимальную стратегию, то другому невыгодно отклоняться от своей оптимальной стратегии. (Проверьте это, рассмотрев строку A_2 и столбец B_2). Такое свойство устойчивости положено в основу понятия оптимального решения любой игры.

8.3. Решение игр в смешанных стратегиях

Если в игре верхняя и нижняя цена не совпадают (a не равно b), то значит, что седловой точки нет, но можно улучшить средний выигрыш игроков А и В путем смешивания их стратегий. Это значит, что игрок А, каждый раз играя игру, чередует свои чистые стратегии $A_1 A_2 \ldots\ldots A_m$. Вопрос состоит в том, какова должна быть доля случаев применения стратегий игроков.

Назовем смешанной стратегией игрока А вектор $S_A = (p_1, p_2, \ldots, p_m)$, в котором показаны вероятности применения соответствующих стратегий. Например: $S_A = (0{,}1; 0{,}2; 0{,}7; 0)$, $\sum_{i=1}^{m} p_i = 1$ - свойство полноты.

Аналогично для игрока В получаем: $S_B = (q_1, q_2, \ldots, q_m)$, $\sum_{i=1}^{n} q_j = 1$.

Заметим, что чистая стратегия – это частный случай смешанной стратегии. $A_3 = S_A = (0, 0, 1, 0)$.

Справедлива основная теорема теории игр: любая матричная игра имеет решение в виде пары смешанных стратегий игроков А и В $(S_A^*; S_B^*)$, дающих оптимальное значение цены игры V*. Это оптимальное решение обладает следующим свойством: если один игрок придерживается своей оптимальной стратегии, то другому невыгодно отклоняться от своей оптимальной стратегии.

8.4. Упрощение игр и аналитическое решение игр 2×2

Прежде чем решать игру, ее необходимо упростить, т. е. если для игры построена платежная матрица, можно вычеркнуть заведомо невыгодные стратегии для игроков А и В.

Таблица 45

	B_1	B_2	B_3
A_1	5	0	3
A_2	4	-1	2
A_3	2	2	1

→

	B_2	B_3
A_1	0	3
A_3	2	1

Здесь A_2 уступает по всем компонентам A_1, а B_1 уступает и B_2, и B_3. Заметим, что строки лучше, если элементы больше, а столбцы лучше, если элементы меньше.

Если по каждой компоненте стратегии одна строка хуже другой, то ее можно вычеркнуть. При этом решение игры упрощается.

В результате остаются такие строки и столбцы, которые нельзя сравнить. Если число оставшихся стратегий хотя бы одного игрока равно двум, то игру можно решить аналитически или геометрически.

Рассмотрим игры, в которых у каждого игрока лишь две стратегии. Если в такой игре нет седловой точки, то значит, что все стратегии активны, т. е. они войдут в решение с какой-то вероятностью, не равной нулю.

Справедлива теорема об активных стратегиях: если один из игроков придерживается своей оптимальной смешанной стратегии, то выигрыш остается неизменным и равен цене игры V независимо от того, что делает другой игрок, если только тот не выходит за пределы своих активных стратегий.

Пусть мы применяем оптимальную смешанную стратегию S_A^*, а игрок B отклоняется от своей оптимальной стратегии S_B^* и применяет чистую стратегию B_1 или B_2.

Таблица 46

		q_1	q_2
		B_1	B_2
p_1	A_1	a_{11}	a_{12}
p_2	A_2	a_{21}	a_{22}

По свойству решения игры это приведет к ухудшению выигрыша для B. Пусть $(S_A^*; S_B^*)$ дает величину V, тогда если A смешивает A_1 и A_2, а B применяет одну и ту же стратегию B_1, то выигрыш, который получится, будет равен математическому ожиданию.

$$\left.\begin{aligned} &a_{11}p_1 + a_{21}p_2 > v, \text{ если } (S_A^*, B_2), \text{ то} \\ &a_{12}p_1 + a_{22}p_2 > v, \\ &p_1 + p_2 = 1. \end{aligned}\right\} \qquad (8.1)$$

Так как все стратегии активны, то на основании вышеприведенной теоремы можно решить эту систему при равенстве первых двух строк.

$$p_2 = 1 - p_1; \quad a_{11}p_1 + a_{21}(1 - p_1) = v;$$
$$p_1(a_{11} - a_{12} - a_{21} + a_{22}) = -a_{21} + a_{22};$$

$$p_1 = \frac{a_{22} - a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}}; \qquad (8.2)$$

$$p_2 = \frac{a_{11} - a_{12}}{a_{11} + a_{22} - a_{12} - a_{21}}; \qquad (8.3)$$

$$V = \frac{a_{11} \cdot a_{22} - a_{12} \cdot a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}}. \qquad (8.4)$$

Аналогично можно составить систему и решить ее для игрока B:

$$\left.\begin{aligned} &(S_B^*, A_1): \quad a_{11}q_1 + a_{21}q_2 < v; \\ &(S_B^*, A_2): \quad a_{21}q_1 + a_{22}q_2 < v; \\ &q_1 + q_2 = 1; \end{aligned}\right\} \qquad (8.5)$$

$$q_1 = \frac{a_{22} - a_{12}}{a_{11} + a_{22} - a_{12} - a_{21}}; \qquad (8.6)$$

$$q_2 = \frac{a_{11} - a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}}. \qquad (8.7)$$

Например, для платежной матрицы

Таблица 47

	B_1	B_2
A_1	0	3
A_2	2	1

$$p_1 = \frac{1-2}{0+1-2-3} = \frac{1}{4}; \quad q_1 = \frac{1}{2};$$
$$p_2 = 3/4; \quad q_2 = 1/2; \quad v = 1/4;$$

8.5. Геометрическое решение игр

Рассмотрим игру 2x2 . Отложим по оси абсцисс единичный отрезок. Левый конец будет соответствовать чистой стратегии A_1, а правый – стратегии A_2. Любая точка внутри отрезка обозначает смешанную стратегию $S(p_1, p_2)$. На оси ординат будем откладывать выигрыши. На левой вертикали будут выигрыши, полученные при применении чистых стратегий A_1B_1 и A_1B_2, а на правой – A_2B_1 и A_2B_2. Если А применяет смешанные стратегии, а В - чистые, то выигрыши будут ординатами соответствующих прямых. Решим игру для игрока А: для этого найдем сначала самые маленькие выигрыши при различных смешанных стратегиях – это будет линия минимумов. Затем на этой линии найдем наивысшую точку. Получим максимин.

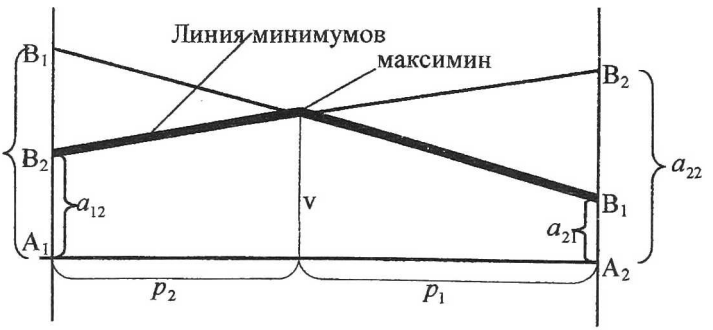


Рис. 61

Аналогично задача решается для игрока B: находим линию максимумов и минимакс.

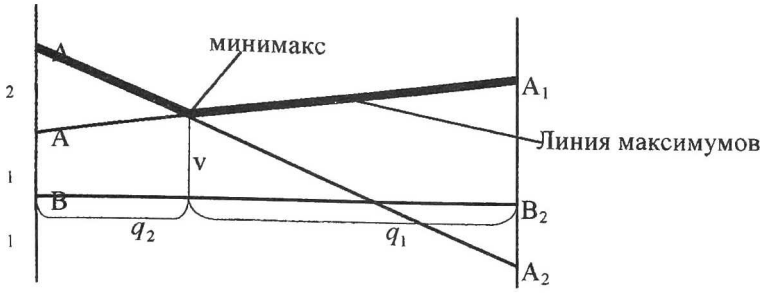


Рис. 62

По принципу максимина игрок A сначала фиксирует свои минимальные выигрыши, а затем выбирает максимальный выигрыш. Так получаются величины p_1 и p_2, q_1 и q_2 на единичном отрезке и величина цены игры v.

Примечания:

- решение не обязательно находится на пересечении прямых;
- если у одного игрока две стратегии, а у другого три и более стратегий, то игра также может быть решена геометрически.

Пример:

Таблица 48

	B_1	B_2	B_3	B_4
A_1	3	0	-1	2
A_2	1	-1	-2	0
A_3	-1	-2	4	-3

Все элементы строки A_2 меньше элементов строки A_1, поэтому A_2 можно отбросить как заведомо невыгодную. Все элементы столбца B_1 больше соответствующих элементов столбца B_2, поэтому стратегию B_1 отбрасываем как заведомо невыгодную.

Таблица 49

	B_2	B_3	B_4
A_1	0	-1	2
A_3	-2	4	-3

После вычеркивания столбца B_1 и строки A_2 игра упрощается. Тогда останутся B_2, B_3, B_4 и A_1 и A_3.

Строим график решения со стороны игрока A.

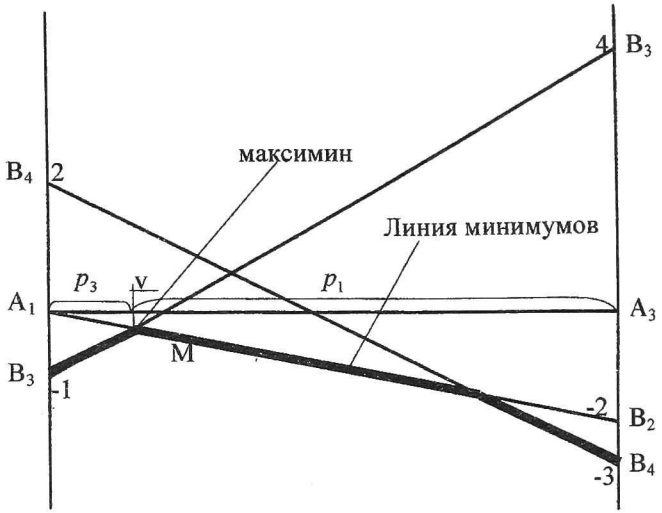


Рис. 63

Находим линию минимумов (на рисунке выделено жирной линией) и определяем на ней максимум, точку М.

Найдем точное решение, для чего определим координаты точки М.

С этой целью зададим уравнения прямых B_2 и B_3 в виде $v = f(p_3)$.

Уравнение прямой построим в виде уравнения с угловым коэффициентом $(y = kx + b)$.

$v = kp_3 + b$, где k – тангенс угла наклона прямой; b – смещение по оси ординат (если прямая возрастает – k>0, если убывает, то k<0).

Построим прямую $\begin{cases} B_3 : v = 5p_3 - 1; \\ B_2 : v = -2p_3 + 0. \end{cases}$

$$p = \frac{1}{7};\ p = \frac{6}{7};\ v = -\frac{2}{7}.$$

Таблица 50

	B_2	B_3
A_1	0	-1
A_4	-2	4

Решим игру со стороны игрока В. Из графика видно, что стратегия B_4 не формирует оптимальное решение – точку М, т. е. не является активной, следовательно, ее отбрасываем (она в согласованном решении не участвует).

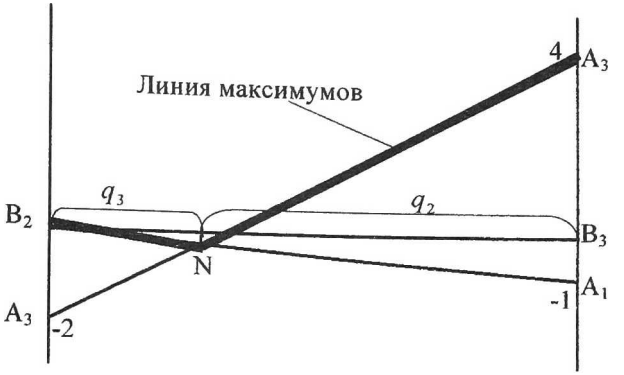


Рис. 64

Находим линию максимумов и определяем на ней минимум – т. N. Запишем уравнения прямых B_1 и A_3.

$$A_3 : v = 6q_3 - 2;$$

$$A_1 : v = -q_3; \quad q_3 = 2/7; \quad q_2 = 5/7.$$

Ответ: Игрок А должен чередовать стратегии A_1 с вероятностью 6/7 и A_3 с вероятностью 1/7. Игрок В должен чередовать стратегии B_2 с вероятностью 5/7 и B_3 с вероятностью 2/7. При этом цена игры v=-2/7.

8.6. Решение игр со многими стратегиями на основе метода линейного программирования

Пусть в игре m стратегий у игрока А и n – у В, матрица не упрощается. Такую игру можно свести к двум задачам линейного программирования. Рассмотрим это на примере матрицы для «игры полковника Блотто» (табл. 51).

Решение не изменится, если матрицу поднять на одну и ту же величину, чтобы все элементы были больше нуля. Получим табл. 52:

Таблица 51

	B_1	B_2	B_3	B_4
A_1	4	2	1	0
A_2	1	3	0	-1
A_3	-2	2	2	-2
A_4	-1	0	3	1
A_5	0	1	2	4

Таблица 52

	B_1	B_2	B_3	B_4
A_1	6	4	3	2
A_2	3	5	2	1
A_3	0	4	4	0
A_4	1	2	5	3
A_5	2	3	4	6

Если мы примем оптимальную смешанную стратегию, и B примет свою оптимальную смешанную стратегию, то получим:

$$\begin{cases} S_A^* = (p_1, p_2, p_3, p_4, p_5); \\ S_B^* = (q_1, q_2, q_3, q_4, q_5). \end{cases}$$

Если A примет оптимальную стратегию, а B – чистую стратегию B_1, т. е. отклонится от оптимальной, то средний выигрыш будет равен $p_1 \cdot 6 + p_2 \cdot 3 + p_3 \cdot 0 + p_4 \cdot 1 + p_5 \cdot 2$ и будет не меньше v.

($M(x) = \sum P_i X_i$ - математическое ожидание случайной величины).

$$\begin{cases} (S_A^*, B_1): 6p_1 + 3p_2 + 0p_3 + 1p_4 + 2p_5 \geq v; \\ (S_A^*, B_2): 4p_1 + 5p_2 + 4p_3 + 2p_4 + 3p_5 \geq v; \\ (S_A^*, B_3): 3p_1 + 2p_2 + 4p_3 + 5p_4 + 4p_5 \geq v; \\ (S_A^*, B_4): 2p_1 + 1p_2 + 0p_3 + 3p_4 + 6p_5 \geq v; \\ p_1 + p_2 + p_3 + p_4 + p_5 = 1. \end{cases} \qquad (8.8)$$

Обозначим $p_i / v = x_i$; $1/v = L$.

Так как при выборе p_i необходимо, чтобы $v \to \max$, то

$$L = x_1 + x_2 + x_3 + x_4 + x_5 = 1/v \to \min.$$

$$\begin{cases} 6x_1 + 3x_2 + 0x_3 + x_4 + 2x_5 \geq 1; \\ 4x_1 + 5x_2 + 4x_3 + 2x_4 + 3x_5 \geq 1; \\ 3x_1 + 2x_2 + 4x_3 + 5x_4 + 4x_5 \geq 1; \\ 2x_1 + x_2 + 0x_3 + 3x_4 + 6x_5 \geq 1. \end{cases} \qquad (8.9)$$

Получим задачу линейного программирования, которую можно решить симплекс-методом (см. главу 2) , в том числе и с использованием стандартных компьютерных программ, например, в системе LINDO или EXCEL.

Аналогичную задачу получаем и со стороны игрока B.

(S_B^*, A_i); $q_j / v = y_j$; $1/v = L$.

При выборе q_j игрок B стремится уменьшить v ($v \to \min$), поэтому

$$L = y_1 + y_2 + y_3 + y_4 = 1/v \to \max. \qquad (8.10)$$

$$\begin{cases} 6y_1 + 4y_2 + 3y_3 + 2y_4 \le 1; \\ 3y_1 + 5y_2 + 2y_3 + y_4 \le 1; \\ 0y_1 + 4y_2 + 4y_3 + 0y_4 \le 1; \\ y_1 + 2y_2 + 5y_3 + 3y_4 \le 1; \\ 2y_1 + 3y_2 + 4y_3 + 6y_4 \le 1. \end{cases} \qquad (8.11)$$

Таким образом, решение игры $m \times n$ сводится к паре задач линейного программирования. Анализируя задачи (8.8), (8.9) и (8.10), (8.11), можно показать, что они всегда имеют решение! Действительно, в качестве допустимого решения (8.8), (8.9) всегда будет: $x_1 = 1/d$; $x_2 = x_3 = \ldots = x_n = 0$; где d – самый маленький коэффициент среди a_{ij} . При этом оптимальное решение также всегда будет, т.к. линейная форма (8.8.) ограничена нулем (все слагаемые положительны). Этим доказывается основная теорема теории игр.

8.7. Биматричные игры

Во многих ситуациях не обязательно, чтобы один игрок выигрывал, а другой столько же проигрывал. Такие игры называются неантагонистическими (с ненулевой суммой). Для двух игроков выигрыши в таких играх приходится записывать не одной, а двумя платежными матрицами, поэтому они имеют название биматричные игры. Интересы игроков здесь не являются полностью противоположными, и их поведение может быть более разнообразным. Различают некооперативные биматричные игры и кооперативные, в зависимости от того, могут ли договариваться игроки, или соглашение невозможно по правилам игры.

Рассмотрим классическую модель некооперативной биматричной игры, известной как «Дилемма Заключенного». Два преступника ожидают приговора суда за совершение преступления. Адвокат конфиденциально предлагает каждому из преступников освободить его, если он сознается и даст показания против сообщника, которому грозит угодить в тюрьму на 10 лет. Если никто не сознается, то обоим угрожает получить 1 год за незначительное преступление. Если сознаются оба преступника, то с учетом чистосердечного признания они получат 5 лет.

Построим платежные матрицы игроков со стратегиями «сознаться А A_1», «сознаться В - B_1», «не сознаваться А – B_2», «не сознаваться В – B_2». Первое число – выигрыш А, а второе - выигрыш В.

Здесь игроки договориться не могут, поэтому им необходимо руководствоваться принципом здравого пессимизма и выбрать максиминную стратегию.

Таблица 52

	B_1	B_2
A_1	(5; 5)	(0; 10)
A_2	(10; 0)	(1; 1)

Если А выберет A_1, то в худшем случае он получит 5 лет, если A_2, то в худшем случае он получит 10 лет, поэтому А «из двух зол» выберет меньшее - A_1.

Игрок В если выберет B_1, то в худшем случае получит 5 лет, а если B_2, то 10 лет. Поэтому, чтобы не рисковать, он выберет B_1. Получили «седловую точку» (A_1,B_1).

Таким образом, обоим заключенным лучше вместе сознаться и получить по 5 лет. Здесь ни одному из игроков не выгодно отклоняться от своей оптимальной стратегии в одиночку. Приведенный подход к решению некооперативных биматричных игр подобен принципу «минимакса» для матричных игр и называется нахождением точки равновесия по Нэшу. Нэш показал, что любая биматричная игра имеет решение в чистых или смешанных стратегиях.

8.8.Кооперативные игры

Кооперативной игрой называют такую модель ситуации, когда выигрыши игроков не имеют нулевой суммы, и в которой игрокам разрешается обсуждать свои стратегии и договариваться о совместных действиях, т. е. образовывать коалиции. Основной задачей здесь является дележ общего выигрыша между членами коалиции.

В случае двух игроков выигрыш в игре – это пара выигрышей, представляющих общие выигрыши. Если игроки будут перебирать всевозможные стратегии (чистые и смешанные), то все пространство выигрышей в игре образует некоторое множество точек, ограниченных некоторой линией (см. рисунок). На рисунке можно указать точку с координатами выигрышей игроков, которые они могут гарантировать себе, не прибегая к договоренностям (T_1, T_2). Такая точка называется точкой угрозы.

На множестве выигрышей можно найти множество Парето-оптимальных решений. Парето-оптимальные точки, безусловно, лучше всех других точек множества выигрышей, так как они являются самыми правыми на горизонтали и самыми левыми на вертикали. Очевидно, что Парето-оптимальные точки – это северно-восточная граница множества выигрышей.

Точки Парето-оптимального множества, находящиеся одновременно выше и правее точки угрозы T, образуют переговорное множество. Очевидно, что игрокам выгодно договариваться, если их выигрыш будет лучше, чем тот, который получается в точке T. На переговорном множестве выделяется

точка решения Нэша (равновесная точка), в которой достигается максимум произведения превышения выигрышей над величинами T_1, T_2:

$$F_{max} = \max(a_1 - T_1)(a_2 - T_2).$$

Рассмотрим пример кооперативной игры «Семейный спор».

Муж и жена каждый вечер решают проблему развлечения: пойти на Футбол или пойти на Балет. Муж любит Футбол и терпеть не может Балета. Жена любит Балет и не любит Футбол, но посещение этих зрелищ в одиночку не дает полного удовлетворения. Поэтому если они вместе, то получают удовольствие на 4 единицы на любимом зрелище и 1 единицу на нелюбимом, а если поодиночке, то 2 единицы на любимом и 0 на нелюбимом. Платежные матрицы имеют такой вид (левое число – выигрыш жены, правое – выигрыш мужа):

Таблица 53

		Жена	
		Балет	Футбол
Муж	Балет	(4; 1)	(0; 0)
	Футбол	(2; 2)	(1; 4)

Множество возможных выигрышей – треугольник с вершинами, соответствующими чистым стратегиям. Ясно, что гарантированный выигрыш супругов – точка угрозы (2; 2). Множество Парето – точки на стороне BC. Переговорное множество – точки на DE. Решение Нэша – точка N. Здесь супруги договариваются половину вечеров проводить вместе на Футболе, а половину - на Балете.

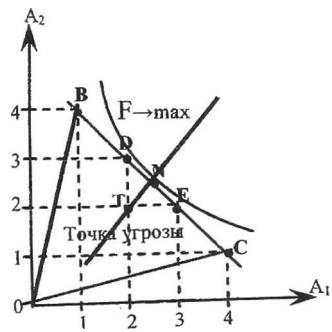


Рис 65

Эта точка получается как точка касания линий постоянного уровня величины $F=(a_1-T_1)(a_2-T_2)$ при $F\to\max$. Следовательно, решение такой игры дает супругам выигрыш (2,5; 2,5).

9. ЭЛЕМЕНТЫ ТЕОРИИ СТАТИСТИЧЕСКИХ ОПТИМАЛЬНЫХ РЕШЕНИЙ

Это чрезвычайно развитая область в экономике, в военном деле, в области обработки информации на фоне шумов и т. д. Рассмотрим элементы этой теории как продолжение теории игр.

Существуют задачи, в которых B - «бессознательный игрок», который мешает нам принимать правильные решения, но он не противодействует активно, а действует в соответствии с природными случайными явлениями, поэтому такую ситуацию называют игрой с природой. Например, помехи в канале связи для передачи информации, шумы при записи или воспроизведении звука и т. д. Ясно что бессознательное воздействие в целом вредит нам меньше, чем сознательное. С другой стороны эта «бессознательность» приводит к непредсказуемому поведению операции. Можно считать ,что как и в теории игр мы боремся с противником, который неосознанно мешает нам. Например, это погодные условия, или случайный спрос при продаже товара. Вот почему в теории статистических решений, главной проблемой является обоснование принципов оценки различных ситуаций со стороны A. Заметим, что в теории игр был один принцип – принцип минимакса.

Таблица 54

	P_1	P_2		P_n
A_1	a_{11}	a_{12}		a_{1n}
A_2	a_{21}	a_{22}		a_{2n}
A_m	a_{m1}	a_{m2}		a_{mn}

Мы будем рассматривать дискретные альтернативы (стратегии) природы. Тогда, если у A имеется m стратегий, а у «природы» имеется n альтернатив, то может быть получена матрица выигрышей, при применении каждой пары $A_i P_j$.

Условия P_j иногда называются гипотезами. Если платежная матрица построена, то задача состоит в анализе матрицы с целью получить стратегию A_i, которая наиболее выгодна по отношению к другим. В простейшем случае, если какие-то строки матрицы заведомо невыгодны, то их можно отбросить и оставить только одну, безусловно лучшую. Столбцы платёжной матрицы

нельзя отбрасывать, т. к. условия «природы» могут быть и в нашу пользу. При анализе платёжной матрицы можно сделать неверный вывод о качестве нашего решения.

Пусть сравниваются два выигрыша, находящихся в разных столбцах a_{ij} и a_{kl}, причём $j \neq l$. Если $a_{ij} > a_{kl}$, то вроде бы решение в i- строке лучше, чем решение в k- строке, но так просто можно сравнивать, если выигрыш соответствует одинаковым условиям.

Пример: Пусть в Томской области, приняв определенные управляющие решения, получили урожайность пшеницы 20 центнеров с гектара, а в Краснодарском крае - 25. Эти значения сравнивать нельзя, т. к. для Томской области это может быть рекордный (наилучший) результат, а для Краснодарского края – плохой, т.к. рекорд 50 . Решение нужно сравнивать с потенциальными возможностями.

Вот почему необходимо преобразовывать платёжную матрицу таким образом, чтобы каждый наш выигрыш соотносился с тем максимумом, который можно достигнуть в данных условиях P_j Для каждого P_j можно найти максимальную величину β_j и вычислить величину $r_{ij} = \beta_j - a_{ij}$, называемую риском. Здесь

$$\beta_j = \max_i a_{ij} \ .$$

Чем риск меньше, тем лучше.

9.1. Принятие решений при известных априорных вероятностях

Будем обозначать вероятности гипотез: $Q_1 = p(P_1)$, $Q_2 = p(P_2)$, …, $Q_n = p(P_n)$. Таким образом, мы считаем, что вероятности известны до того, как мы решили принять решение. Решение - выбор оптимальной стратегии A. Говорят, что это ситуация идеального наблюдателя.

Естественно, в качестве критерия выбирается средний выигрыш, который мы получим, если выберем стратегию A_i :

$$\overline{a_i} = \sum_{j=1}^{n} a_{ij} Q_j \ .$$

Это аналог математического ожидания. Решение принимается по критерию:

$$\max_i \overline{a_i} \ . \qquad (9.1)$$

Это критерий максимального среднего выигрыша. Если задана матрица рисков, то можно для каждой строки вычислить средний риск:

$$\bar{r}_i = \sum_{j=1}^{n} r_{ij} Q_j$$

оптимальным будет являться решение, которое обеспечивает величину

$$\min_i \bar{r}_i \quad . \tag{9.2}$$

Покажем, что оптимальное решение можно искать как по (9.1), так и по (9.2). Сложим средний выигрыш и средний риск,получим константу

$$\bar{a}_i + \bar{r}_i = \sum_{i=1}^{n} a_{ij} Q_j + \sum_{j=1}^{n} r_{ij} Q_j = \sum_{j=1}^{n} a_{ij} Q_j + \sum_{j=1}^{n} \left(\beta_j - a_{ij}\right) Q_j = \sum_{j=1}^{n} \beta_j Q_j = c\,,$$

огда

$$\bar{a}_i + \bar{r}_i = c \Rightarrow \bar{a}_i = c - \bar{r}_i; \quad \max \bar{a}_i = \min \bar{r}_i; \quad \arg\max_i \bar{a}_i = \arg\min_i \bar{r}_i\,.$$

C – величина, не зависящая от *i*, это постоянная величина для данной троки.

В результате решения выбирается чистая стратегия a_i. Есть ли смысл смешивать стратегии? Пусть мы смешиваем наши стратегии с вероятностями p_i, тогда в результате применения смешанной стратегии мы получим средний выигрыш в таком виде:

$\bar{\bar{a}} = \bar{a}_1 p_1 + \bar{a}_2 p_2 + \ldots + \bar{a}_m p_m$ - математическое ожидание (МО).

Мы знаем, что МО меньше максимального значения, следовательно в игре с природой нет смысла смешивать стратегии; чистая стратегия обеспечивает наилучший результат. Слабым местом в этом подходе является то, что надо знать априорные вероятности. Если они неизвестны, то необходимо их изучить. Это можно делать путём экспериментов, которые изучают условия «природы». Говорят, что этим мы обучаем нашу систему. Такой подход называется принципом адаптации к условиям.

Если априорные вероятности изучить не удаётся, то применяется принцип недостаточности основания: если не знаем о вероятности, то считаем, что гипотезы природы равновероятны. После этого применяем критерий идеального наблюдателя. Так как при равных вероятностях энтропия (неопределённость) максимальна, то мы рассчитываем на худший случай , т.е.применяем принцип пессимизма. Если сами значения вероятности неизвестны, но есть информация о предпочтениях гипотез, то существуют методы обработки предпочтений и получения вероятностей.

Если вероятности гипотез относятся как

$$n:(n-1):(n-2):\ldots:2:1=Q_1:Q_2:\ldots:Q_n;\quad \sum_{j=1}^{n}Q_j=1,$$

то можно подсчитать саму величину вероятности в следующем виде:

$$Q_j=\frac{2(n-j+1)}{n(n+1)};\quad j=\overline{1,n}.$$

В некоторых случаях учитывается не только средний выигрыш, но также и дисперсия, т. е. величина разброса выигрыша в каждой строке:

$$\max_i\left(\overline{a_i}-k\times\sigma_{a_i}\right);\quad \sigma_{a_i}=\sqrt{D_{a_i}}.$$

9.2. Методы принятия решений в условиях априорной неопределенности

Если априорная информация неизвестна или ненадёжна, то применяются другие критерии.

Критерий Вальда (*W*): это максиминный критерий, т. е. выбирается стратегия A_k, для которой

$$W=\max_i\min_j a_{ij}.$$

При таком критерии мы подходим к этой задаче, рассчитывая на самый худший случай, как и в игре с разумным противником.

Критерий Сэвиджа (*S*): это критерий минимаксного риска

$$S=\min_i\max_j r_{ij}.$$

Этот критерий не эквивалентен критерию Вальда, т. е. стратегия, оптимальная по Сэвиджу, не обязательно также будет оптимальна по Вальду.

Критерий Гурвица (*H*): это комбинированный критерий, его также называют критерием пессимизма-оптимизма

$$H=\max_i\left(\kappa\min_j a_{ij}+(1-\kappa)\max_j a_{ij}\right),$$

где $0<\kappa<1$ - коэффициент, который выполняет требования критерия быть более или менее оптимистичным. При k=1 критерий H=W, а при k=0 получаем крайний оптимизм.

Существуют другие критерии и однозначный выбор одного критерия невозможен. Но если одну и ту же ситуацию рассматривать по разным критериям и

лучаются одинаковые решения, то это говорит об устойчивости данной си-ации и однозначности найденного оптимального решения. В противном слу-е ситуация неустойчивая и необходимо либо изучать априорную информа-ию, либо доказывать верность лишь одного из критериев.

9.3. Планирование эксперимента при принятии решений

Если априорной информации нет или она ненадёжна, то можно путём проведения эксперимента получить более надёжные данные о вероятностях Q_j. Под экспериментом понимают систему мероприятий позволяющих уточнить информацию о состоянии «природы». Например, это метеонаблюдения, маркетинговые исследования в экономике, радиолокация, военная или промышленная разведка и т.д. Насколько может помочь в принятии решения эксперимент и как сопоставить стоимость эксперимента с тем оптимальным выигрышем, который мы получим?

Соответствующую теорию можно построить исходя из знания вероятностей Q_j, а также на основе критериев при неизвестной априорной информации. Мы рассмотрим случай, когда есть априорная информация, т. е. ситуацию идеального наблюдателя. Рассмотрим вопрос: есть ли смысл проводить эксперимент? Рассмотрим два случая.

1) Идеальный эксперимент. Результат этого эксперимента однозначно определяет каковы условия природы. Пусть заданы выигрыши a_{ij} и априорные вероятности Q_j. Стоимость эксперимента c сопоставима с выигрышем т. е. эти величины имеют одинаковую размерность. Сравним средний выигрыш без проведения эксперимента со средним выигрышем при проведении эксперимента:

Если нет эксперимента, мы обеспечим себе

$$\max_i \left(\sum_{i=1}^{n} a_{ij} Q_j \right) = \overline{a_{\max}} .$$

Если мы проведём эксперимент, то мы точно узнаем $P_j = P_k$, и тогда найдя в k-м столбце максимальный выигрыш, мы найдём наш выигрыш:

$$\max_i a_{ij} = \beta_k .$$

Но нам нужно оценить эффективность эксперимента до его проведения, поэтому мы должны ориентироваться на средний ожидаемый выигрыш, который мы получим, если будем проводить эксперимент. Таким образом, перед проведением эксперимента мы можем ожидать выигрыш

$$\sum_{j=1}^{n} Qj \times \beta_j .$$

Поэтому чтобы решить, проводить эксперимент или нет, надо определить, что больше: $a_{\max}$ или

$$\sum_{j=1}^{n} Qj \times \beta_j .$$

Получается, что необходимо проводить эксперимент, если

$$\sum_{j=1}^{n} Q_j \beta_j - c > \max_i \sum_{j=1}^{n} Q_j a_{ij} .$$

Преобразовав это неравенство, получим, что идеальный эксперимент проводить нужно, если его стоимость меньше минимального среднего риска

$$c < \min_i \bar{r}_i .$$

2) Неидеальный эксперимент. В результате проведения неидеального эксперимента мы не находим однозначно P_j, а лишь изменяем вероятности Q_j. Пусть проводится неидеальный эксперимент. В результате появляются некоторые несовместные события $B_1, B_2, \ldots B_L$. Вероятности этих событий зависят от условий, в которых они проводятся. Пусть известны $P(B_l/P_j)$. Эти вероятности называются прямыми. После эксперимента, давшего исход B_l необходимо пересмотреть вероятности Q_j, т. е. вместо вероятности Q_j мы перейдём к вероятности Q_{jl}. Это так называемые апостериорные вероятности

$$\overline{Q_{jl}} = P\left(P_j/B_l\right) = \frac{Q_j \times P\left(B_l/P_j\right)}{\sum_{j=1}^{n}\left(Q_j \times P\left(B_l/P_j\right)\right)}.$$

Это формула Байеса.

Но результаты эксперимента могут быть и B_1, и B_2, и B_k, поэтому мы можем только ожидать всякие исходы B_l, которые получатся в результате эксперимента. Причём, каждый исход B_l привёл бы к некоторым оптимальным стратегиям A^*_1. А величина выигрыша, которая бы при этом получилась:

$$\bar{\tilde{a}}_l = \max_i \bar{\tilde{a}}_{il} = \max_i \sum_{j=1}^{n}\left(\tilde{Q}_{jl} \times a_{ij}\right).$$

Эти выигрыши $\overline{\tilde{a}}_l$ могут произойти с вероятностью события B_l, т. е. с вероятностью $P(B_l)$, которую можно получить по формуле полной вероятности:

$$P(B_l)=\sum_{j=1}^{n} Q_j P(B_l/P_j).$$

Тогда средний ожидаемый выигрыш будет:

$$\overline{\overline{\tilde{a}}}=\sum_{l=1}^{k} a_l P(B_l) \Rightarrow \overline{\overline{\tilde{a}}}-c>\max_i \sum Q_j a_{ij} \Rightarrow \overline{\overline{\tilde{a}}}-c>\overline{a}_{\max}.$$

Планирование экспериментов можно рассмотреть и для случаев, когда проводят 2, 3, 4, ... эксперимента [8].

9.4. Многоэтапное принятие решений

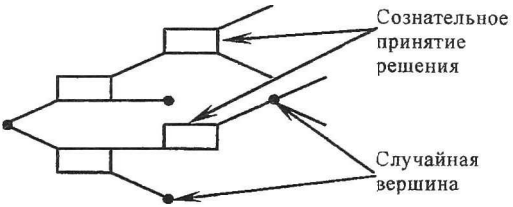


Рис. 66

Мы рассмотрели различные критерии принятия решений в условиях неопределённости. На практике, в таких задачах как проектирование изделий, управление процессами, создание сложных программных комплексов, мы можем столкнуться с принятием последовательных решений (цепочкой решений).

Особое значение такие многоэтапные решения имеют при создании автоматизированных экспертных систем. Рассмотрим вопрос оптимизации многоэтапных решений. Многоэтапность приводит к тому, что схема принятия решения может быть представлена в виде дерева, в каждой вершине которого осуществляется:

1) Сознательный выбор между двумя и более альтернативами.

2) Либо случайный переход из одной ветви в другую под воздействием внешних случайных факторов.

Рассмотрим пример оптимизации многоэтапных решений на примере экономической задачи.

Пример. Фирма может принять решение о строительстве крупного или мелкого предприятия. Строительство крупного предприятия относительно дешевле, в случае если будет высокий спрос на производимые товары, мелкое предприятие можно расширить. Деятельность фирмы рассматривается в течение десяти лет, причём в случае строительства мелкого предприятия вопрос о расширении будет рассматриваться через два года. Спрос заранее неизвестен.

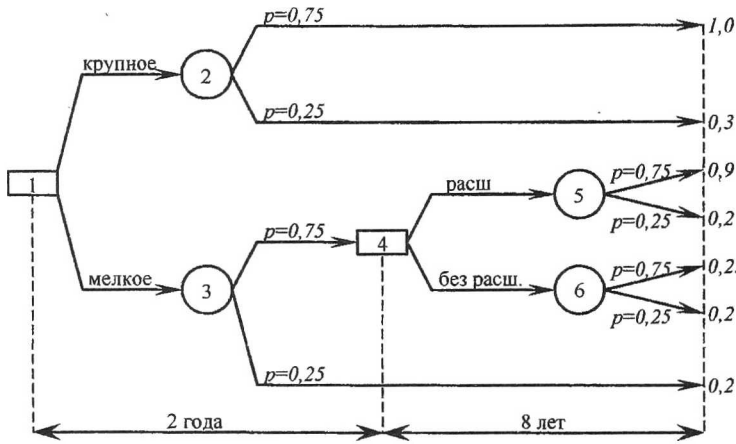


Рис. 67

Введём градацию случайного спроса: высокий (p>0,75) и низкий (p<0,25). Затраты и доходы: строительство крупного предприятия - 5 млн. $; строительство мелкого - 1 млн. $; затраты на расширение - 4,2 млн. $; крупное предприятие при высоком спросе даёт доход - 1 млн. $ ежегодно, а при низком - 300 тыс. $; мелкое предприятие при высоком спросе - 250 тыс. $ ежегодно, при низком - 200 тыс. $; расширенное предприятие в случае высокого спроса приносит доход - 900 тыс. $ в год, и при низком спросе - 200 тыс. $; мелкое предприятие без расширения при высоком спросе на производимый продукт приносит в течение двух лет по 250 тыс. $ ежегодно, а в течение следующих восьми по 200 тыс. $. Нарисуем дерево решений.

Применим для решения этой задачи метод динамического программирования. В качестве критерия применим средний выигрыш, т. е. МО выигрыша. Сама величина критерия равна доходу без затрат на строительство. Начнём с последнего четвёртого шага, подсчитаем средний выигрыш:

$$\overline{a}^4_{расш} = (0,9*0,75+0,2*0,25)*8-4,2=1,6,$$

$$\overline{a}^4_{без\ расш} = (0,25*0,75+0,2*0,25)*8=1,9,$$

$$\overline{a}^1_{круп} = (1*0,75+0,3*0,25)*10-5,0=3,25,$$

$$\overline{a}^1_{мелк} = (1,9+2*0,25)*0,75+0,2*10*0,25=1,3.$$

Исходя из полученного результата, оптимальным будет сразу строить крупное предприятие.

Другим примером оптимизации многоэтапных операций является известная «задача о секретарше».

Директор собирается принять на работу секретаршу. Прежний опыт делит секретарш на три категории: отличных (3 балла), хороших (2 балла) и посредственных (1 балл). Анализ учебных заведений по подготовке секретарш даёт статистику выпускниц заведений: вероятность взять на работу отличную секретаршу - 0,2, хорошую - 0,5, посредственную - 0,3. директор может испытать только трёх претенденток, причём в случае отказа директора кандидат убывает на другую работу. Построим дерево решений.

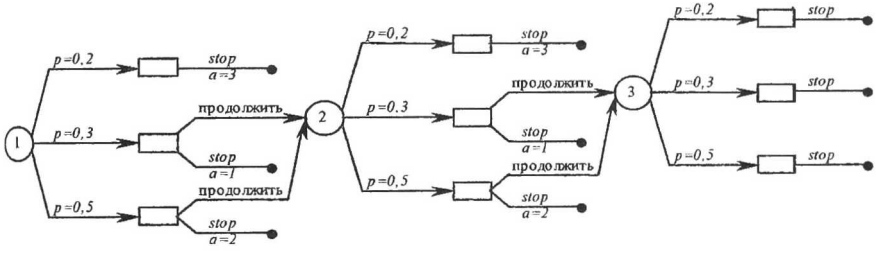


Рис. 68

В соответствии с процедурой динамического программирования начнём искать оптимальное решение с последнего шага. Определим математическое ожидание «выигрыша», если мы испытываем третьего кандидата:

$$\overline{a}_3 = 3*0,2+2*0,5+1*0,3=1,9.$$

Далее определим средний выигрыш, если мы испытываем второго и третьего, с учетом того, что если второй будет посредственный, то мы продолжим и получим в среднем 1.9.

$$\overline{a}_2 = 3*0,2+2*0,5+1,9*0,3=2,17.$$

Поэтому, если во втором испытании попалась хорошая секретарша, надо остановиться, т.к. получим 2, а если продолжим, то только 1,9.

При первом испытании, надо остановиться, только если попалась отличная, а в третьем испытании берём любую. Найдём средний оптимальный выигрыш при оптимальном правиле испытания трех кандидатов:

$$\overline{a}_1 = 3*0,2+2,17*0,5+2,17*0,3=2,336.$$

Следовательно, за счет возможности испытывать трех секретарш мы получаем дополнительный выигрыш 2,336 – 1,9 = 0,436.

10. ЭКСПЕРТНЫЕ ПРОЦЕДУРЫ ДЛЯ ПРИНЯТИЯ РЕШЕНИЙ

В практических задачах принятия оптимального решения альтернативы не являются «математическими объектами», а чаще представляют собой конкретные физические системы: продукты, технологии, организация технического мероприятия, системы и т.д. Для описания альтернатив и оценки последствий их принятия необходимо решить следующие задачи:

– построить множество возможных и допустимых альтернатив;

– сформировать набор аспектов, существенных для оценки альтернатив;

– определить критериальное пространство;

– упорядочить альтернативы по аспектам;

– получить оценку альтернатив по критериям, то есть найти отображение Ω в критериальное пространство (см. главу 7).

Все эти задачи являются модификацией общей задачи оценивания: сопоставление числа или нескольких чисел рассматриваемой альтернативе.

Методы решения задачи оценивания основаны на использовании экспертных процедур, поэтому их называют методами экспертных оценок.

10.1. Общая схема экспертизы

В общем случае из-за сложности оценивания систем привлекаются люди – специалисты в данной предметной области – эксперты. Решение задач оценивания называют экспертизой. Вопросы, связанные с экспертизой, рассматриваются и решаются консультантом. Он определяет Ω, а иногда и вспомогательное множество для экспертизы $\Omega_э$ и организует всю процедуру экспертизы.

1. Консультант находит множество допустимых оценок Ω, в которых содержится исходная оценка.
2. Он определяет множество допустимых оценок $\Omega_э$, из которого осуществляют выбор эксперты.
3. Каждый эксперт выбирает свою оценку $a_i = C_i$ $(\Omega_э)$, $i=\overline{1,N}$. При этом эксперты могут взаимодействовать между собой.
4. По заранее разработанному алгоритму (формуле) консультант производит обработку полученной от экспертов информации и находит результирующую оценку, являющуюся решением исходной задачи оценивания.
5. Если полученное решение не устраивает консультанта, то он предоставляет экспертам дополнительную информацию, то есть организует обратную связь, после чего они вновь решают задачу оценивания.

10.2. Задача оценивания

Смысл оценивания состоит в сопоставлении альтернативе вектора из E_m. Перечислим типичные варианты этой задачи:

1. Пусть $X \in \Omega$ – альтернатива в задаче принятия решений. Имеются m критериев. Требуется альтернативе $X \in \Omega$ сопоставить вектор

   $[f_1(x), f(x), \ldots f_m(x)] \in E_m$.

2. Пусть k_1, k_2, ..., k_m – критерии, учитывающиеся при выборе. Их необходимо упорядочить по важности. Тогда системе $S=(k_1, k_2, \ldots, k_m)$ сопоставляется перестановка натуральных чисел от 1 до m, $i_1, \ldots i_m$, где i_k – номер k-го критерия при упорядочивании по важности.
3. Пусть множество D разбито на l подмножеств D_1, D_2, ..., D_l. Для элемента $x \in D$ необходимо указать, к какому из подмножеств D_i $(i = \overline{1,l})$ он относится. То есть x сопоставляется одно из чисел от 1 до l. Пусть x отрезок, длину которого надо измерить. То есть отрезку надо сопоставить действительное число; $f(x)$ – длина отрезка.

Задача №1 – это общая задача многокритериальной оценки.
Задача №2 – это задача ранжирования.
Задача №3 – это задача классификации.
Задача №4 – это обычная задача измерения.
Обозначим Ω – исходное множество допустимых значений оценок (МДО).
$\Omega_э$ - МДО для экспертов;
L – взаимодействие между экспертами;
Q – обратная связь;
φ – обработка (отображение $\Omega_э^N \rightarrow \Omega$).

Назовем схемой экспертизы пятерку параметров $(\Omega, \Omega_э, L, Q, \varphi)$.

Подготовка экспертизы – это предварительная разработка схем экспертизы и подбор экспертов.

Реализация экспертизы – получение от экспертов информации и её обработка.

10.3. Подготовка экспертизы

Определение МДО (множества допустимых оценок).

Перечислим типы МДО и укажем определяющие их задачи оценивания.

1. $\Omega = \{0,1\}$. Соответствующая задача попарного сравнения заключается в выявлении лучшего из двух имеющихся объектов a и b:

$$C(\Omega)=\begin{cases}1, \textit{если } a \textit{ лучше } b,\\ 0, \textit{в противном случае.}\end{cases}$$

2. $\Omega=\{(1,2,\ldots,n)(1.3,\ldots n)\ldots(n,\ n\text{-}1,\ldots 1)\}$. То есть $\mathbf{\Omega}$ состоит из множества перестановок длины n. Соответствующая задача ранжирования состоит в упорядочивании объектов:

   $C(\Omega)=(i_1,i_2,\ldots,i_n)$.

3. $\Omega=\{1\ldots l\}$. Соответствующая задача классификации

   $C(\Omega)=i$,если $X \in S_i$.

Определение МДО для экспертов $\Omega_э$.

Типы $\Omega_э$ те же, что и Ω. Но $\Omega_э$ зависит от формы опроса: интервью, анкетирование, метод докладной записки эксперта – высказывание мнения в свободной форме.

Взаимодействие экспертов L.

1. Свободный обмен информацией между экспертами.
2. Обмен регламентирован.
3. Эксперты изолированы.

Пункт (2) может иметь вид «Мозговой атаки»: в течение заданного времени любое мнение не может быть отвергнуто и не подлежит обсуждению.

Обратная связь. После обработки можно ознакомить экспертов с результатами и попросить дать повторную оценку.

Известен метод Делфи. После экспертизы проверяется согласованность, а затем, если она недостаточная, экспертам сообщается дополнительная информация и аргументация других экспертов. Затем экспертиза повторяется.

Подбор экспертов. Существуют методы оценки компетентности экспертов. Эксперты могут выставлять свои оценки компетентности себе и другим. Затем эта информация обобщается и используется в алгоритмах оценки.

10.4. Методы обработки экспертной информации

Существуют три вида методов обработки:

– статистические методы;
– алгебраические методы;
– методы шкалирования.

Рассмотрим статистические методы экспертных оценок.

Результаты оценок каждого эксперта можно рассматривать как реализацию некоторой случайной величины из Ωэ и применять к ним методы математической статистики. Статистические методы позволяют определить согласованность мнений экспертов, значимость полученных оценок и т.д., то есть качество экспертизы.

Численные оценки. Экспертиза $Э_1$.

Задача состоит в сопоставлении оцениваемой альтернативе (системе) одного числа.

(Ω=E_1, $\Omega_э$=E_1, L – эксперты изолированы; Q – обратная связь отсутствует)

$$\varphi(x_1, x_2, \ldots x_N) = \frac{\sum_{i=1}^{N} x_i \alpha_i}{\sum_{i=1}^{N} \alpha_i} = a, \tag{10.1}$$

где α_i – вес (коэффициент компетентности) экспертов;

x_i – числовые оценки экспертов.

При отсутствии информации о компетентности экспертов α_i=1. Степенью согласованности мнений экспертов служит дисперсия:

$$\sigma^2 = \frac{\sum_{i=1}^{N} (a - x_i)^2 \alpha_i}{\sum_{i=1}^{N} \alpha_i}. \tag{10.2}$$

Другая *экспертиза* $Э_2$ повышает точность оценивания ($\Omega_э$=E_3).

$$\varphi\left(x_1^1, x_2^1, x_3^1, x_1^2, x_2^2, x_3^2, \ldots, x_1^N, x_2^N, x_3^N\right) = \sum_{i=1}^{N} \frac{x_1^i \gamma_1 + x_2^i \gamma_2 + x_3^i \gamma_3}{\gamma_1 + \gamma_2 + \gamma_3} \alpha_i \cdot \frac{1}{\sum_{i=1}^{N} \alpha_i}. \tag{10.3}$$

где x_1^i, x_2^i, x_3^i – оптимистическая, наиболее вероятная, и пессимистическая оценки i –го эксперта.

$\gamma_1, \gamma_2, \gamma_3$ – определяются эмпирически (например, по одной методике γ_1=1, γ_2=4, γ_3=36, по другой γ_1=3, γ_2=0, γ_3=2, γ_4=25, где γ_4 – степень неуверенности эксперта в своем ответе, $\gamma_1 > \gamma_3$, так как человек склонен к занижению оценки).

Степень согласованности экспертов определяется дисперсией

$$\sigma^2 = \sum \alpha_i \sigma_i^2 \cdot \frac{1}{\sum_{i=1}^{N} \alpha_i} + \sum_{i=1}^{N} \left(a - \frac{x_1^i \gamma_1 - x_2^i \gamma_2 - x_3^i \gamma_3}{\gamma_1 + \gamma_2 + \gamma_3} \right)^2 \cdot \alpha_i \frac{1}{\sum_{i=1}^{N} \alpha_i}, \tag{10.4}$$

$$\sigma_i^2 = \left(x_3^i - x_1^i\right)^2 \cdot \frac{1}{\gamma_4}.$$

В *экспертизах* $Э_1$ и $Э_2$ можно определить статистическую значимость полученных результатов. Задавшись $P_{ош}$, укажем интервал, в который оцениваемая величина попадет с вероятностью $1\text{-}P_{ош}$.

$$\bar{a} - \Delta < a < \bar{a} + \Delta, \qquad (10.5)$$

a - распределена нормально с центром $\bar{a}$ и дисперсией (10.2).

Тогда

$$\Delta = t\sigma\sqrt{N},$$

где t – стандартная ошибка, находящаяся по таблице распределения коэффициента Стьюдента, $t = f(N\text{-}1, P_{ош})$.

Пример. Десять экспертов с одинаковыми весами α_i=1 оценивают величину Т. От них получены результаты:

T_1=33 T_2=35 T_3=32,2 T_4=34 T_5=38

T_6=34 T_7=37 T_8=40 T_9=36 T_{10}=35,5

По формуле (10.1) получаем Т=35,5, σ^2=4,9, σ=2,2136.

Задав $P_{ош}$=0,05, определим $t=f(9;0,005)$=2,262. По формуле (10.5) Δ=1,583. Таким образом, с вероятностью 0,95 величина Т находится в интервале [33,917; 37,083].

10.5. Метод Делфи для численной оценки

$\Omega = E_1$, $\Omega_э\{ Z \in E_к \mid \sum Z_i = 1; Z_i \geq 0\}$;

L – эксперты изолированы;

Q – экспертам предоставляется медиана q_2;

$P(T \leq q_2) = 0,5$;

а также диапазон квантилей

$\Delta q = q_3 - q_1$;

$P(T \leq q_3) = 0,75$;

$P(T \leq q_1) = 0,25$;

φ задается следующим образом.

Весь интервал допустимых значений оцениваемой величины разбивается на k интервалов $t_1, t_2, \ldots, t_k$; эксперт оценивает вероятность попадания оцениваемой величины в каждый из интервалов; по результатам их оценок составляется таблица,

Таблица 55

Эксперты	Интервалы				
	t_1	t_2	t_3	...	t_k
1	P_{11}	P_{12}	P_{13}	...	P_{1k}
2	P_{21}	P_{22}	P_{23}	...	P_{2k}
...	...	...	...	...	...
n	P_{n1}	P_{n2}	P_{n3}	...	P_{nk}

где P_{ij} – оценка вероятности попадания оцениваемой величины в j-й интервал, данная i-м экспертом.

На основе этой таблицы определяется мнение экспертов о попадании оцениваемой величины в каждый из интервалов

$$P_{t_j} = \sum_{i=1}^{N} P_{ij}\alpha_i \cdot \frac{1}{\sum_{i=1}^{N}\alpha_i}. \qquad (10.6)$$

Результирующей оценкой является медиана распределения

$P(T \le q_2) = 0{,}5.$

Ее показывают экспертам, а также показывают диапазон

$\Delta q = q_3 - q_1.$

Затем снова эксперты дают оценки P_{ij}, и т.д. до тех пор, пока диапазон Δq не уменьшится в 1,6 раза по сравнению с предварительным.

10.6. Строгое ранжирование

Ω – множество всех перестановок, $\Omega_э = \Omega$;
L – эксперты изолированы друг от друга;
Q – обратная связь отсутствует.

Отображение φ определяется так: результаты опроса сводятся в таблицу рангов данных i-м экспертом j-му объекту.

$$r_s = \sum_{i=1}^{N} r_{iS}.$$

На первое место ставится объект с меньшей величиной r, $r_{lj} \neq r_{lk}$ (в строках нет одинаковых рангов).

Таблица 56

Эксперты	Объекты			
	1	2	...	N
1	r_{11}	r_{12}	...	r_{1N}
2	r_{21}	r_{22}	...	r_{2N}
...				
N	r_{N1}	r_{N2}	...	r_{NN}
$\sum$ рангов	r_1	r_2	...	r_n

Степень согласованности мнений экспертов определяется при помощи коэффициента конкордации:

$$W = \frac{12\sum_{j=1}^{N}\left[r_j - \frac{1}{2}N(n+1)\right]^2}{N^2(n^3-n)}, \qquad (10.7)$$

где N – число экспертов;

n – число объектов;

r_j – ранги объектов, полученных в результате экспертизы.

По сути дела, W – это сумма квадратов отклонений рангов r_j от среднего значения

$$r_{jcp} = \sum_{j=1}^{N}\frac{r_j}{n} = \frac{N(n+1)}{2}. \qquad (10.8)$$

Пример. Проводится экспертиза по оценке технологического процесса выплавки стали в конверторе. Задан список из шести признаков, влияющих на процесс. Десять экспертов ранжировали признаки по важности.

Таблица 57

Признаки	Номера экспертов										
	1	2	3	4	5	6	7	8	9	10	$\sum r_{ij}$
Шум	6	1	6	6	6	6	4	5	6	6	52
Цвет футеровки	4	5	4	5	5	3	5	6	4	5	46
Цвет пламени	2	2	2	3	3	2	1	1	1	2	19
Цвет дыма	1	4	3	2	2	4	3	3	3	3	28
Качество дыма	3	3	1	1	1	1	2	2	2	1	17
Искры	5	6	5	4	4	5	6	4	5	4	48

r_{icp}=35;
W=0,69.

При четном $N(n+1)$ может быть W=0.

10.7. Нестрогое ранжирование

Если объекты делят места (4) и (5), то каждый получает 4,5.
В этом случае

$$W = \frac{12\sum_{j=1}^{N}\left[r_j - \frac{1}{2}N(n+1)\right]^2}{N^2(n^3-n) - N\sum_{i=1}^{N}\sum_{j=1}^{k_i}(t_{ij}^3 - t_{ij})}, \qquad (10.9)$$

где k_i - число групп равных рангов, введенных i-м экспертом;

t_{ij} - количество дробных рангов в j-й группе, введенной i-м экспертом.

Можно учесть компетентность α_i:

$$r_j = \sum_{i=1}^{N} r_{ij}\alpha_i .$$

10.8. Метод попарных сравнений

В этом методе эксперт i фиксирует предпочтение S-го объекта над j-м объектом в виде чисел {1,0}.

$$d_{s_j}^i = \begin{cases} 1, \text{если } S \text{ превосходит объект } j; \\ 0, \text{в противном случае.} \end{cases}$$

Далее эти результаты записываются в матрицу

$$D = \left\{ d_{s_j} \right\}, \text{ где } d_{s_j}^i = \begin{cases} 1, \text{при } d_{s_j}^i(+) > d_{s_j}(-); \\ 0, \text{в противном случае;} \end{cases}$$

$$\mathrm{d}_{s_j}^i(-) = \sum_{i \in M^-} \mathrm{d}_{s_j}^i ;$$

$$\mathrm{d}_{s_j}^i(+) = \sum_{i \in M^+} \mathrm{d}_{s_j}^i ;$$

M^+-множество экспертов, считающих что S доминирует j;

M^- - множество экспертов, считающих наоборот;

$M^- = M - M^+$.

Окончательный ранг объекта определяется по сумме элементов строки

$$r_s = \sum_{j=1}^{n} d_{Sj} . \qquad (10.10)$$

Пример:

$$D = \left\{ d_{s_j} \right\} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

r_S=(2, 3,0,1).

Следовательно, объекты O_S необходимо упорядочить по ценности так: (O_2, O_1, O_4, O_3).

СПИСОК ЛИТЕРАТУРЫ

1. Таха Хэмди А. Введение в исследование операций.-6-е изд.- М.: Изд-во Дом «Вильямс», 2001. – 912 с.
2. Исследование операций в экономике: Учеб. пособие для втузов/ Н.Ш.Кремер, Б.А. Путко, И.М.Тришин, М.Н.Фридман; под ред. проф. Н.Ш.Кремера. – М.: ЮНИТИ, 2006. – 407с.
3. Волков И.К., Загоруйко Е.А. Исследование операций: Учеб. пособие для вузов. 2-е изд./ Под ред. В.С.Зарубина, А.П. Крищенко. – М.: Изд-во МГТУ им. Н.Э.Баумана, 2002. – 436 с.
4. Пантелеев А.В. Методы оптимизации в примерах и задачах: Учеб. пособие / А.В.Пантелеев, Т.А. Летова – М.: Высш. шк., 2002.—544 с.
5. Акулич И.Л. Математическое программирование в примерах и задачах: Учеб. пособие. – М.: Высш. шк., 1986. – 319 с.
6. Кузнецов Ю.Н., Кузнецов В.И., Волощенко А.Б. Математическое программирование: Учеб. пособие. – 2-е изд. – М.: Высш. шк., 1980. – 300 с.
7. Линейное и нелинейное программирование / Ляшенко И.Н., Карагодова Е.А., Черникова Н.В., Шор Н.З.: Учеб. пособие. – Киев: Вища школа, 1975. – 370 с.
8. Вентцель Е.С. Исследование операций. – М.: Советское радио, 1972. – 550 с.
9. Черноруцкий И.Г. Методы оптимизации и принятия решений: Учеб. пособие. – СПб.: Изд-во «Лань», 2001. – 384 с.
10. Зайченко Ю.П. Исследование операций. – Киев: Вища школа, 1986. – 390 с.
11. Дегтярев Ю.И. Исследование операций: Учеб. пособие для втузов по спец. АСУ. – М.: Высш. шк., 1986. – 320 с.
12. Базара М., Шетти К. Нелинейное программирование. Теория и алгоритмы. – М.: Мир, 1982. – 410 с.
13. Морозов В.В., Сухарев А.Г., Фёдоров В.В. Исследование операций в задачах и упражнениях: Учеб. пособие. - М.: Высш. шк., 1986. – 287 с.
14. Кудрявцев Е.М. Исследование операций в задачах, алгоритмах и программах. – М.: Радио и связь, 1984. - 184 с.
15. Решение задач по курсу «Исследование операций». Дискретное программирование: Метод. пособие / Сост. Б.А.Есипов. – Куйбышев: КуАИ, 1984. – 26 с.
16. Решение задач по курсу «Исследование операций». Нелинейное программирование: Метод. пособие / Сост. Б.А.Есипов. – Куйбышев: КуАИ, 1984. – 26 с.
17. Практические занятия по исследованию операций: Метод. пособие / Сост. Б.А.Есипов. – Куйбышев: КуАИ, 1982. – 32 с.
18. Пакет прикладных программ для решения задач по исследованию операций на ЭВМ. Целочисленное программирование: Метод. пособие/ Сост. Б.А.Есипов, А.В. Баландин, В.Г.Гашников. – Куйбышев: КуАИ, 1982. – 32 с.

19. Пакет прикладных программ для решения задач по исследованию операций на ЭВМ. Линейное частично целочисленное и нелинейное программирование: Метод. пособие / Сост. Б.А.Есипов, А.В. Баландин, В.Г.Гашников. – Куйбышев: КуАИ, 1982. – 32 с.
20. Математические методы и модели в экономике: Метод. пособие / Сост. Б.А.Есипов. – Самара: МИР, 2003. – 40 с.
21. Аоки М. Введение в методы оптимизации. – М.: Наука, 1977. – 344 с.
22. Зангвилл У.И. Нелинейное программирование. – М.: Сов. радио, 1973.
23. Саати Т. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы. – М.: – Мир, 1973. – 302 с.
24. Фиакко А., Мак-Кормик Г. Нелинейное программирование. Методы последовательной безусловной минимизации. – М.: Мир, 1972. – 238 с.
25. Сборник задач по математике для втузов. Ч. 4. Методы оптимизации: Учеб. пособие / Под ред. А.В.Ефимова. – 2-е изд. – М.: Наука, 1990. – 304 с.
26. Орлов А.И. Теория принятия решений: Учебник / А.И.Орлов. – М.: Изд-во «Экзамен», 2006. – 573 с.
27. Шапкин А.С., Мазаева Н.П. Математические методы и модели исследования операций: Учебник. -3-е изд.- М.: Изд-во «Дашков и К», 2006. – 400 с.
28. Шикин Е.В., Шикина Г.Е. Исследование операций: Учебник. – М.: ТК «Велби», Изд-во «Проспект», 2006. – 280 с.

Учебное издание

Есипов Борис Алексеевич

МЕТОДЫ ОПТИМИЗАЦИИ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Учебное пособие

Технический Н. Н. Павлова
Редакторская обработка В. С. Телепова
Корректорская обработка Е. А. Ларионова
Доверстка А. С. Кочеулова

Подписано в печать 20.11.07. Формат 60x84 1/16.
Бумага офсетная. Печать офсетная.
Печ. л. 11,25.
Тираж 120 экз. Заказ 209 ИП-77/2007

Самарский государственный
аэрокосмический университет.
443086 Самара, Московское шоссе, 34.

Изд-во Самарского государственного
аэрокосмического университета.
443086 Самара, Московское шоссе, 34.